User’s Guide
Build Number 1424

Anton Betten

March 19, 2022
# Contents

1 Introduction 7

2 Getting Started 9
  2.1 Running and Installing Orbiter ................................................. 9
  2.2 The Orbiter Session ............................................................. 14
  2.3 Makefiles and Shell Scripts ..................................................... 17
  2.4 Objects and Activities ............................................................ 20
  2.5 Mathematical Data ................................................................. 24
  2.6 Set Builder .............................................................................. 26
  2.7 Vector Builders ........................................................................ 27

3 Basic Algebra 31
  3.1 Basic Number Theory ............................................................... 31
  3.2 Prime Fields ............................................................................ 38
  3.3 Extension Fields ....................................................................... 44
  3.4 Linear Algebra Over Finite Fields ............................................. 51
  3.5 Advanced Topics in Finite Fields ............................................. 63
  3.6 Basic Ring Theory .................................................................... 72

4 Geometry 73
  4.1 Finite Projective Spaces ............................................................ 73
  4.2 Indexing Points and Lines .......................................................... 80
  4.3 Finite Desarguesian Projective Planes ...................................... 85
  4.4 The Grassmannian .................................................................... 89
  4.5 Algebraic Sets ........................................................................... 91
  4.6 The Klein Quadric and the Plücker Map .................................. 96
  4.7 Orthogonal Spaces ................................................................... 98
  4.8 Hermitian Varieties .................................................................. 105
  4.9 Advanced Topics ...................................................................... 107
  4.10 Geometric Objects ................................................................... 114

5 Group Theory 119
  5.1 Permutation Groups ................................................................. 119
  5.2 Linear Groups ......................................................................... 127
  5.3 Subgroups ............................................................................... 144
Chapter 1

Introduction

Orbiter is a computer algebra system for the classification of combinatorial objects. Orbiter contributes to the knowledge base of combinatorial structures, and to provide useful tools to investigate structures from various points of view, including their symmetry properties. Orbiter is optimized for efficiency in terms of memory and execution speed. Orbiter is a library of C++ classes, together with a command line driven front end. There is no graphical user interface. The system offers two modes of use, programming or command line interface. This manual is about the command line interface. Readers who are interested in the Orbiter C++ class library should consult the programmer’s guide. A makefile with all commands used in this guide can be found in the examples subdirectory. For background on Orbiter, see [10].
Chapter 2

Getting Started

2.1 Running and Installing Orbiter

There are two ways to run Orbiter: Native and Docker. Native means that Orbiter is compiled from scratch, using the source code from the github repository (cf. [7]). Docker [24] is a system to run preconfigured software in an encapsulated way on various platform, including Windows. We describe using Orbiter through unix makefiles, which are run through the tool make (cf. [27]). This is a software tool that allows collecting short command snippets in the form of text files that can easily be handled. However, the conventions in the tool involve some subtleties regarding the use of whitespace, which can cause problems to novice users. We will point out possible pitfalls along the way. Note that it is not necessary to use makefiles. Another possibility would be to use shell scripts. Ultimately, it would be possible to type out all commands into a terminal window. This could be a little tedious though, considering the fact that most Orbiter commands expect lengthy parameters from the command line.

Let us start by discussing how to run Orbiter as a native application. To do so, a unix-like compile environment is required, including a modern C++ compiler and the tools git and make. Windows users may need to install Cygwin [21]. The following steps are required:

Using git, clone the repository. Then enter the directory orbiter and type

make

Once compiled, the Orbiter executable is

```
src/apps/orbiter/ orbiter.out
```

within the Orbiter directory. We then recommend creating a separate work directory not within the orbiter directory. For the following, we assume the following directory tree structure:

```
|--- orbiter
    |--- work
```

In the work directory, create a small makefile like so:
OP=../orbiter
ORBITER_PATH=$(OP)/src/apps/orbiter/

test:
  $(ORBITER_PATH)orbiter.out

Different directory structures can be accommodated by changing the first line. Next, typing

make test

within the work directory will invoke Orbiter. Here, test is the makefile “target.” The
makefile target must appear in the makefile. In the example above, the block

test:
  $(ORBITER_PATH)orbiter.out

is the makefile target “test.” It is important that the indentation after the makefile target is
done using tab characters (no spaces). There can be multiple targets in one makefile, as long
as they are separated by an empty line. For more information about the syntax of makefiles,
see [27].

A second way to run Orbiter is through Docker [24]. This does not require a compile environ-
ment. However, it comes at a small performance cost when running Orbiter commands that
are computationally heavy. Orbiter has already been precompiled (by the Orbiter developer)
into an image, which is a completely self-sustained copy of a unix-environment that can run
by the user under the docker front-end. The image is stored on a docker server under the
name abetten/orbiter. Docker will receive the name of the image from the command line,
pull a local copy of the image, and run the image in an encapsulated environment called a
container. A copy of the image is stored locally, so that subsequent calls to Orbiter can be
satisfied using the local copy, which increases turnaround speed. For instance, the following
bare-bones makefile sets up Orbiter for use through Docker:

DOCKER_OPTIONS=run -it \
  --volume ${PWD}:/mnt -w \
  /mnt abetten/orbiter
ORBITER_PATH=docker $(DOCKER_OPTIONS)

test:
  $(ORBITER_PATH)orbiter.out

In this file, there is a space character in line three after abetten/orbiter which is important
(and unfortunately cannot be seen). By typing

make test

into a terminal window, Docker starts up and pulls a copy of Orbiter to the local machine,
which is then executed. Orbiter will start up, produce a few messages and then shut down.
Interestingly, this will work on a Windows machine also (using supershell as terminal). The
make command is passed through to the container, which contains the unix-like software
environment, including make. The associated *makefile* resides on the local machine, as do input and output files.

Orbiter comes with a version numbering system called a build number. The build number should match the commit number on the github tree, shown in Figure 2.1. When Orbiter starts up, the build number is displayed. In order to update to a more recent version of Orbiter, Docker needs to be instructed to discard the local image. To do so, the command

```
docker rmi -f abetten/orbiter
```

is used. After that, any new invocation of Orbiter will cause Docker to pull the latest Orbiter *image* from the Docker repository. It is convenient to combine the Docker and Native compile environment into a single makefile and use the comment symbol (hash #) to switch between the two modes (the line numbers are not part of the file).

```make
OP=~/orbiter
OP2=$(OP)/src/apps/orbiter/
DOCKER_OPTS=run--it-
---volume=${PWD}:/mnt--w-
/mnt/abetten/orbiter
#ORBITER_PATH=docker$(DOCKER_OPTS)
ORBITER_PATH=$(OP2)
```

Here, whitespace characters can be seen: (spaces are shown as dots, and tab is a little triangle pointing to the right). Please observe the space at the end of line 5 and that the line(s) after the target(s) must start with a tab symbol (and no spaces). Also, the backslash signs are used to break long lines. Please make sure that there are no spaces after the backslash sign. In order to switch to Docker mode, the hash symbol can be removed in line 6 and instead put at the beginning of line 7. In the following examples, we assume that the 7 lines just shown are present at the beginning of the makefile. For brevity, we will only show the commands and their labels. These snippets must come after the top part.

For use with Docker, the installation of Orbiter requires the following steps:
(a) Install Docker from www.docker.com, including the Linux kernel.

(b) Open a terminal window (for instance PowerShell on Windows).

(c) Type

```
docker run -it --volume ${PWD}:/mnt -w /mnt abetten/orbiter orbiter.out
```

This will produce an output similar to the following:

```
sh-3.2$ docker run -it --volume $PWD:/mnt -w /mnt abetten/orbiter orbiter.out
Unable to find image 'abetten/orbiter:latest' locally
latest: Pulling from abetten/orbiter
004f1eed87df: Pull complete
5d6f1e8117db: Pull complete
48c2f9a66abe: Pull complete
234b70d0479d: Pull complete
6fa07a00e2f0: Pull complete
9187bd98e241: Pull complete
ae87b7ef500b: Pull complete
260a2765fa99: Pull complete
27d6ff93a58: Pull complete
7a09ec574418: Pull complete
1336494f74e1: Pull complete
Digest: sha256:889099d7e0b0a9ee168b7cb261d2da8ff64bd7d861c357e1caec59580d629ee9
Status: Downloaded newer image for abetten/orbiter:latest
Welcome to Orbiter! Your build number is 1311.
A user’s guide is available here:
https://www.math.colostate.edu/~betten/orbiter/users_guide.pdf
The sources are available here:
https://github.com/abetten/orbiter
An example makefile with many commands from the user’s guide is here:
https://github.com/abetten/orbiter/tree/master/examples/users_guide/makefile
SYSTEMUNIX is defined
sizeof(int)=4
sizeof(long int)=8
Orbiter session finished.
User time: 0:00
```

The first part is Docker downloading Orbiter as a container. This can take a while, depending on the Internet speed. The second part (Welcome to Orbiter!) is the actual Orbiter session. No specific commands were given, so Orbiter simply starts up and quits. The first part is done only once. Once it has been downloaded, Docker will recycle the copy of orbiter and a download is no longer required. However, once Orbiter updates, Docker will update the local copy of Orbiter as well.
To use Orbiter in native mode, the sources have to be installed and compiled. This is more complicated on Windows machines, because the unix environment is missing. Windows users can use cygwin to install Orbiter. The installation of Orbiter requires the following steps:

(a) Ensure that git and the C++ development suite are installed (gnuc and make). Windows users may have to install cygwin (plus the extra packages git, make, gnuc). Macintosh users may have to install the xcode development tools from the appstore (it is free). Linux users may have to install the development packages. Orbiter often produces latex reports. In order to compile these files, make sure you have latex installed.

(b) Clone the Orbiter source tree from github (abetten/orbiter). The commands are:

\begin{verbatim}
git clone <github-orbiter-path>
\end{verbatim}

where \texttt{<github-orbiter-path>} has to be replaced by the actual address provided by github. To obtain this path, find Orbiter on github, then click on the green box that says “Code” and copy the address into the clipboard by clicking the clipboard symbol (see Figure 2.2). Back in the terminal, paste this text after the \texttt{git clone} command. After cloning is complete, enter the orbiter directory (\texttt{cd orbiter}).

(c) Issue the following commands to compile Orbiter:

\begin{verbatim}
make
make install
\end{verbatim}

These two commands compile the Orbiter source tree and copy the executables to the subdirectory bin inside the Orbiter source tree. The orbiter executable is called \texttt{orbiter.out}.

Figure 2.2: GitHub Orbiter Page
2.2 The Orbiter Session

The orbiter workflow is depicted in Figure 2.3. Commands are issued through the command line, which invokes Orbiter sessions, which in turn perform the required computations and read and write data to files. The commands are parsed and separated into three basic types. Commands that create objects, commands that apply to previously created objects, and all other commands. Objects are maintained in a symbol table. The command line calls to Orbiter may or may not be organized in the form of makefiles, as discussed in Section 2.3.

Let us take a closer look at an Orbiter session. Any orbiter session is invoked through the orbiter command `orbiter.out`, which is the name of the executable. Unless the executable resides in a directory contained in the search path of the shell, a path must be given. Several options apply to the orbiter session. They are listed in Table 2.1. Once started, the Orbiter session will produce a short welcome message:

Welcome to Orbiter! Your build number is 1081.
A user’s guide is available here:
https://www.math.colostate.edu/~betten/orbiter/users_guide.pdf
The sources are available here:
https://github.com/abetten/orbiter
An example makefile with many commands from the user’s guide is here:
https://github.com/abetten/orbiter/tree/master/examples/users_guide/makefile
Orbiter session finished.
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>v</td>
<td>Set verbosity to ( v ). Larger values of ( v ) lead to more text output. ( v = 0 ) gives minimal output.</td>
</tr>
<tr>
<td>-list_arguments</td>
<td></td>
<td>Prints the command line arguments.</td>
</tr>
<tr>
<td>-seed</td>
<td>s</td>
<td>Seed the pseudo random number generator with the integer value ( s ).</td>
</tr>
<tr>
<td>-memory_debug</td>
<td></td>
<td>Turn on dynamic memory debugging.</td>
</tr>
<tr>
<td>-override_polynomial</td>
<td>poly</td>
<td>Set the override polynomial for finite fields to ( \text{poly} ).</td>
</tr>
<tr>
<td>-orbiter_path</td>
<td>p</td>
<td>Set the orbiter path to ( p ). This is useful in case the Orbiter session has to clone or fork new Orbiter sessions. In most cases, the orbiter path will end with a forward slash “/.”</td>
</tr>
<tr>
<td>-magma_path</td>
<td>p</td>
<td>Set the magma path to ( p ). This is useful in case the Orbiter session has to create a magma process.</td>
</tr>
<tr>
<td>-fork</td>
<td>( L \ M f t s )</td>
<td>Fork new Orbiter sessions in parallel. The new sessions will be indexed by the values ( i ) that result from a loop with start value ( f ) and increment ( s ) bounded from above by ( t ), equivalent to a C-loop of type “for (i=f; i &lt; t; i+= s).” Every occurrence of the string ( L ) in the argument list is replaced by the resulting value of the loop variable ( i ). The forked process will write to a file whose name is described through the mask ( M ). The actual file name results from using the printf command from the C-library for ( M ) with the integer value of the loop variable. All of the command line arguments after the fork command are passed through to the new Orbiter session, with all arguments ( L ) replaced by the integer value of the loop counter. The number of Orbiter sessions forked is ((t-f)/s). The orbiter path from -orbiter_path is used when starting the forked sessions.</td>
</tr>
</tbody>
</table>

Table 2.1: Orbiter session commands
The build number is the version number of the Orbiter software, as defined by the number of submits to the Git repository. Higher numbers mean more recent versions. After this message, Orbiter will start parsing the command line arguments. Once this is done, the session will execute these commands. At the end of the session, a short message is given that specifies the processor time used up by the session.
2.3 Makefiles and Shell Scripts

Orbiter is a command line driven system. There is no graphical user interface. This means that commands are typed into a terminal, and executed by the operating system. In this mode of operation, Orbiter is just like any other program installed on the computer. This also means that Orbiter can be mixed with other applications, using files to share data between the processes.

The command line is entered into an application that is called Terminal (or SuperShell in Windows). Orbiter is called from the command line, and command options are given to instruct Orbiter what to do. The process that calls orbiter is the shell. There are different types of shells, but they all provide the necessary interface to allow the user to start jobs and maintain files. Shells can be programmed by means of shell scripts. Programming by means of shell scripts is called scripting. Orbiter can be programmed using shell scripting.

One tool that stands out in the unix world is called make. Make is a command that allows to execute certain processes on a need basis. The need is defined by means of time stamps on files. The rules are defined in a file called makefile. Make is very popular in software engineering, where there are dependencies between source code, object code and executable files. We note that it is not necessary to use makefiles. However, because of the convenience they offer in defining lightweight commands, this user’s guide will rely on the make / makefile tool. It would also be possible to define shell scripts for each of the commands.

Orbiter can be used through makefiles, with or without using the dependency functionality. One feature of makefiles that is very useful is that commands can be defined very quickly, and that one makefile can hold many commands. This provides an advantage over shell scripting, where separate shell scripts are needed for each command. For instance, this user’s guide is based on a makefile that contains all commands shown. The makefile is listed in full in Section 19.1. In the user’s guide, the relevant pieces of code are shown one at a time. Make also allows to use variables, which are used by means of text substitution. A variable is defined as

A="I am a variable"

and used anywhere later using the

$(A)

syntax. Rules are defined using the following syntax

Label:
   Do something

Here, label is the name of the rule, and Do something is the code that is executed whenever make is called with the given label in the command line. For instance

make Label
will execute **Do something**. The shell will take the command and peel off the first word, which is **Do**. It will then search the system for a command called **Do**. Of course, this will result in an error because there is no command called **Do**. The remaining piece of the command line, i.e. **something** is considered as an argument to the command. For instance, suppose we have an Orbiter command with several options, say

```bash
orbiter.out -v 3 -define F -finite_field -q 16 -end \n  -with F -do -finite_field_activity -cheat_sheet_GF -end
```

The purpose of this command is to produce a file called

```
GF_16.tex
```

which can then be processed through LaTeX to give the report. Observe that the command is quite long, and stretches over two lines. The backslash at the end of the first line indicates that the command continues on to the next line. Using `make`, we can assign a label to this command. Suppose we want to call this command **F_16**. We can create a makefile like this:

```bash
F_16:
  $(ORBITER_PATH)orbiter.out -v 3 -define F -finite_field -q 16 -end \n  -with F -do -finite_field_activity -cheat_sheet_GF -end
```

With this file present, type the terminal command `make F_16` to execute the two line Orbiter command. Windows users can use **SuperShell**. The program `make` will look for the file `makefile` in the current directory. Once found, it will search for the label **F_16** in it and execute the commands beneath it. The given commands will invoke Orbiter and produce the **GF_16.tex** containing the desired report. If we wanted to do some other Orbiter command, we could edit the makefile. We would also have a sequence of commands listed in the same target. In this case, `makefile` will process these commands one after the other.

Makefiles are somewhat picky when it comes to whitespacing. The command sequence needs to be indented with tab symbols. Leading spaces will cause `make` to issue an error message. Also, there should be no whitespace after the trailing backslash symbol. Some editors can display whitespace characters. This may be helpful when editing the makefile.

A sample makefile with all of the commands discussed in this user’s guide is distributed with Orbiter (in the examples directory). The file is reproduced in Section 19.1. It is advised to copy the example makefile from the orbiter tree to a location outside the orbiter distribution directory (otherwise, `git` update will cause error messages). It is also fine to create a new custom makefile, considering the remarks about **ORBITER_PATH** below.

One difficulty in installing Orbiter is the path of installation. In the sample makefile, there is a makefile variable called **ORBITER_PATH** which contains the path to the orbiter executable **orbiter.exe**. Depending on the local installation of orbiter, the makefile variable needs to be changed accordingly. The actual command to run the **F_16** example is as follows:
F_16:
  $(ORBITER_PATH)orbiter.out -v 3 -define F -finite_field -q 16 -end \ 
  -with F -do -finite_field_activity -cheat_sheet_GF -end

We recommend the following configuration. The orbiter installation directory orbiter and a second directory called work should be next to each other. The orbiter example makefile should be copied into the work directory. The top of the file should contain the line

MY_PATH=../orbiter

This will set ORBITER_PATH to point to the correct location of the orbiter executable. inside the work directory, any of the commands listed in this guide will function correctly. Another possibility is to install orbiter.out in a central location. In this case, we should change the line

ORBITER_PATH=$(MY_PATH)/src/apps/orbiter/

to

ORBITER_PATH=

in the makefile.
2.4 Objects and Activities

Orbiter follows the object oriented paradigm. Mathematical objects of various types can be defined. Objects are maintained in a symbol table. New objects can be created from old. Activities can be applied to objects according to their type. By associating activities to objects of a certain type, Orbiter becomes more structured. It is easier to find the place where a certain functionality is defined, simply by searching by the type of object. This resembles the object oriented programming paradigm, where global functions are to be avoided and member functions of classes are preferred.

Objects can be of two types: primary or secondary. Objects of primary type can be created directly from scratch. Secondary objects depend on other objects that have to be created first. For instance, a finite field object is an object of primary type. A projective space is an object of secondary type because it needs a finite field object to be created first (the field over which the projective space is defined). Yet another type of objects are created from activities that are applied to objects. For instance, a cubic surface object can be created from a projective space object using the \texttt{-define_surface} command.

In this section, a brief overview of the types of objects is given, as well as the activities that can be applied. More details will be provided in later sections of this guide.

The syntax to create an object is

\begin{verbatim}
-define \textit{LABEL} \textit{KEYWORD} \textit{EXTRAS} -end
\end{verbatim}

Here, \textit{LABEL} is any label under which the object is stored in the symbol table. Any object with the same label already in the symbol table will be overwritten. The \textit{KEYWORD} can be any of the commands in Table 2.2. The \textit{EXTRAS} depend on the type of object created. The command \texttt{-end} is necessary to finish the definition. For more details on the objects that exist, see the appropriate section listed in the table. For instance, the command

\begin{verbatim}
object F_2:
\> $(ORBITER_PATH)orbiter.out\-v\3\-define\F\-finite_field\-q\2\-end
\end{verbatim}

creates a finite field object $F$ for the field with two elements (see Section 3.2). Once the field is created, the orbiter session terminates. The command

\begin{verbatim}
object PG_3_2:
\> $(ORBITER_PATH)orbiter.out\:
\> \> -define\F\-finite_field\-q\2\-end\:
\> \> \> -define\P\-projective_space\3\F\-end:
\end{verbatim}

creates the same finite field $F$ as well as an object $P$ representing $\text{PG}(3,2)$. Note how the creation of $P$ relies on the existence of $F$. The \texttt{-projective_space} option requires two parameters, the dimension of the projective space and the field over which it is defined. In
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-finite_field</code></td>
<td>A finite field $\mathbb{F}_q$. See Sections 3.2 and 3.3.</td>
</tr>
<tr>
<td><code>-projective_space</code></td>
<td>A projective space of dimension $n$ over a finite field $F$. See Section 4.1.</td>
</tr>
<tr>
<td><code>-orthogonal_space</code></td>
<td>A non-degenerate orthogonal space. See Section 4.7.</td>
</tr>
<tr>
<td><code>-linear_group</code></td>
<td>A linear group. See Section 5.2.</td>
</tr>
<tr>
<td><code>-permutation_group</code></td>
<td>A permutation group. See Section 5.1.</td>
</tr>
<tr>
<td><code>-formula</code></td>
<td>A symbolic expression. See Section 10.6.</td>
</tr>
<tr>
<td><code>-collection</code></td>
<td>A collection of objects.</td>
</tr>
<tr>
<td><code>-graph</code></td>
<td>A graph. See Section 13.1.</td>
</tr>
<tr>
<td><code>-spread_table</code></td>
<td>A table of spreads. See Section 12.3.</td>
</tr>
<tr>
<td><code>-packing_with_symmetry_assumption</code></td>
<td>A generator for packings with assumed symmetry. See Section 12.3.</td>
</tr>
<tr>
<td><code>-packing_choose_fixed_points</code></td>
<td>A selection of fixed orbits for packings with assumed symmetry. See Section 12.3.</td>
</tr>
<tr>
<td><code>-packing_long_orbits</code></td>
<td>A search for long orbits for packings with assumed symmetry. See Section 12.3.</td>
</tr>
<tr>
<td><code>-graph_classification</code></td>
<td>An object which allows classifying graphs and tournaments. See Section 13.3.</td>
</tr>
<tr>
<td><code>-diophant</code></td>
<td>A diophantine system, i.e., a system of positive integer equations). See Section 11.2.</td>
</tr>
<tr>
<td><code>-design</code></td>
<td>A combinatorial design. See Section 11.5.</td>
</tr>
<tr>
<td><code>-design_table</code></td>
<td>A table of designs. It can be used to construct large sets of designs. A large set is a set of designs satisfying certain properties. See Section 11.5.</td>
</tr>
<tr>
<td><code>-large_set_with_symmetry_assumption</code></td>
<td>An object to create a large set of designs. See Section 11.5.</td>
</tr>
<tr>
<td><code>-set</code></td>
<td>A set. See Section 2.6.</td>
</tr>
<tr>
<td><code>-vector</code></td>
<td>A vector over a finite field. See Section 2.7.</td>
</tr>
</tbody>
</table>

Table 2.2: Orbiter Objects
the example, the field $F$ which has been created earlier is referenced by its label as the second argument.

In order to do something with an object, we need to invoke an *activity*. To select an object for an activity, the

```
-with LABEL -do DESCRIPTION -end
```

command sequence is used. Here, *LABEL* is the name under which the object is registered in the symbol table. *DESCRIPTION* is the activity that should be applied. Some activities require more than one object, in which case the syntax

```
-with LABEL1 -and LABEL2 -do DESCRIPTION -end
```

is used. Here, *LABEL1* and *LABEL2* are the objects for which the activity is invoked. For an example of an activity requiring two objects, see Sections 12.1 and 12.2.

Table 2.3 list the possible activities for Orbiter objects. More details will be given in the later sections of this guide.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-finite_field_activity</td>
<td>An activity for finite fields, see Sections 3.2 and 3.3.</td>
</tr>
<tr>
<td>-projective_space_activity</td>
<td>An activity for a projective space, see Section 4.1.</td>
</tr>
<tr>
<td>-orthogonal_space_activity</td>
<td>An activity for an orthogonal space, see Section 4.7.</td>
</tr>
<tr>
<td>-group_theoretic_activity</td>
<td>An activity for a group, see Section 5.6.</td>
</tr>
<tr>
<td>-cubic_surface_activity</td>
<td>An activity for a cubic surface, see Section 7.1.</td>
</tr>
<tr>
<td>-quartic_curve_activity</td>
<td>An activity for a quartic curve, see Section 7.2.</td>
</tr>
<tr>
<td>-combinatorial_object_activity</td>
<td>An activity for a combinatorial object, see Section 4.5.</td>
</tr>
<tr>
<td>-graph_theoretic_activity</td>
<td>An activity for a graph, see Section 13.1.</td>
</tr>
<tr>
<td>-classification_of_cubic_surfaces_with_double_sixes_activity</td>
<td>An activity for a cubic surface, see Section 7.1.</td>
</tr>
<tr>
<td>-spread_table_activity</td>
<td>An activity associated with a table of spreads, see Section 12.3.</td>
</tr>
<tr>
<td>-packing_with_symmetry_assumption_activity</td>
<td>An activity related to creating packings with assumed symmetry group, see Section 12.3.</td>
</tr>
<tr>
<td>-packing_fixed_points_activity</td>
<td>An activity related to creating packings with assumed symmetry group, see Section 12.3.</td>
</tr>
<tr>
<td>-graph_classification_activity</td>
<td>An activity for a classification of graphs problem, see Section 13.3.</td>
</tr>
<tr>
<td>-diophant_activity</td>
<td>An activity for a diophantine system, see Section 11.2.</td>
</tr>
<tr>
<td>-design_activity</td>
<td>An activity for a combinatorial design, see Section 11.5.</td>
</tr>
<tr>
<td>-large_set_with_symmetry_assumption_activity</td>
<td>An activity related to creating large sets of designs with assumed symmetry group, see Section 11.6.</td>
</tr>
</tbody>
</table>

Table 2.3: Orbiter Activities
<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLT-sets</td>
<td>BLT sets of $Q(4, q)$ exist for all odd prime powers. The classification of BLT-sets of $Q(4, q)$ is known to Orbiter for all $q \leq 73$.</td>
</tr>
<tr>
<td>Cubic Surfaces</td>
<td>Cubic surfaces with 27 lines exist for all finite fields apart from $\mathbb{F}_2$, $\mathbb{F}_3$, $\mathbb{F}_5$. Orbiter knows the classification of cubic surfaces with 27 lines for all fields $\mathbb{F}_q$ of order $q \leq 128$.</td>
</tr>
<tr>
<td>Quartic curves</td>
<td>Orbiter knows the classification of smooth quartic curves with 28 bitangents in projective planes over all fields $\mathbb{F}_q$ for $q = 9, 13, 19, 25, 27, 29, 31$.</td>
</tr>
<tr>
<td>Spreads</td>
<td>A spread is a set of $q^k + 1$ pairwise non-intersecting $k$-dimensional subspaces of $\mathbb{F}_q^{2k}$. Spreads are related to translation planes of order $q^k$. Orbiter knows the classification of spreads for $(q, k) \in {(2, 2), (3, 2), (2, 4), (4, 2), (5, 2), (3, 3)}$.</td>
</tr>
<tr>
<td>Hyperovals</td>
<td>A hyperoval in PG$(2, 2^e)$ is a set of $2^e + 2$ points, no three collinear. Orbiter knows the classification of hyperovals for $e = 3, 4, 5$.</td>
</tr>
<tr>
<td>Dual hyperovals</td>
<td>A $k$-dimensional dual hyperoval in an ambient space $\mathbb{F}_2^n$ is called a DH$(k, n)$. Orbiter knows the classification of dual hyperovals DH$(4, 7)$ and DH$(4, 8)$.</td>
</tr>
<tr>
<td>Packings</td>
<td>Orbiter knows the classification of packings of PG$(3, 3)$.</td>
</tr>
</tbody>
</table>

Table 2.4: Mathematical Data Available in Orbiter

2.5 Mathematical Data

Orbiter serves as a repository for mathematical data. The knowledge base is concerned with classifications of geometric and combinatorial objects for small parameters. The types of objects for which a classification is available in Orbiter are listed in Table 2.4. The mathematical objects are stored in a catalogue, together with generators for their automorphism groups. The objects are indexed by a zero-based integer, called the *Orbiter Catalogue Number* (OCN). It is possible to access any object in the catalogue. Let us consider some examples:

The command

```
create_BLT_5_1:
  ▶ $(ORBITER_PATH)orbiter.out -v 2 \n  ▶ ▶ -define:F:-finite_field=q:5 -end \n  ▶ ▶ -define:O:-orthogonal_space=0:5:F:-end \n```
recalls the BLT-set with Orbiter Catalogue Number 1 in \( Q(4,5) \). A latex report `catalogue_q5_iso1.tex` is written. For more details about BLT-sets, see Section 12.4.

The command

```
create_surface_4_0:
  $(ORBITER_PATH)orbiter.out -v 3
  -define F -finite_field -q 4 -end
  -define P -projective_space -3F -end
  -with P -do
  -projective_space_activity
  -define_surface S4 0 -q 4 -catalogue 0 -end
  -end
  -with S4 0 -do
  -cubic_surface_activity
  -report
  -end
```

recalls the cubic surface with Orbiter Catalogue Number 0 in \( PG(3,4) \). A latex report `surface_catalogue_q4_iso0_report.tex` is written. For more details about cubic surfaces, see Section 7.1.
2.6 Set Builder

Orbiter allows to create objects of type set. Here is an example. We create the set $S$ of the first six prime numbers $\{2, 3, 5, 7, 11, 13\}$:

```plaintext
set_of_primes:
▷ $(\text{ORBITER\_PATH})\text{orbiter.out} -v 2 -define S -set -here "2,3,5,7,11,13" -end \-
▷ ▷ -print\_symbols
```

The next command creates the interval $[0, 63]$. We use the -loop command to save us from typing out all elements of the set. The -loop command has three arguments: the start value, the end value plus one, and the increment.

```plaintext
set_interval:
▷ $(\text{ORBITER\_PATH})\text{orbiter.out} -v 2 -define S -set -loop 0 64 1 -end \-
▷ ▷ -print\_symbols
```

For C programmers, -loop $a \ b \ c$ is equivalent to

```plaintext
for (i=a; i < b; i += c) {
}
```
2.7 Vector Builders

Orbiter allows to create objects of type vector. A vector is simply a data structure for a sequence of integers. It is similar to an array in a programming language. Orbiter does not force any kind of typing. The same vector can have many different meanings. For instance, indexing allows us to identify different types of objects with integers. For instance, a vector could be considered as a vector of elements of a finite field. This is because in Orbiter, finite field elements are ranked and represented as integers.

There are two different ways to define a vector, called dense and sparse format. In the dense format, the coefficients are listed in order from the lowest to the highest term. The -dense command creates the vector from a list of coefficients. The sparse format can be useful for coefficient vectors with few nonzero entries. It is a list of coefficient pairs, each of which describing one entry in the vector. One pair consists of the coefficient and the index of the term. The pairs are listed in sequence. The -sparse command creates the vectors from a given list of coefficient pairs.

If the option -field is given together with a field object, then Orbiter will force the vector entries to lie in the interval \([0, q - 1]\), where \(q\) is the order of the finite field. Otherwise, they can be any integer values. Note that there are limitations due to the word size of the machine and the processor. Most machines today have 64 bits, so any integer of absolute value less than \(2^{63} - 1\) can be represented (recall that the sign takes one bit away). On an older style 32 bit machine, only integers whose absolute value is less that \(2^{31} - 1\) can be stored. For more about limitations, see Section 17.2.

Here is an example. We first create the field \(\mathbb{F}_5\), and then create the vector \(v = (0, 1, 2, 3, 4)\). The -field option refers to the finite field created previously. The -dense option allows to enter the vector coefficients on the command line.

vector example1:

```
$ (ORBITER_PATH) orbiter.out -v:2 -
  -define:F:finite_field:-q:5:-end-
  -define:v:vector-field:F:-dense:"0,1,2,3,4":-end-
  -print_symbols
```

Vectors can also be read from file. The -file option can be used to name a csv file. In this case, the -dense option should not be used. A vector can also serve as a matrix. The -format \(k\) option can be used to specify the number \(k\) of rows. The number of columns is determined as \(n/k\), where \(n\) is the length of the vector given. For instance, the next example creates a \(2 \times 3\) matrix over \(\mathbb{F}_5\):

vector example2:

```
$ (ORBITER_PATH) orbiter.out -v:2 -
  -define:F:finite_field:-q:5:-end-
  -define:v:vector-field:F:-format:2:-dense:"0,1,2,3,4,0":-end-
  -print_symbols
```
For larger matrices, we can use makefile variables. For instance, the following command creates the generator matrix of the Hamming code:

```
HAMMING_CODE="1,0,0,0,0,1,1,0,1,0,0,1,0,1,0,0,1,1,0,0,0,1,1,1"
```

```
matrix_example1:
  $(ORBITER_PATH)orbiter.out -v 2
  -define:F:-finite_field:-q 2 -end
  -define:v:-vector_field:F:-format:4:-dense:\$(HAMMING_CODE) -end
  -print_symbols
```

For large matrices over small fields, the `-compact` option can be given (instead of `-dense`). For instance, the following code creates a $22 \times 22$ matrix over the binary field:

```
CONWAY_GEN1="110111000100001010000\n1111001111001001000101\n01010100000010011101\n00000000000010010101\n01000000000010010101\n00100001100000111111\n11010010011010001011\n00000000000011001010\n00000000000010010101\n01011111101001101111\n00000000000011001010\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101\n00000000000010010101"

matrix_example_co_1:
  $(ORBITER_PATH)orbiter.out -v 2
  -define:F:-finite_field:-q 2 -end
  -define:v:-vector_field:F:-format:22:-compact:\$(CONWAY_GEN1) -end
  -print_symbols
```

Using the dense option, spaces in the input string are ignored. For large vectors, the `sparse` command can be used to enter non-zero coefficients as a list of pairs. For instance,
vector_example_sparse:

```
$ (ORBITER_PATH) orbiter.out -v.2 -
   -define:F:-finite_field:-q.5:-end:-
   -define:v:-vector-field:F:-format:4:-sparse:20:"1,0,1,19":-end:-
   -print_symbols
```

creates a vector of length 20 and sets the 0-th and the 19-th coefficient to 1. Finally, the vector is displayed as a four-rowed matrix:

```
1 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 1
```

Orbiter has a command to create vectors whose entries repeat. For instance, the following code creates a vector of length 11 whose entries repeat over the sequence 0,1,2,3. It is not necessary that the vector length is an integer multiple of the length of the repeating sequence.

```
vector_example_repeat:

```
$ (ORBITER_PATH) orbiter.out -v.2 -
   -define:v:-vector-repeat:"0,1,2,3":11:-end:-
   -print_symbols
```

The sequence 0,1,2,3 is repeated sufficiently often to make a vector of length 11. This creates the vector

```
(0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2).
```

In order to create a constant vector, the `repeat` command can be used as well. Simply use a repeat sequence consisting of a single number. For instance, the following example creates the all-one vector of length 11:

```
vector_example_all_one_11:

```
$ (ORBITER_PATH) orbiter.out -v.2 -
   -define:v:-vector-repeat:1:11:-end:-
   -print_symbols
```

This code will create the all-one vector of length 11:

```
(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1).
```
Chapter 3

Basic Algebra

3.1 Basic Number Theory

Orbiter provides functions for computing with the ring of integers and integer factor rings. Computations with large integers are supported through a long integer data type which allows unrestricted precision. Table 3.1 shows Orbiter commands for basic number theory, including integer factor rings and the Euclidean algorithm. The command \texttt{order\_of\_q\_mod\_n} computes the order $\text{ord}(q,n)$ of $q$ modulo $n$ for all $n$ with $n_{\text{min}} \leq n \leq n_{\text{max}}$ for which $\text{gcd}(n,q) = 1$. Also computes Euler’s totient function $\varphi(n)$ and the cofactor $\varphi(n)/\text{ord}(q,n)$. For instance,

\begin{verbatim}
order_of_2_mod_n:
  $\langle ORBITER\_PATH\rangle$orbiter.out\-v\-3\-order\_of\_q\_mod\_n\2\3\151
  $\langle ORBITER\_PATH\rangle$orbiter.out\-v\-1\-csv\_file\_latex\1\$
  \texttt{order\_of\_q\_mod\_n\_q2\_3\_151.csv}
  pdflatex\texttt{order\_of\_q\_mod\_n\_q2\_3\_151.tex}
  open\texttt{order\_of\_q\_mod\_n\_q2\_3\_151.pdf}
\end{verbatim}

produces the output shown in Table 3.2.

The command

\begin{verbatim}
Eulerfunction\_150:
  $\langle ORBITER\_PATH\rangle$orbiter.out\-v\-1\-eulerfunction\_interval\1\150
  $\langle ORBITER\_PATH\rangle$orbiter.out\-v\-1\-csv\_file\_latex\1\$
  \texttt{table\_eulerfunction\_1\_150.csv}
  pdflatex\texttt{table\_eulerfunction\_1\_150.tex}
  open\texttt{table\_eulerfunction\_1\_150.pdf}
\end{verbatim}

computes the values of the Euler totient function for all $n$ with $1 \leq n \leq 150$. The result is shown in Table 3.3.

The function which raises every element to the $k$-th power modulo $n$ can be computed. For instance, the following command computes the function $a \mapsto a^k \mod 11$:
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-power_mod</td>
<td>$a \ n \ p$</td>
<td>Computes $a^n \mod p$.</td>
</tr>
<tr>
<td>-discrete_log</td>
<td>$b \ a \ p$</td>
<td>Computes $n$ such that $a^n \equiv b \mod p$.</td>
</tr>
<tr>
<td>-extended_gcd</td>
<td>$a \ b$</td>
<td>Computes integers $g, u, v$ such that $g = \gcd(a, b) = ua + vb$.</td>
</tr>
<tr>
<td>-square_root_mod</td>
<td>$a \ p$</td>
<td>Computes a square root of $a$ modulo $p$, i.e. an integer $b$ such that $b^2 \equiv a \mod p$.</td>
</tr>
<tr>
<td>-square_root</td>
<td>$a$</td>
<td>Computes $\lfloor \sqrt{a} \rfloor$ of an integer $a$.</td>
</tr>
<tr>
<td>-inverse_mod</td>
<td>$a \ p$</td>
<td>Computes the modular inverse of $a$ modulo $p$, i.e. an integer $b$ with $ab \equiv 1 \mod p$.</td>
</tr>
<tr>
<td>-draw_mod_n</td>
<td>descr</td>
<td>Draws the integers modulo $n$ on a circle.</td>
</tr>
<tr>
<td>-order_of_q_mod_n</td>
<td>q $n_{\text{min}}$ $n_{\text{max}}$</td>
<td>Computes the order $\text{ord}(q,n)$ of $q$ modulo $n$ for all $n$ with $n_{\text{min}} \leq n \leq n_{\text{max}}$ for which $\gcd(n,q) = 1$. Also computes $\varphi(n)$ and $\varphi(n)/\text{ord}(q,n)$.</td>
</tr>
</tbody>
</table>

Table 3.1: Basic Number Theory Commands
Table 3.2: The order of 2 modulo $n$

<table>
<thead>
<tr>
<th>N</th>
<th>ORD</th>
<th>PHI</th>
<th>COF</th>
<th>N</th>
<th>ORD</th>
<th>PHI</th>
<th>COF</th>
<th>N</th>
<th>ORD</th>
<th>PHI</th>
<th>COF</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>53</td>
<td>52</td>
<td>52</td>
<td>1</td>
<td>103</td>
<td>51</td>
<td>102</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>55</td>
<td>20</td>
<td>40</td>
<td>2</td>
<td>105</td>
<td>12</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>57</td>
<td>18</td>
<td>36</td>
<td>2</td>
<td>107</td>
<td>106</td>
<td>106</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>59</td>
<td>58</td>
<td>58</td>
<td>1</td>
<td>109</td>
<td>36</td>
<td>108</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>61</td>
<td>60</td>
<td>60</td>
<td>1</td>
<td>111</td>
<td>36</td>
<td>72</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>12</td>
<td>1</td>
<td>63</td>
<td>6</td>
<td>36</td>
<td>6</td>
<td>113</td>
<td>28</td>
<td>112</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>65</td>
<td>12</td>
<td>48</td>
<td>4</td>
<td>115</td>
<td>44</td>
<td>88</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>16</td>
<td>2</td>
<td>67</td>
<td>66</td>
<td>66</td>
<td>1</td>
<td>117</td>
<td>12</td>
<td>72</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>69</td>
<td>22</td>
<td>44</td>
<td>2</td>
<td>119</td>
<td>24</td>
<td>96</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>71</td>
<td>35</td>
<td>70</td>
<td>2</td>
<td>121</td>
<td>110</td>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>11</td>
<td>22</td>
<td>2</td>
<td>73</td>
<td>9</td>
<td>72</td>
<td>8</td>
<td>123</td>
<td>20</td>
<td>80</td>
<td>4</td>
</tr>
<tr>
<td>25</td>
<td>20</td>
<td>20</td>
<td>1</td>
<td>75</td>
<td>20</td>
<td>40</td>
<td>2</td>
<td>125</td>
<td>100</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>77</td>
<td>30</td>
<td>60</td>
<td>2</td>
<td>127</td>
<td>7</td>
<td>126</td>
<td>18</td>
</tr>
<tr>
<td>29</td>
<td>28</td>
<td>28</td>
<td>1</td>
<td>79</td>
<td>39</td>
<td>78</td>
<td>2</td>
<td>129</td>
<td>14</td>
<td>84</td>
<td>6</td>
</tr>
<tr>
<td>31</td>
<td>5</td>
<td>30</td>
<td>6</td>
<td>81</td>
<td>54</td>
<td>54</td>
<td>1</td>
<td>131</td>
<td>130</td>
<td>130</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>10</td>
<td>20</td>
<td>2</td>
<td>83</td>
<td>82</td>
<td>82</td>
<td>1</td>
<td>133</td>
<td>18</td>
<td>108</td>
<td>6</td>
</tr>
<tr>
<td>35</td>
<td>12</td>
<td>24</td>
<td>2</td>
<td>85</td>
<td>8</td>
<td>64</td>
<td>8</td>
<td>135</td>
<td>36</td>
<td>72</td>
<td>2</td>
</tr>
<tr>
<td>37</td>
<td>36</td>
<td>36</td>
<td>1</td>
<td>87</td>
<td>28</td>
<td>56</td>
<td>2</td>
<td>137</td>
<td>68</td>
<td>136</td>
<td>2</td>
</tr>
<tr>
<td>39</td>
<td>12</td>
<td>24</td>
<td>2</td>
<td>89</td>
<td>11</td>
<td>88</td>
<td>8</td>
<td>139</td>
<td>138</td>
<td>138</td>
<td>1</td>
</tr>
<tr>
<td>41</td>
<td>20</td>
<td>20</td>
<td>2</td>
<td>91</td>
<td>12</td>
<td>72</td>
<td>6</td>
<td>141</td>
<td>46</td>
<td>92</td>
<td>2</td>
</tr>
<tr>
<td>43</td>
<td>14</td>
<td>42</td>
<td>3</td>
<td>93</td>
<td>10</td>
<td>60</td>
<td>6</td>
<td>143</td>
<td>60</td>
<td>120</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>12</td>
<td>24</td>
<td>2</td>
<td>95</td>
<td>36</td>
<td>72</td>
<td>2</td>
<td>145</td>
<td>28</td>
<td>112</td>
<td>4</td>
</tr>
<tr>
<td>47</td>
<td>23</td>
<td>46</td>
<td>2</td>
<td>97</td>
<td>48</td>
<td>96</td>
<td>2</td>
<td>147</td>
<td>42</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>49</td>
<td>21</td>
<td>42</td>
<td>2</td>
<td>99</td>
<td>30</td>
<td>60</td>
<td>2</td>
<td>149</td>
<td>148</td>
<td>148</td>
<td>1</td>
</tr>
<tr>
<td>51</td>
<td>8</td>
<td>32</td>
<td>4</td>
<td>101</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>151</td>
<td>15</td>
<td>150</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3.3: The values of the Eulerfunction
Table 3.4: The function $a \mapsto a^2 \mod 11$

<table>
<thead>
<tr>
<th>A</th>
<th>APOWK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

The result is shown in Table 3.3.

The command

PR29:

$\backslash$(ORBITER\_PATH)orbiter.out--v.5--smallest\_primitive\_root.29

computes a primitive root modulo 29 using a randomized algorithm. The answer in this case is 2. For a large example, consider

PR2:

$\backslash$(ORBITER\_PATH)orbiter.out--v.5--primitive\_root.915839

which computes a primitive root modulo 915839. The answer is 43085. The command

PM2:

$\backslash$(ORBITER\_PATH)orbiter.out--v.5--power\_mod.43085.49842.915839

computes

$$43085^{49842} \mod 915839$$

which is 487320. Conversely, the discrete log of 487320 with respect to the base 43085 modulo 915839 can be computed using the command
The answer to this command is 49842. This command is a brute force search, and can be quite expensive. The command

```
IM:
```

```bash
$ (ORBITER_PATH)orbiter.out\-v\-5\-inverse_mod\1865025205\2147483647
```

computes the inverse of 1865025205 modulo 2147483647 which is 579785381. A different way of computing the inverse is using the 1-trick. The `-extended_gcd` command can be used:

```
IM\_gcd:
```

```bash
$ (ORBITER_PATH)orbiter.out\-v\-5\-extended_gcd\1865025205\2147483647
```

This command produces the output

```
1 = -503526232 \times 2147483647 + 579785381 \times 1865025205
```

which is the gcd written as a lattice combination of the input arguments. The inverse of 1865025205 mod 2147483647 is the coefficient in front of the 1865025205. In order to compute the modular power

```
a^e \mod n,
```

the `-power_mod` command can be used. For instance,

```
PM3a:
```

```bash
$ (ORBITER_PATH)orbiter.out\-v\-5\-power_mod\16807\1073741823\2147483647
```

computes 16807 raised to the power 1073741823 modulo 2147483647, which is 2147483646. In order to compute the modular square root, i.e. to solve for $x$ in

```
x^2 \equiv a \mod p
```

the `-square_root_mod` command can be used. For instance,

```
sqrt\_mod:
```

```bash
$ (ORBITER_PATH)orbiter.out\-v\-2\-square_root_mod\33\41
```

finds that the square root of 33 mod 41 is 19, i.e.

```
19^2 \equiv 33 \mod 41.
```

This command applies the algorithm of Tonelli and Shanks (cf. [19]).

The command
computes the powers of 2 mod 13 and connects consecutive powers along the circle modulo 13. By changing the value of the base, the diagrams in Figure 3.1 are created. The cases $b = 2$ and $b = 6$ are special. In those cases, the sequence of powers of $b$ mod 13 loops back unto itself after visiting all non-zero elements modulo 13. This is because 2 and 6 are primitive elements modulo 13. Because $-1$ is a square modulo 13, the power cycles of $b$ and of $-b$ have the same length, so $-2 = 11$ and $-6 = 5$ are primitive elements also. In total, there are 4 primitive elements modulo 13. This agrees with $\varphi(12) = 4$, where $\varphi(k)$ is Euler’s totient function, which counts the number of generators in the cyclic group of order $k$. However, this reasoning relies on the fact that 13 is prime, which implies that the group of prime residues modulo 13 is cyclic.

The command
draw_mod_13:

computes the powers of 2 mod 13 and connects consecutive powers along the circle modulo 13. By changing the value of the base, the diagrams in Figure 3.1 are created. The cases $b = 2$ and $b = 6$ are special. In those cases, the sequence of powers of $b$ mod 13 loops back unto itself after visiting all non-zero elements modulo 13. This is because 2 and 6 are primitive elements modulo 13. Because $-1$ is a square modulo 13, the power cycles of $b$ and of $-b$ have the same length, so $-2 = 11$ and $-6 = 5$ are primitive elements also. In total, there are 4 primitive elements modulo 13. This agrees with $\varphi(12) = 4$, where $\varphi(k)$ is Euler’s totient function, which counts the number of generators in the cyclic group of order $k$. However, this reasoning relies on the fact that 13 is prime, which implies that the group of prime residues modulo 13 is cyclic.

The command
draw_mod_127:
Figure 3.2: Cycle of powers of 3 modulo 127

\[
\text{\texttt{\$(ORBITER\_PATH)orbiter\_out\_v\_2\_draw\_options\_scale\_0.8\_embedded\_end}}
\]

\[
\text{\texttt{-draw\_mod\_n\_n\_127\_file\_mod\_127\_power\_cycle\_3\_end}}
\]

\[
\text{\texttt{pdflatex\_mod\_127\_draw\_tex}}
\]

\[
\text{\texttt{open\_mod\_127\_draw\_pdf}}
\]

creates the drawing shown in Figure 3.2.
3.2 Prime Fields

Let \( \mathbb{F}_q \) denote the finite field with \( q \) elements. Up to isomorphism, there is only one field of order \( q \). Finite fields of prime order can be created as integer factor ring.

\[
I(p) = p\mathbb{Z} = \{ pk \mid k \in \mathbb{Z} \} = \{ 0, \pm k, \pm 2k, \pm 3k, \ldots \}
\]
is the ideal of all integer multiples of \( p \). The elements of \( \mathbb{F}_p \) are the residue classes of the ideal given by the integer multiples of \( p \). Each residue class has the form
\[
\{ a + kp \mid k \in \mathbb{Z} \}.
\]

Standard representatives of the equivalence classes can be chosen as the smallest non-negative member in each class. This means that the standard representatives are the integers from 0 to \( p - 1 \). This canonical representative is the remainder after division by \( p \). Two integers belong to the same residue class if they have the same remainder after division by \( p \). For instance, 11 and 46 are in the same residue class modulo 5 because both have a remainder of 1 after division by five. It is convenient to identify the residue classes mod \( p \) with the integers from 0 to \( p - 1 \). In Orbiter, this convention is used automatically. The addition table and the multiplication table can be used to add and multiply in \( \mathbb{F}_p \). For instance, in Figure 3.3 the addition and multiplication tables of \( \mathbb{F}_7 \) are shown, both numerically and using colors. The natural ordering of the integers in the interval \([0, 6]\) is used. Different integers are represented by different colors. It is customary to restrict the multiplication table to the non-zero elements of the field.

A finite field \( \mathbb{F}_q \) can be created using the -finite_field command. Table 3.5 lists Orbiter commands for creating a finite field that can come after -finite_field. For instance,

\[
\text{F}_2:
\]

\[
\text{\textgreater} \quad \text{\$\{ORBITER\_PATH\}orbiter.out\cdot-v\cdot3\cdot\text{-define-}F\cdot\text{-finite_field}\cdot-q\cdot2\cdot\text{-end}::}
\]

\[
\text{\textgreater} \quad \text{\textgreater} \quad \text{-with-F\cdot\-do-\text{-finite_field_activity}\cdot\text{-cheat_sheet_GF}\cdot\text{-end}}
\]

\[
\text{\textgreater} \quad \text{pdflatex:GF_2.tex}
\]

\[
\text{\textgreater} \quad \text{open:GF_2.pdf}
\]

creates the finite field \( \mathbb{F}_2 \) and produces a report for it.

Table 3.6 lists basic Orbiter activities for finite fields. More activities will follow in Section 3.3. Here is the cheat sheet for \( \mathbb{F}_7 \). The element \( \alpha \) is a primitive element.
Figure 3.3: Addition and multiplication tables of $\mathbb{F}_7$

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--q</code></td>
<td>$q$</td>
<td>Specify the order of the field. Here, $q = p^k$ for some prime $p$ and some positive integer $k$.</td>
</tr>
<tr>
<td><code>--override_polynomial</code></td>
<td>$n$</td>
<td>Specify the polynomial used to create the finite field. The polynomial is given as integer, using the base $p$ representation. See Section 3.3.</td>
</tr>
<tr>
<td><code>--without_tables</code></td>
<td></td>
<td>Create the field without precomputing the tables.</td>
</tr>
</tbody>
</table>

Table 3.5: Options for Creating Finite Fields
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cheat_sheet_GF</td>
<td></td>
<td>Produce a cheat sheet in latex which shows information about the field, including addition and multiplication tables.</td>
</tr>
<tr>
<td>-product_of</td>
<td>$v$</td>
<td>Compute the product of all field elements in the vector $v$.</td>
</tr>
<tr>
<td>-sum_of</td>
<td>$v$</td>
<td>Compute the sum of all field elements in the vector $v$.</td>
</tr>
<tr>
<td>-negate</td>
<td>$v$</td>
<td>Negate each field element in the vector $v$.</td>
</tr>
<tr>
<td>-inverse</td>
<td>$v$</td>
<td>Compute the multiplicative inverse of each field element in the vector $v$.</td>
</tr>
<tr>
<td>-power_map</td>
<td>$k, v$</td>
<td>Compute the $k$-th power of each field element in the vector $v$.</td>
</tr>
</tbody>
</table>

Table 3.6: Finite Field Activities

$$Z_i = \log_\alpha (1 + \alpha^i)$$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\gamma_i$</th>
<th>$-\gamma_i$</th>
<th>$\gamma_i^{-1}$</th>
<th>$\log_\alpha(\gamma_i)$</th>
<th>$\alpha^i$</th>
<th>$Z_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 = 0</td>
<td>0</td>
<td>DNE</td>
<td>DNE</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1 = 1</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2 = $\alpha^2$</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3 = $\alpha$</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>DNE</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4 = $\alpha^4$</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5 = $\alpha^5$</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>6 = $\alpha^3$</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

$+$

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Suppose we want to check Wilson’s theorem that the product of all nonzero field elements of negative one. The following command so so, assuming that \( p = 11 \). We first create a vector of all nonzero field elements, which we take as the integers from 1 to 10. After that, we use the product_of finite field activity to compute the product of these elements. The answer is 10 which is congruent to \(-1 \mod 11\):

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 1 & 2 & 3 & 4 & 5 \\
2 & 2 & 4 & 6 & 1 & 3 \\
3 & 3 & 6 & 2 & 5 & 1 \\
4 & 4 & 1 & 5 & 2 & 6 \\
5 & 5 & 3 & 1 & 6 & 4 \\
6 & 6 & 5 & 4 & 3 & 2 \\
\end{array}
\]

\[
\begin{array}{ccc}
3^0 \equiv 1 & 3^4 \equiv 4 \\
3^1 \equiv 3 & 3^5 \equiv 5 \\
3^2 \equiv 2 & 3^6 \equiv 1 \\
3^3 \equiv 6 & \\
\end{array}
\]

Suppose we want to create the Vandermonde matrix whose entries are \( x_i^j \). Here \( x_0, \ldots, x_{q-1} \) are the elements of the field \( \mathbb{F}_q \) and \( j \) ranges from 0 to \( q - 1 \). The following command does so for \( q = 7 \). The command also computes the inverse of the Vandermonde matrix:

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 1 & 2 \\
2 & 2 & 4 \\
3 & 3 & 6 \\
4 & 4 & 1 \\
5 & 5 & 3 \\
6 & 6 & 5 \\
\end{array}
\]

The output is shown below. The first matrix is \( V = (x_i^j) \). The second matrix is \( V^{-1} \).
There is a second ordering of the elements which is used occasionally. In this labeling, every non-zero element is written as a power of a fixed primitive element. So, if $\alpha$ is a primitive element, we arrange the elements of $\mathbb{F}_p$ as $0, 1, \alpha, \alpha^2, \ldots, \alpha^{q-2}$.

The cheat sheet contains this list of field elements at the very end. In Figure 3.4, the addition and multiplication tables of $\mathbb{F}_7$ are shown with respect to the cyclic ordering of elements as $0, 1, 3, 0, 3^1, 3^2, \ldots, 3^6 = 0, 1, 3, 2, 6, 4, 5, 1$.

In the second ordering, the addition table of the prime field no longer exhibits cyclic structure.

The finite field class is based on precomputed tables for the arithmetic operations. This can be a problem for large fields. Precomputing the tables may not be worthwhile. For this reason, the option `-without_tables` can be given. The field is constructed without precomputed tables. Here is an example. We create the field $\mathbb{F}_{101}$ without precomputed tables:

F_101_wo:
$\$(ORBITER\_PATH)\text{orbiter.out}\ -v.3\$

\texttt{-define F:-finite\_field:-q.101:-without\_tables:-end}\$

\texttt{-with F:-do:-finite\_field\_activity:-cheat\_sheet\_GF:-end}

\texttt{pdflatex GF_101.tex}

\texttt{open GF_101.pdf}
3.3 Extension Fields

Let $F$ be a field. An extension field of $F$ is any field $E$ which contains $F$. Because $E$ is a vector space over $F$, the dimension of $E/F$ is well-defined. It may be finite or infinite. An example of a field extension is a field of the form $E = F(\alpha)$, where $\alpha$ is any element over $F$. Here, $F(\alpha)$ is the smallest field which contains $F$ and $\alpha$. If $\gamma \in E$ satisfies a polynomial equation with coefficients in $F$, then $\gamma$ is called algebraic over $F$. The minimum polynomial of an element $\gamma$ in $E$ over $F$ is the monic, lowest degree polynomial in $F[X]$ which has $\gamma$ as a root. A field extension $E/F$ is algebraic if every element in $E$ is algebraic over $F$. In particular, $F(\alpha)$ is algebraic over $F$ if $\alpha$ is. The degree of $E/F$ equals the degree of the minimum polynomial of $\alpha$ over $F$.

In this section, we will consider algebraic extension of finite fields. If $F = \mathbb{F}_q$ is a field of order $q$, then any algebraic extension $E$ of $F$ has order $q^e$ where $e$ is the degree of $E$ over $F$. If $E = F(\alpha)$ is algebraic, the degree of $E$ over $F$ is the degree of the minimum polynomial of $E$ over $F$. If $F = \mathbb{F}_q$ and $E = F(\alpha)$ is algebraic of degree $e$, then $|F| = q^e$. Every finite field $E$ is of this form, where $F = \mathbb{F}_p$ and $p$ is the characteristic of $E$.

Any such $E$ can be constructed as a polynomial factorring of the ring $\mathbb{F}_p[X]$. For a polynomial $m(X)$ we consider the ideal

$$I(m) = m(X)\mathbb{F}_p[X] = \{m(X)k(X) \mid k(X) \in \mathbb{F}_p[X]\}$$

of all polynomial multiples of $m(X)$. Under the assumption that $m(X)$ has degree $e > 1$ and is irreducible, the residue class ring

$$\mathbb{F}_p[X]/I(m)$$

is a field with $q = p^e$ elements. Each residue class has a canonical representative. The canonical representative is the unique element in the residue class which has degree less than $e$ and leading coefficient one. By means of identification, we can take these polynomials to be the set of standard representatives of the residue classes. So, for instance, for $q = 4 = 2^2$, we can pick the irreducible polynomial $m(X) = X^2 + X + 1$ over $\mathbb{F}_2$ and have four standard representatives modulo $I(m)$, namely

$$0, \quad 1, \quad X, \quad X + 1.$$ 

Together, these make up a complete set of representatives of the residue classes modulo $I(m)$, and hence can be identified with the elements of $\mathbb{F}_4$:

$$\mathbb{F}_4 = \{0, 1, X, X + 1\}.$$
The addition of polynomials is as in $\mathbb{F}_2[X]$, so

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>$X$</th>
<th>$X + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$X$</td>
<td>$X + 1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$X + 1$</td>
<td>$X$</td>
</tr>
<tr>
<td>$X$</td>
<td>$X$</td>
<td>$X + 1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$X + 1$</td>
<td>$X + 1$</td>
<td>$X$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

To compute the multiplication table for the field $\mathbb{F}_4$. We can use polynomial arithmetic modulo $m(X)$: It is clear how multiplication by 0 or 1 works, so we need to focus on the polynomials $X$ and $X + 1$:

- $X \cdot X = X^2 \equiv X + 1 \pmod{X^2 + X + 1}$,
- $X \cdot (X + 1) = X^2 + X \equiv 1 \pmod{X^2 + X + 1}$,
- $(X + 1) \cdot X = X^2 + X \equiv 1 \pmod{X^2 + X + 1}$,
- $(X + 1) \cdot (X + 1) = X^2 + 1 \equiv X \pmod{X^2 + X + 1}$,

so the multiplication table of $\mathbb{F}_4$ turns out to be

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>$X$</th>
<th>$X + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$X$</td>
<td>$X + 1$</td>
</tr>
<tr>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X + 1$</td>
<td>1</td>
</tr>
<tr>
<td>$X + 1$</td>
<td>0</td>
<td>$X + 1$</td>
<td>1</td>
<td>$X$</td>
</tr>
</tbody>
</table>

Figure 3.5 shows a graphical representation of the addition and multiplication tables of $\mathbb{F}_4$ using colors to represent the different elements: White is zero, black is one, red is $X$ and green is $X + 1$. In the multiplication table, the row and column associated with the zero elements are removed.

Table 3.7 lists Orbiter activities for finite fields. This extends Table 3.6 in Section 3.3.

The isomorphism type of the resulting field only depends on the order $q$ of the field, and not on the choice of the polynomial. However, for practical computations, the choice of the polynomial matters. For instance, results can only be shared between different computer algebra
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-trace</td>
<td></td>
<td>Computes the partitioning of the field elements according to the value of their absolute trace.</td>
</tr>
<tr>
<td>-norm</td>
<td></td>
<td>Computes the partitioning of the field elements according to the value of their absolute norm.</td>
</tr>
<tr>
<td>-normal_basis</td>
<td>$d$</td>
<td>Computes a normal basis for $\mathbb{F}_{q^d}$.</td>
</tr>
</tbody>
</table>

Table 3.7: More Finite Field Activities

systems if the same polynomials are used. Orbiter has a large collection of polynomials built in. Besides these, a polynomial can be specified. The polynomials that Orbiter offers are in fact primitive, which means that the root $\alpha$ is a primitive element for the field $\mathbb{F}_q$. This just means that it is a generator of the multiplicative group. So, any non-zero element in $\mathbb{F}_q$ is a suitable power of $\alpha$.

If $\mathbb{F}_q$ is an extension of the prime field $\mathbb{F}_p$, we use a different labeling. This time, we exploit the fact that $\mathbb{F}_q$ is a vector space over $\mathbb{F}_p$. Let $\alpha$ be a root of the irreducible polynomial $m(X) \in \mathbb{F}_p[X]$ used to create the field. Suppose that $q = p^e$, i.e., the degree of $m(X)$ is $e$. An $\mathbb{F}_p$-basis for the vector space $\mathbb{F}_q$ over $\mathbb{F}_p$ is given by the powers $\alpha^i$, for $0 \leq i < e$. Therefore, any element $\gamma$ of $\mathbb{F}_q$ has a unique expression of the form

$$\gamma = \sum_{i=0}^{e-1} a_i \alpha^i, \quad 0 \leq a_i < p \text{ for all } i.$$  

The associated integer rank of $\gamma$ is obtained by replacing $\alpha$ by $p$ in this expression and evaluating the expression over the integers. So, the rank of $\gamma$ is

$$\sum_{i=0}^{e-1} a_i p^i.$$  

As $\gamma$ ranges over all field element in $\mathbb{F}_q$, the rank values take on every value in the interval $[0, q - 1]$. The ordering of elements of $\mathbb{F}_q$ according to these ranks is called the lexicographical ordering. The numerical rank of zero is 0 and the numerical rank of one is 1. The numerical rank of $\alpha$, the primitive element, is $p$. The numerical ranks of the elements of the prime subfield are exactly the elements of $[0, p - 1]$.

The primitive polynomials used by Orbiter to create small finite fields are listed in Table 3.8. The relation is given using the Greek letter that is used in orbiter cheat sheets for that particular field.

Let us look at a few examples. The command
<table>
<thead>
<tr>
<th>$q$</th>
<th>Polynomial</th>
<th>Numerical</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$X^2 + X + 1$</td>
<td>7</td>
<td>$\omega^2 = \omega + 1$</td>
</tr>
<tr>
<td>8</td>
<td>$X^3 + X^2 + 1$</td>
<td>13</td>
<td>$\gamma^3 = \gamma^2 + 1$</td>
</tr>
<tr>
<td>9</td>
<td>$X^2 + X + 2$</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>$X^4 + X^3 + 1$</td>
<td>25</td>
<td>$\delta^4 = \delta^3 + 1$</td>
</tr>
<tr>
<td>25</td>
<td>$X^2 + X + 2$</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>$X^3 + 2X + 1$</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>$X^5 + X^2 + 1$</td>
<td>37</td>
<td>$\eta^5 = \eta^2 + 1$</td>
</tr>
<tr>
<td>49</td>
<td>$X^2 + X + 3$</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>$X^6 + X^5 + 1$</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>$X^4 + X^3 + 2$</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>$X^2 + 4X + 2$</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>$X^3 + X^2 + X + 2$</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>$X^7 + X^6 + 1$</td>
<td>193</td>
<td>$\zeta^7 = \zeta^6 + 1$</td>
</tr>
<tr>
<td>169</td>
<td>$X^2 + X + 2$</td>
<td>184</td>
<td></td>
</tr>
<tr>
<td>243</td>
<td>$X^5 + 2X + 1$</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>$X^8 + X^4 + X^3 + X^2 + 1$</td>
<td>285</td>
<td></td>
</tr>
<tr>
<td>289</td>
<td>$X^2 + X + 3$</td>
<td>309</td>
<td></td>
</tr>
<tr>
<td>343</td>
<td>$X^3 + 3X + 2$</td>
<td>366</td>
<td></td>
</tr>
<tr>
<td>361</td>
<td>$X^2 + X + 2$</td>
<td>382</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>$X^9 + X^4 + 1$</td>
<td>529</td>
<td></td>
</tr>
<tr>
<td>529</td>
<td>$X^2 + 2X + 5$</td>
<td>580</td>
<td></td>
</tr>
<tr>
<td>625</td>
<td>$X^4 + X^3 + X + 2$</td>
<td>326</td>
<td></td>
</tr>
<tr>
<td>729</td>
<td>$X^6 + X^5 + 2$</td>
<td>974</td>
<td></td>
</tr>
<tr>
<td>841</td>
<td>$X^2 + 5X + 2$</td>
<td>988</td>
<td></td>
</tr>
<tr>
<td>961</td>
<td>$X^2 + 2X + 3$</td>
<td>1026</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>$X^{10} + X^3 + 1$</td>
<td>1033</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.8: Orbiter primitive polynomials for fields $\mathbb{F}_q$ with $q \leq 1024$
Table 3.9: The field $F_{16}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\gamma_i$</th>
<th>$-\gamma_i$</th>
<th>$\gamma_i^{-1}$</th>
<th>$\log_{\alpha}(\gamma_i)$</th>
<th>$\alpha^i$</th>
<th>$Z_i$</th>
<th>$T(\gamma_i)$</th>
<th>$\phi(\gamma_i)$</th>
<th>$N(\gamma_i)$</th>
<th>$N_2(\gamma_i)$</th>
<th>$N_3(\gamma_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 = 0</td>
<td>0</td>
<td>DNE</td>
<td>1</td>
<td>DNE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 = 1</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>2</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha = \delta$</td>
<td>2</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>$\alpha + 1 = \delta^{12}$</td>
<td>3</td>
<td>8</td>
<td>12</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>$\alpha^2 = \delta^2$</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>$\alpha^2 + 1 = \delta^9$</td>
<td>5</td>
<td>15</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>$\alpha^2 + \alpha = \delta^{13}$</td>
<td>6</td>
<td>4</td>
<td>13</td>
<td>15</td>
<td>8</td>
<td>13</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>$\alpha^3 + \alpha + 1 = \delta^7$</td>
<td>7</td>
<td>14</td>
<td>7</td>
<td>7</td>
<td>13</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>$\alpha^3 = \delta^3$</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>14</td>
<td>6</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>$\alpha^3 + 1 = \delta^4$</td>
<td>9</td>
<td>13</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>$\alpha^3 + \alpha = \delta^{10}$</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>$\alpha^3 + \alpha + 1 = \delta^5$</td>
<td>11</td>
<td>10</td>
<td>5</td>
<td>13</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>$\alpha^3 + \alpha^2 = \delta^{14}$</td>
<td>12</td>
<td>2</td>
<td>14</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>$\alpha^3 + \alpha^2 + 1 = \delta^{11}$</td>
<td>13</td>
<td>9</td>
<td>11</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>$\alpha^3 + \alpha^2 + \alpha = \delta^8$</td>
<td>14</td>
<td>7</td>
<td>8</td>
<td>12</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>$\alpha^3 + \alpha^2 + \alpha + 1 = \delta^6$</td>
<td>15</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>DNE</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

Unlike other computer algebra systems (GAP [28] and Magma [14]), Orbiter does not use Conway polynomials to create field extensions. Instead, it provides the option to override the polynomial used to create the finite field. For subfield relationships, the cheat sheet will indicate the irreducible polynomials of all subfields for a given field. For instance, Table 3.10
<table>
<thead>
<tr>
<th>Subfield</th>
<th>Polynomial</th>
<th>Numerical rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{F}_4 )</td>
<td>( X^2 + X + 1 )</td>
<td>7</td>
</tr>
<tr>
<td>( \mathbb{F}_8 )</td>
<td>( X^3 + X + 1 )</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 3.10: The subfields of \( \mathbb{F}_{64} \)

Figure 3.6: Addition and multiplication table of \( \mathbb{F}_3 \) and \( \mathbb{F}_9 \) using the lexicographic ordering of elements

shows the subfields of \( \mathbb{F}_{64} \) generated by the polynomial \( X^6 + X^5 + 1 \) whose numerical rank is 97.

The lexicographic ordering has an interesting side-effect for the ordering of elements in extension fields. The elements of the prime subfield are always listed before any other elements in the extension field. For this reason, the addition and multiplication tables of the extension field contain the respective table of the prime field in the upper left corner. For instance, in Figure 3.6, we find the tables for \( \mathbb{F}_3 \) in the upper left corners of the tables of \( \mathbb{F}_9 \), for instance. Recall that omit the zero element in the multiplication tables.

Orbiter uses primitive polynomials for creating extension fields. Because of this, the element \( \alpha \) is always primitive. Since the numerical rank of \( \alpha \) is \( p \), this means that the rank \( p \) always represents a primitive element in an extension field. For the addition and multiplication tables of \( \mathbb{F}_9 \) arranged with respect to powers of a primitive element, see Figure 3.7.
Figure 3.7: Addition and multiplication table of $\mathbb{F}_9$ using the cyclic ordering of elements
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-RREF</td>
<td>m n L</td>
<td>Compute the RREF of the $m \times n$ matrix $L$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td>-nullspace</td>
<td>m n L</td>
<td>Compute a basis for the right nullspace of the $m \times n$ matrix $L$</td>
</tr>
<tr>
<td>-normalize_from_the_right</td>
<td></td>
<td>Normalizes the result of -RREF or nullspace from the right</td>
</tr>
<tr>
<td>-normalize_from_the_left</td>
<td></td>
<td>Normalizes the result of -RREF or nullspace from the left</td>
</tr>
<tr>
<td>-eigenstuff</td>
<td>d M</td>
<td>Computes the eigenvalues and eigenvectors of the given $d \times d$ matrix $M$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td>-all_rational_normal_forms</td>
<td>d</td>
<td>Produces a report of all rational normal forms of endomorphisms of $\mathbb{F}_q^d$</td>
</tr>
</tbody>
</table>

Table 3.11: Finite Field Activities for Linear Algebra

### 3.4 Linear Algebra Over Finite Fields

In Table 3.11, some finite field activities regarding linear algebra are shown. For instance, the command

```
RREF:
▷ $(ORBITER_PATH)orbiter.out\cdot-v.2\·
▷ ▷ -define F\cdot finite_field\cdot-2.2\cdot-end\·
▷ ▷ -define v\cdot vector\cdot-field F\cdot format.2\·
▷ ▷ ▷ -dense:1,1,1,0,1,1,0,0,1\·
▷ ▷ -end\·
▷ ▷ -with F\cdot do\cdot finite_field_activity\·
▷ ▷ -RREF v\cdot normalize_from_the_right\·
▷ ▷ -end
```

computes the RREF form of the matrix

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1
\end{bmatrix}
$$

over $\mathbb{F}_2$. The output is the matrix
The -RREF command produces a latex log of the steps. This can be used to follow the algorithm along. For a somewhat longer example, consider the Vandermonde matrix over the field $\mathbb{F}_7$. Suppose we want to compute the inverse matrix directly. We can use the following command to do so. Notice how we first create the matrix and an identity matrix next to it. After that we apply the -RREF command:

```
V7_VANDERMONDE_EXTENDED="\1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,\1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,2,4,1,2,4,1,0,0,1,0,0,0,0,0,0,0,0,1,3,2,6,4,5,1,0,0,0,1,0,0,0,0,0,0,0,0,1,4,2,1,4,2,1,0,0,0,0,1,0,0,0,0,0,0,0,0,1,5,4,6,2,3,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,6,1,6,1,6,1,0,0,0,0,0,0,0,0,0,0,0,1"_RREF_V7:

$$(\text{ORBITER \textsc{path}})\text{orbiter.out} -v 2$$

```

The following (long) output is produced. Observe how the inverse matrix appears in the second half once the -RREF algorithm is finished:

```
A matrix over the field $\mathbb{F}_7$

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 2 & 4 & 1 & 2 & 4 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 3 & 2 & 6 & 4 & 5 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 4 & 2 & 1 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 5 & 4 & 6 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 6 & 1 & 6 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$
```
Position \((i,j) = (0,0),\) found pivot in column 0

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 2 & 4 & 1 & 2 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 3 & 2 & 6 & 4 & 5 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 4 & 2 & 1 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 5 & 4 & 6 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 6 & 1 & 6 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

After making pivot 1:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 2 & 4 & 1 & 2 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 3 & 2 & 6 & 4 & 5 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 4 & 2 & 1 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 5 & 4 & 6 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 6 & 1 & 6 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 \\
0 & 2 & 4 & 1 & 2 & 4 & 1 & 6 & 0 & 1 & 0 & 0 & 0 \\
0 & 3 & 2 & 6 & 4 & 5 & 1 & 6 & 0 & 0 & 1 & 0 & 0 \\
0 & 4 & 2 & 1 & 4 & 2 & 1 & 6 & 0 & 0 & 0 & 1 & 0 \\
0 & 5 & 4 & 6 & 2 & 3 & 1 & 6 & 0 & 0 & 0 & 0 & 1 \\
0 & 6 & 1 & 6 & 1 & 6 & 1 & 6 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

Position \((i,j) = (1,1),\) found pivot in column 1

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 \\
0 & 2 & 4 & 1 & 2 & 4 & 1 & 6 & 0 & 1 & 0 & 0 & 0 \\
0 & 3 & 2 & 6 & 4 & 5 & 1 & 6 & 0 & 0 & 1 & 0 & 0 \\
0 & 4 & 2 & 1 & 4 & 2 & 1 & 6 & 0 & 0 & 0 & 1 & 0 \\
0 & 5 & 4 & 6 & 2 & 3 & 1 & 6 & 0 & 0 & 0 & 0 & 1 \\
0 & 6 & 1 & 6 & 1 & 6 & 1 & 6 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

After making pivot 1:
After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 4 & 1 & 2 & 4 & 1 & 6 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 2 & 6 & 4 & 5 & 1 & 6 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 4 & 2 & 1 & 4 & 2 & 1 & 6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 5 & 4 & 6 & 2 & 3 & 1 & 6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 6 & 1 & 6 & 1 & 6 & 1 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6 & 0 & 2 & 6 & 1 & 5 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6 & 3 & 1 & 2 & 5 & 2 & 4 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 4 & 0 & 5 & 4 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 6 & 1 & 4 & 5 & 3 & 4 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 2 & 0 & 2 & 0 & 2 & 5 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

Position \((i, j) = (2, 2)\), found pivot in column 2

After making pivot 1:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6 & 0 & 2 & 6 & 1 & 5 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6 & 3 & 1 & 2 & 5 & 2 & 4 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 4 & 0 & 5 & 4 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 6 & 1 & 4 & 5 & 3 & 4 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 2 & 0 & 2 & 0 & 2 & 5 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6 & 3 & 1 & 2 & 5 & 2 & 4 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 4 & 0 & 5 & 4 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 6 & 1 & 4 & 5 & 3 & 4 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 2 & 0 & 2 & 0 & 2 & 5 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

After elimination below pivot:
Position \((i,j) = (3,3)\), found pivot in column 3

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 6 & 1 & 3 & 1 & 6 & 3 & 4 & 1 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 3 & 4 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 4 & 4 & 6 & 6 & 1 & 1 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 5 & 3 & 4 & 3 & 6 & 0 & 0 \\
\end{bmatrix}
\]

After making pivot 1:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 3 & 4 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 4 & 4 & 6 & 6 & 1 & 1 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 5 & 3 & 4 & 3 & 6 & 0 & 0 \\
\end{bmatrix}
\]

After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 3 & 4 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 4 & 4 & 6 & 6 & 1 & 1 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 5 & 3 & 4 & 3 & 6 & 0 & 0 \\
\end{bmatrix}
\]

Position \((i,j) = (4,4)\), found pivot in column 4
After making pivot 1:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 0 & 3 & 2 & 6 & 1 & 3 & 6 & 3 & 1 \\
0 & 0 & 0 & 0 & 1 & 4 & 3 & 4 & 6 & 6 & 4 & 0 & 1 \\
0 & 0 & 0 & 0 & 3 & 1 & 4 & 3 & 6 & 3 & 1 & 0 & 0 & 1
\end{bmatrix}$$

After elimination below pivot:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 1 & 2 & 1 & 5 \\
0 & 0 & 0 & 0 & 1 & 4 & 3 & 4 & 6 & 6 & 4 & 0 & 1 \\
0 & 0 & 0 & 0 & 3 & 1 & 4 & 3 & 6 & 3 & 1 & 0 & 0 & 1
\end{bmatrix}$$

Position \((i, j) = (5, 5)\), found pivot in column 5

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 1 & 2 & 1 & 5 \\
0 & 0 & 0 & 0 & 1 & 1 & 6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 6 & 5 & 2 & 3 & 4 & 5 & 6 & 0 & 1
\end{bmatrix}$$

After making pivot 1:
After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 1 & 2 & 1 & 5 \\
0 & 0 & 0 & 0 & 0 & 1 & 6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 6 & 5 & 2 & 3 & 4 & 5 & 6 \\
\end{bmatrix}
\]

Position \((i,j) = (6,6)\), found pivot in column 6

After making pivot 1:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 1 & 2 & 1 & 5 \\
0 & 0 & 0 & 0 & 0 & 1 & 6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 6 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

After elimination below pivot:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 1 & 2 & 1 & 5 \\
0 & 0 & 0 & 0 & 0 & 1 & 6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination below pivot:
Did not find pivot. The rank of the matrix is 7.

After elimination above pivot 6 in position (6,6):

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 1 & 2 & 1 & 5 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination above pivot 5 in position (5,5):

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 2 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 3 & 0 & 1 & 0 & 0 & 2 & 0 & 3 & 3 & 3 \\
0 & 0 & 0 & 1 & 6 & 4 & 0 & 0 & 3 & 2 & 5 & 6 & 6 \\
0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 3 & 4 & 3 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 6 & 5 & 4 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination above pivot 4 in position (4,4):

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 4 & 0 & 6 & 0 & 3 \\
0 & 0 & 1 & 3 & 0 & 0 & 0 & 0 & 3 & 2 & 6 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 6 & 6 & 1 & 6 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 6 & 3 & 5 & 5 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 6 & 5 & 4 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination above pivot 3 in position (3,3):
The inverse matrix agrees with the output obtained in Section 3.2.

Another task is computing the nullspace of a matrix. The command
nullspace:
$\$(ORBITER\_PATH)orbiter.out:-v.2\$
$\$-define\_F2\_finite\_field\_q.2\$-end$
$\$-define\_v\_vector\_field\_F2\_format.2$
$\$-dense:\$1,1,1,0,1,0,0,1\$-end$
$\$-with\_F2\_do$
$\$-finite\_field\_activity$
$\$-nullspace\_v$
$\$-normalize\_from\_the\_right$
$\$-end$

computes the right nullspace of the matrix from the first example. The output is the matrix

$$
\begin{bmatrix}
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}.
$$

Orbiter can compute eigenvalues and eigenvectors of matrices over finite fields. For instance, the command

eigenstuff:
$\$-v.6$
$\$-define\_F\_finite\_field\_q.5\$-end$
$\$-eigenstuff\_F.4\$"0,1,0,2,0,1,2,1,4,2,3,1,2,0,4,3"

computes all eigenvectors and eigenvalues of the matrix

$$
\begin{bmatrix}
0 & 1 & 0 & 2 \\
0 & 1 & 2 & 1 \\
4 & 2 & 3 & 1 \\
2 & 0 & 4 & 3
\end{bmatrix}
$$

over $\mathbb{F}_5$.

Orbiter can produce a list of all conjugacy classes of endomorphisms of $\mathbb{F}_q^d$ by means of their rational normal forms. For instance

classes\_GL.3.2:
$\$$-v.7\$
produces a list of all conjugacy classes of GL(3, 2). There are 6 of them. The report includes the order of the centralizer and the order of the conjugacy class. The order of the centralizer is computed using Kung’s formula [40]. This command relies on the Orbiter catalogue of irreducible polynomials. For an introduction to the rational normal form of endomorphisms, see [46].

Conjugacy Classes of GL(3, 2)

The number of conjugacy classes of GL(3, 2) is 6:

\[
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix},
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 1
\end{pmatrix},
\begin{pmatrix}
0 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & 1
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{pmatrix}
\]

Class 0 / 6
3, 1, 0
centralizer order 7
class size 24
Class 1 / 6
2, 1, 0
centralizer order 7
class size 24
Class 2 / 6
0, 1, 0; 1, 1, 0

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

centralizer order 3
class size 56
Class 3 / 6
0, 3, 0

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 1 & 1
\end{bmatrix}
\]

centralizer order 4
class size 42
Class 4 / 6
0, 3, 1

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

centralizer order 8
class size 21
Class 5 / 6
0, 3, 2

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

centralizer order 168
class size 1

62
3.5 Advanced Topics in Finite Fields

Let us now look at some advanced topics in the theory of finite fields.

First, in Tables 3.12-3.13, a summary of finite field activities is shown.

A normal basis for a field extension $\mathbb{F}_{q^d}$ over $\mathbb{F}_q$ is a basis of $\mathbb{F}_{q^d}$ as vector space over $\mathbb{F}_q$ which consists of one cycle of the Frobenius automorphism of $\mathbb{F}_{q^d}$ over $\mathbb{F}_q$. For instance, the command

```
normal_basis_2.3:
▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot2\cdot$
▷ ▷ -define:F--finite_field-q\cdot2--end:
▷ ▷ -with:F--do--finite_field_activity:
▷ ▷ -normal_basis_3--end
```

computes a normal basis of $\mathbb{F}_8$ over $\mathbb{F}_2$. Using the polynomial $X^3 + X^2 + 1$, the normal basis in terms of the standard polynomial basis $1, X, X^2, \ldots$ is given by the columns of the matrix

$$
\begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 0 & 1
\end{bmatrix}.
$$

Reading the columns as coefficient vectors with respect to the standard basis, the normal basis is

$$
b_1 = 1 + X + X^2, \quad b_2 = X, \quad b_3 = X^2.
$$

Let us apply the Frobenius mapping $\Phi$ to the elements of the normal bases:

$$
b_1^\Phi = (1 + X + X^2)^2 = 1 + X^2 + X^4 = 1 + X^2 + X^3 + X = 1 + X + X^2 + X^2 + 1 = X = b_2,
$$

$$
b_2^\Phi = X^2 = b_3,
$$

$$
b_3^\Phi = X^4 = X^3 + X = X^2 + X + 1 = b_1.
$$

Thus,

$$
b_1 \mapsto b_2 \mapsto b_3 \mapsto b_1$$

under $\Phi$, as required.

A field is a vector space over any of its subfields. Using a field basis, the elements of the large field can be identified with invertible matrices. So, for $\mathbb{F}_{q^r}$ over $\mathbb{F}_q$, and for $a \in \mathbb{F}_{q^r}$, we consider the $\mathbb{F}_q$-linear map

$$
\mathbb{F}_{q^r} \rightarrow \mathbb{F}_{q^r}, x \mapsto ax.
$$

The following code computes the field reduction from $\mathbb{F}_{64}$ to $\mathbb{F}_8$. Elements in the small field are represented as colors. The $(i, j)$-th block is the matrix of $a = i8 + j$ in the field chosen basis.

63
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-write_code_for_division</td>
<td>fname A B</td>
<td>Write C++ source code for the polynomial division of A by B. See Section 10.4.</td>
</tr>
<tr>
<td>-polynomial_division</td>
<td>A B</td>
<td>Divides polynomial B by polynomial A.</td>
</tr>
<tr>
<td>-extended_gcd_for_polynomials</td>
<td>A B</td>
<td>Computes the extended gcd of polynomials A and B.</td>
</tr>
<tr>
<td>-polynomial_mult_mod</td>
<td>A B M</td>
<td>Computes the product of polynomials A and B modulo the polynomial M.</td>
</tr>
<tr>
<td>-polynomial_power_mod</td>
<td>A N M</td>
<td>Computes the n-th power of the polynomial A modulo the polynomial M.</td>
</tr>
<tr>
<td>-Berlekamp_matrix</td>
<td>A</td>
<td>Compute the Berlekamp matrix associated with the polynomial A.</td>
</tr>
<tr>
<td>-normal_basis</td>
<td>d</td>
<td>Computes a normal basis for $F_{q^d}$ over $F_q$.</td>
</tr>
<tr>
<td>-polynomial_find_roots</td>
<td>A</td>
<td>Computes the roots of the polynomial A.</td>
</tr>
<tr>
<td>-nullspace</td>
<td>A</td>
<td>Computes the right nullspace of the matrix A.</td>
</tr>
<tr>
<td>-RREF</td>
<td>A</td>
<td>Computes the RREF of the matrix A.</td>
</tr>
<tr>
<td>-weight enumerator</td>
<td>A</td>
<td>Computes the weight enumerator of the code whose generator matrix is A.</td>
</tr>
<tr>
<td>-Walsh_Hadamard_transform</td>
<td>fname n</td>
<td>Computes the Walsh-Hadamard transform for the n-variable boolean function in the given file.</td>
</tr>
<tr>
<td>-algebraic_normal_form</td>
<td>fname n</td>
<td>Computes the algebraic normal form for the n-variable boolean function in the given file.</td>
</tr>
<tr>
<td>-apply_trace_function</td>
<td>fname</td>
<td>Applies the absolute trace function to the function in the given file.</td>
</tr>
<tr>
<td>-apply_power_function</td>
<td>fname d</td>
<td>Applies the raise-to-the-power-d function to the function in the given file.</td>
</tr>
<tr>
<td>-identity_function</td>
<td>fname_csv</td>
<td>Creates the identity function and stores in the given csv file.</td>
</tr>
<tr>
<td>-Walsh_matrix</td>
<td>n</td>
<td>Creates the Walsh matrix of order n.</td>
</tr>
</tbody>
</table>

Table 3.12: Finite Field Activities (Part I)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Vandermonde_matrix</td>
<td>$n$</td>
<td>Creates the Vandermonde matrix of order $q \times q$. The entry $(i, j)$ is $x_i^j$ where $w_0, \ldots, x_{q-1}$ is the list of field elements in ordered according to the Orbiter ranks.</td>
</tr>
<tr>
<td>-transversal</td>
<td>$L_1$ $L_2$ $P$</td>
<td>Computes the unique transversal to the lines $L_1$ and $L_2$ through the point $P$ in $\text{PG}(3, q)$. The lines are given by a basis consisting of 8 field elements.</td>
</tr>
<tr>
<td>-intersection_of_two_lines</td>
<td>$L_1$ $L_2$</td>
<td>Computes the intersection of two lines in $\text{PG}(3, q)$. The lines are given by a basis consisting of 8 field elements.</td>
</tr>
<tr>
<td>-rank_point_in_PG</td>
<td>$P$</td>
<td>Computes the orbiter point rank of the point $P$ in $\text{PG}(n, q)$. $P$ is a label of a vector, which is the coefficient vector.</td>
</tr>
<tr>
<td>-unrank_point_in_PG</td>
<td>$r$</td>
<td>Computes the orbiter point in $\text{PG}(n, q)$ from the Orbiter rank value $r$.</td>
</tr>
<tr>
<td>-inverse_isomorphism_klein_quadric</td>
<td>$L36$</td>
<td></td>
</tr>
<tr>
<td>-NTT</td>
<td>$k$ $n$</td>
<td>Computes the Number-theoretic transform for $n = 2^k$, which must divide $q - 1$.</td>
</tr>
</tbody>
</table>

Table 3.13: Finite Field Activities (Part II)
The field reduction from $F_{64}$ to $F_8$ is shown in Figure 3.8. Note that the dimension of the vector space is 2, so the block matrices are $2 \times 2$. Observe that $F_{64}$ has many subfields. Figure 3.9 shows the field reduction from $F_{64}$ to $F_4$ (left) and from $F_{64}$ to $F_2$ (right). Here, the block matrices have size $3 \times 3$ and $6 \times 6$, respectively.

The minimum polynomials associated with the $n$-th roots over $F_q$ can be computed using the `-nth_roots` command, which is a finite field activity. The activity if applied to the field $F_q$ over which the $n$-th roots are defined. The command constructs the filed extension $F_{q^m}$ where $m$ is the order of $q$ modulo $n$. This field extension contains the $n$-th roots of unity. Let $\alpha$ be a primitive element of $F_{q^m}$ and let $\beta$ be a generator of the subgroup of $n$-th roots. Also, let $\gamma$ be the generator of the subgroup of $q - 1$ th roots, which are the elements of the multiplicative group of $F_q$. The output lists the $n$-th roots first, generated by $\beta$. After that,
Figure 3.9: The field reduction from $\mathbb{F}_{64}$ to $\mathbb{F}_4$ (left) and from $\mathbb{F}_{64}$ to $\mathbb{F}_2$ (right)

the $q - 1$th roots are shown, generated by $\gamma$. Finally, a table is produced which shows the irreducible polynomials over $\mathbb{F}_q$ associated with the $n$-th roots of unity. For instance, the following command computes the minimum polynomials of all 21st roots of unity over $\mathbb{F}_8$:

```
F_8_Nth_roots_21:
▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.3\backslash$
▷ ▷ -define:F--finite_field\_q.8--override\_polynomial.11--end\backslash$
▷ ▷ -with:F--do--finite\_field\_activity--nth\_roots.21--end$
▷ pdflatex.Nth\_roots.q8.n21.tex
```

The output is:

Let $\alpha$ be a primitive element of GF(64). Let $\beta$ be a primitive 21-th root in GF(64), so $\beta = \alpha^3$.

$\beta^0 = 100000 = 1$
$\beta^1 = 000100 = \alpha^3$
$\beta^2 = 100001 = \alpha^5 + 1$
$\beta^3 = 111101 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1$
$\beta^4 = 011111 = \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha$
$\beta^5 = 101010 = \alpha^4 + \alpha^2 + 1$
$\beta^6 = 110100 = \alpha^3 + \alpha + 1$
$\beta^7 = 100111 = \alpha^5 + \alpha^4 + \alpha^3 + 1$

67
\[ \beta^8 = 101101 = \alpha^5 + \alpha^3 + \alpha^2 + 1 \]
\[ \beta^9 = 011011 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha \]
\[ \beta^{10} = 011011 = \alpha^5 + \alpha^4 + \alpha^2 + \alpha \]
\[ \beta^{11} = 001011 = \alpha^5 + \alpha^4 + \alpha^2 \]
\[ \beta^{12} = 001001 = \alpha^5 + \alpha^2 \]
\[ \beta^{13} = 111000 = \alpha^2 + \alpha + 1 \]
\[ \beta^{14} = 000111 = \alpha^5 + \alpha^4 + \alpha^2 + \alpha \]
\[ \beta^{15} = 101001 = \alpha^5 + \alpha^2 + 1 \]
\[ \beta^{16} = 111100 = \alpha^3 + \alpha^2 + \alpha + 1 \]
\[ \beta^{17} = 100110 = \alpha^4 + \alpha^3 + 1 \]
\[ \beta^{18} = 010100 = \alpha^3 + \alpha \]
\[ \beta^{19} = 100011 = \alpha^5 + \alpha^4 + 1 \]
\[ \beta^{20} = 001100 = \alpha^3 + \alpha^2 \]

Let \( \gamma \) be a primitive 7-th root in GF(64), so \( \gamma = \alpha^9 \).
\[ \gamma^0 = 100000 = 1 \]
\[ \gamma^1 = 111001 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1 \]
\[ \gamma^2 = 110100 = \alpha^3 + \alpha + 1 \]
\[ \gamma^3 = 011001 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha \]
\[ \gamma^4 = 001011 = \alpha^5 + \alpha^2 \]
\[ \gamma^5 = 101001 = \alpha^5 + \alpha^2 + 1 \]
\[ \gamma^6 = 010100 = \alpha^3 + \alpha \]

The \( q \)-cyclotomic set for \( q = 8 \) are:

\[
\begin{array}{c}
\{0\} \\
\{1, 8\} \\
\{2, 16\} \\
\{3\} \\
\{4, 11\} \\
\{5, 19\} \\
\{6\} \\
\{7, 14\} \\
\{9\} \\
\{10, 17\} \\
\{12\} \\
\{13, 20\} \\
\{15\} \\
\{18\}
\end{array}
\]
Subfield basis, a basis for GF(8) inside GF(64):

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

The irreducible polynomials associated with the 21-th roots over GF(8) are:

<table>
<thead>
<tr>
<th>i</th>
<th>( r_i )</th>
<th>( \text{Cyc}(r_i) )</th>
<th>( m_{\beta_i}(X) )</th>
<th>( m_{\beta_i}(X) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(0)</td>
<td>((100000)X^0 + (100000)X^1)</td>
<td>(X + 1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(1, 8)</td>
<td>((011101)X^0 + (101001)X^1 + (100000)X^2)</td>
<td>(X^2 + 7X + 3)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>(2, 16)</td>
<td>((010100)X^0 + (011101)X^1 + (100000)X^2)</td>
<td>(X^2 + 3X + 5)</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>(3)</td>
<td>((111101)X^0 + (100000)X^1)</td>
<td>(X + 2)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>(4, 11)</td>
<td>((101001)X^0 + (010100)X^1 + (100000)X^2)</td>
<td>(X^2 + 5X + 7)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>(5, 19)</td>
<td>((111101)X^0 + (001001)X^1 + (100000)X^2)</td>
<td>(X^2 + 6X + 2)</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>(6)</td>
<td>((110100)X^0 + (100000)X^1)</td>
<td>(X + 4)</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>(7, 14)</td>
<td>((100000)X^0 + (100000)X^1 + (100000)X^2)</td>
<td>(X^2 + X + 1)</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>(9)</td>
<td>((011101)X^0 + (100000)X^1)</td>
<td>(X + 3)</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>(10, 17)</td>
<td>((110100)X^0 + (111101)X^1 + (100000)X^2)</td>
<td>(X^2 + 2X + 4)</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>(12)</td>
<td>((001001)X^0 + (100000)X^1)</td>
<td>(X + 6)</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>(13, 20)</td>
<td>((001001)X^0 + (110100)X^1 + (100000)X^2)</td>
<td>(X^2 + 4X + 6)</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>(15)</td>
<td>((101001)X^0 + (100000)X^1)</td>
<td>(X + 7)</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>(18)</td>
<td>((010100)X^0 + (100000)X^1)</td>
<td>(X + 5)</td>
</tr>
</tbody>
</table>

In Section 3.2, we considered the Vandermonde matrix over \(\mathbb{F}_7\). Let us do the same for the field \(\mathbb{F}_8\) instead. We use the following command:

\[
\text{\textbf{F}}_{8}.\text{vandermonde:}
\]

\[
\begin{align*}
&\text{\texttt{\$(ORBITER\_PATH)orber.out\,-v\,-3\,\backslash\text{\textbackslash\}}} \\
&\text{\texttt{\textbackslash}}^{-\text{\texttt{define}}:\text{\texttt{\textbackslash-\text{\texttt{finite}}:\text{\texttt{field}}:\text{\texttt{-q}}:\text{\texttt{8}}:\text{\texttt{-end}}}\text{\texttt{\textbackslash}}} \\
&\text{\texttt{\textbackslash}}^{-\text{\texttt{with}}:\text{\texttt{\textbackslash-\text{\texttt{do}}:\text{\texttt{finite}}:\text{\texttt{field}}:\text{\texttt{activity}}}\text{\texttt{\textbackslash}}} \\
&\text{\texttt{\textbackslash}}^{-\text{\texttt{Vandermonde}}:\text{\texttt{matrix}}}\text{\texttt{\textbackslash}}
\end{align*}
\]
The output is shown below. Again, the first matrix is $V = (x_i^j)$. The second matrix is $V^{-1}$:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 4 & 5 & 7 & 3 & 6 & 1 \\
1 & 3 & 5 & 2 & 6 & 7 & 4 & 1 \\
1 & 4 & 7 & 6 & 2 & 5 & 3 & 1 \\
1 & 5 & 6 & 4 & 3 & 2 & 7 & 1 \\
1 & 6 & 3 & 7 & 5 & 4 & 2 & 1 \\
1 & 7 & 2 & 3 & 4 & 6 & 5 & 1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 6 & 4 & 3 & 7 & 2 & 5 \\
0 & 1 & 3 & 7 & 5 & 2 & 4 & 6 \\
0 & 1 & 7 & 6 & 2 & 3 & 5 & 4 \\
0 & 1 & 5 & 2 & 6 & 4 & 7 & 3 \\
0 & 1 & 4 & 5 & 7 & 6 & 3 & 2 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

Let us now do a somewhat larger example of the same problem. The next command computes the Vandermonde matrix and its inverse over the field $\mathbb{F}_{1024}$:

\[
\text{F}_1024\_\text{vandermonde}: \\
\text{\$\{ORBITER\_PATH\}\orbiter\_out:\-v\:\-3\:}\\
\text{\:\-define\:\text{F}\:\text{-\finite\_field\:-q\:1024\:-end\:\}}\\
\text{\:\-with\:\text{F}\:\text{-do\:\text{-finite\_field\_activity\:\}}\\
\text{\:\text{-Vandermonde\_matrix\:\}}\\
\text{\:\text{-end\}}\\
\text{\text{\text{\text{rm\:Vandermonde\_1024\_csv\}}\\
\text{\text{\text{\text{rm\:Vandermonde\_inv\_1024\_csv\}}}}\]
\]

This command takes a bit of time to execute. The matrix is not shown. It would be too big to be printed. In order to save disc space, we delete the output files, using the \texttt{rm} command.

Orbiter can create code for the number theoretic transform. This is the discrete Fourier transform performed over finite fields. The generated code can be compiled with the Orbiter library. Because compiling code is a bit more complicated, additional makefile options are necessary. Suppose we want to create the number theoretic transform for the 16th roots of unity inside the field $\mathbb{F}_{17}$. Here is the command to generate the Orbiter source code:

\[
\text{NTT\_k4\_q17\_cpp}: \\
\text{\$\{ORBITER\_PATH\}\orbiter\_out:\-v\:\-3\:}\\
\text{\:\-define\:\text{F}\:\text{-\finite\_field\:-q\:17\:-end\:\}}\\
\text{\:\-with\:\text{F}\:\text{-do\:\text{-finite\_field\_activity\:-\texttt{NTT\_4\_17\:-end}}}}}\]

70
This produces a C++ file `NTT_k4_q17.cpp`. This file should be compiled and linked against the Orbiter library. Because of this, we use the following makefile variables.

```
SRC=$(MY_PATH)/src
MY_CPP=g++
MY_CC=gcc
CPPFLAGS=-Wall-I./../DEV.22/orbiter/src/lib-std=c++14
LIB=$(SRC)/lib/liborbiter.a-lpthread
LFLAGS=-lm-Wl,-rpath-Wl,/usr/local/gcc-8.2.0/lib64
```

The command

```
F_17_NTT_compile::NTT_k4_q17.cpp
 ▷ $(MY_CPP)·NTT_k4_q17.cpp$(CPPFLAGS)\
 ▷ ▷ $(LIB)$LFLAGS-o NT Teh.pickle.out
 ▷ ./NTT_k4_q17.out
```

can be used to compile the code and run it. Note the dependency on the file `NTT_k4_q17.cpp`. This means that `make` would automatically invoke the first command if only the second one was issued.
3.6 Basic Ring Theory

Orbiter can deal with multivariate polynomial rings with coefficients over finite fields. Orbiter creates the homogenous components only (so it is technically not a ring).

The following command creates the homogeneous component of degree 3 in a polynomial ring in 4 variables. The variables are named. They are \( x_0, x_1, x_2, x_3 \). Note that two sets of names are defined using the \(-variables\) command. The first is the labels for regular text output. The second is the set of names for latex output. Here is the command:

```plaintext
Polynomial ring:
▷ $(ORBITER_PATH)orbiter.out-v.3:\n▷ ▷ -define:F:-finite_field:-q:4:-end:\n▷ ▷ -define:R:-polynomial_ring:-field:F:\n▷ ▷ ▷ -number_of_variables:4\n▷ ▷ ▷ -homogeneous_of_degree:3\n▷ ▷ ▷ -variables:"x0,x1,x2,x3":"x_0,x_1,x_2,x_3"\n▷ ▷ ▷ -end\n```

For more on rings, see Chapter 8.
Chapter 4

Geometry

4.1 Finite Projective Spaces

Orbiter can create the projective space $\text{PG}(n, q)$. In order to do so, an object of type projective_space needs to be define. Once the object exists, various commands are available. Let us look at a very simple example. Suppose we want to create $\text{PG}(3, 2)$. The following command sequence first creates the finite field $\mathbb{F}_2$. The symbol $F$ is used to store the field. After that, the projective space $\text{PG}(3, F)$ is created and stored in the symbol $P$. No other commands are given:

```
PG_3_2_easy:
  $\$(ORBITER_PATH)orbiter.out$-v.3$
  -define $F$--finite_field--q.2--end$
  -define $P$--projective_space.3$F$--end$
```

This means that Orbiter offers indexing for the subspaces of $\text{PG}(n, q)$ of a fixed dimension. For instance, there are enumerators for points and lines. Besides these, there are enumerators for subspaces of any dimension. The incidence matrix between points and lines with respect to this ordering can be computed. The indexing is used to establish the permutation representations of the projective group, as will be described in Section 5.2. The indexing of points is not the lexicographic ordering. It emphasizes the role of frames in the geometry by assigning the smallest rank values to the members of the standard frame. After that, the other points are listed.

Orbiter can create cheat sheets, which summarize the properties of $\text{PG}(n, q)$ and list the various elements. The following command creates a cheat sheet for $\text{PG}(2, 4)$ using a finite field object:

```
P_2_4:
  $\$(ORBITER_PATH)orbiter.out$-v.2$
  -define $F$--finite_field--q.4--end$
  -define $P$--projective_space.2$F$--end$
```
The cheat sheet contains a drawing of the plane as shown in Figure 4.1. The affine plane is shown in the cartesian plane, while the line at infinity is wrapped around the top right corner. The cheat sheet continues by listing the points, including the canonical Baer subgeometry PG(2, 2). After that, the points are listed again, but with left-normalized vectors. Finally, the lines are shown.

PG(2, 4) has 21 points:
$P_0 = (1, 0, 0) = (1, 0, 0)$ 
$P_1 = (0, 1, 0) = (0, 1, 0)$ 
$P_2 = (0, 0, 1) = (0, 0, 1)$ 
$P_3 = (1, 1, 1) = (1, 1, 1)$ 
$P_4 = (1, 1, 0) = (1, 1, 0)$ 
$P_5 = (2, 1, 0) = (\alpha, 1, 0)$ 
$P_6 = (3, 1, 0) = (\alpha^2, 1, 0)$ 
$P_7 = (1, 0, 1) = (1, 0, 1)$ 
$P_8 = (2, 0, 1) = (\alpha, 0, 1)$ 
$P_9 = (3, 0, 1) = (\alpha^2, 0, 1)$ 
$P_{10} = (0, 1, 1) = (0, 1, 1)$ 

$P_{11} = (2, 1, 1) = (\alpha, 1, 1)$ 
$P_{12} = (3, 1, 1) = (\alpha^2, 1, 1)$ 
$P_{13} = (0, 2, 1) = (0, \alpha, 1)$ 
$P_{14} = (1, 2, 1) = (1, \alpha, 1)$ 
$P_{15} = (2, 2, 1) = (\alpha, \alpha, 1)$ 
$P_{16} = (3, 2, 1) = (\alpha^2, \alpha, 1)$ 
$P_{17} = (0, 3, 1) = (0, \alpha^2, 1)$ 
$P_{18} = (1, 3, 1) = (1, \alpha^2, 1)$ 
$P_{19} = (2, 3, 1) = (\alpha, \alpha^2, 1)$ 
$P_{20} = (3, 3, 1) = (\alpha^2, \alpha^2, 1)$ 

Baer subgeometry:

$P_0 = (1, 0, 0)$ 
$P_1 = (0, 1, 0)$ 
$P_2 = (0, 0, 1)$ 
$P_3 = (1, 1, 1)$ 
$P_4 = (1, 1, 0)$ 
$P_5 = (2, 1, 0)$ 
$P_6 = (3, 1, 0)$ 
$P_7 = (1, 0, 1)$ 
$P_8 = (2, 0, 1)$ 
$P_9 = (3, 0, 1)$ 
$P_{10} = (0, 1, 1)$ 

$P_{11} = (2, 1, 1)$ 
$P_{12} = (3, 1, 1)$ 
$P_{13} = (0, 2, 1)$ 
$P_{14} = (1, 2, 1)$ 
$P_{15} = (2, 2, 1)$ 
$P_{16} = (3, 2, 1)$ 
$P_{17} = (0, 3, 1)$ 
$P_{18} = (1, 3, 1)$ 
$P_{19} = (2, 3, 1)$ 
$P_{20} = (3, 3, 1)$ 

There are 7 elements in the Baer subgeometry. 
Normalized from the left:

$P_0 = (1, 0, 0)$ 
$P_1 = (0, 1, 0)$ 
$P_2 = (0, 0, 1)$ 
$P_3 = (1, 1, 1)$ 
$P_4 = (1, 1, 0)$ 
$P_5 = (2, 1, 0)$ 
$P_6 = (3, 1, 0)$ 
$P_7 = (1, 0, 1)$ 
$P_8 = (2, 0, 1)$ 
$P_9 = (3, 0, 1)$ 
$P_{10} = (0, 1, 1)$ 

$P_{11} = (2, 1, 1)$ 
$P_{12} = (3, 1, 1)$ 
$P_{13} = (0, 2, 1)$ 
$P_{14} = (1, 2, 1)$ 
$P_{15} = (2, 2, 1)$ 
$P_{16} = (3, 2, 1)$ 
$P_{17} = (0, 3, 1)$ 
$P_{18} = (1, 3, 1)$ 
$P_{19} = (2, 3, 1)$ 
$P_{20} = (3, 3, 1)$ 

The Lines of PG(2, 4). PG(2, 4) has 21 1-subspaces:
Here is a slightly larger example. The following command creates a cheat sheet for $\text{PG}(3, 2)$.

$\text{PG}_3.2$
\begin{verbatim}
▷ $\$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot0\cdot$
▷ ▷ -define F::finite_field::q\cdot2::end::
▷ ▷ -define P::projective_space::3\cdotF::end::
▷ ▷ -with P::do::projective_space_activity::
▷ ▷ ▷ -cheat_sheet::
▷ ▷ -end
▷ pdflatex PG_3.2.tex
▷ open PG_3.2.pdf
\end{verbatim}

The cheat sheet shows points, lines and planes. The lines are shown together with their Plücker coordinates. The lines whose Plücker coordinates are unit vectors are shown separately.

**The projective space $\text{PG}(3, 2)$**

$q = 2$

$p = 2$

$e = 1$

$n = 3$

Number of points = 15
The points of PG(3, 2)

PG(3, 2) has 15 points:

\[
\begin{align*}
P_0 &= (1,0,0,0) & P_1 &= (0,1,0,0) & P_2 &= (0,0,1,0) & P_3 &= (0,0,0,1) \\
P_4 &= (1,1,1,1) & P_5 &= (1,1,0,0) & P_6 &= (1,0,1,0) & P_7 &= (0,1,1,0) \\
P_8 &= (1,1,1,0) & P_9 &= (1,0,0,1) & P_{10} &= (0,1,0,1) & P_{11} &= (1,1,0,1) \\
P_{12} &= (0,0,1,1) & P_{13} &= (1,0,1,1) & P_{14} &= (0,1,1,1) & \\
\end{align*}
\]

Normalized from the left:

\[
\begin{align*}
P_0 &= (1,0,0,0) & P_1 &= (0,1,0,0) & P_2 &= (0,0,1,0) & P_3 &= (0,0,0,1) \\
P_4 &= (1,1,1,1) & P_5 &= (1,1,0,0) & P_6 &= (1,0,1,0) & P_7 &= (0,1,1,0) \\
P_8 &= (1,1,1,0) & P_9 &= (1,0,0,1) & P_{10} &= (0,1,0,1) & P_{11} &= (1,1,0,1) \\
P_{12} &= (0,0,1,1) & P_{13} &= (1,0,1,1) & P_{14} &= (0,1,1,1) & \\
\end{align*}
\]

The lines of PG(3, 2)

PG(3, 2) has 35 1-subspaces:

\[
\begin{align*}
L_0 &= \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} = \text{Pl}(1,0,0,0,0,0) \\
L_1 &= \begin{bmatrix} 1000 \\ 0110 \end{bmatrix} = \text{Pl}(1,0,1,0,0,0) \\
L_2 &= \begin{bmatrix} 1000 \\ 0101 \end{bmatrix} = \text{Pl}(1,0,0,0,1,0) \\
L_3 &= \begin{bmatrix} 1000 \\ 0111 \end{bmatrix} = \text{Pl}(1,0,1,0,1,0) \\
L_4 &= \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} = \text{Pl}(0,0,1,0,0,0) \\
L_5 &= \begin{bmatrix} 1000 \\ 0011 \end{bmatrix} = \text{Pl}(0,0,1,0,1,0) \\
\end{align*}
\]
\[ L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} = \Pi(0, 1, 0, 0, 0) \]

Lines sorted by Pluecker coordinates

\[ 0 = \Pi(1, 0, 0, 0, 0, 0) = L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \]
\[ 1 = \Pi(0, 1, 0, 0, 0, 0) = L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} \]
\[ 2 = \Pi(0, 0, 1, 0, 0, 0) = L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} \]
\[ 3 = \Pi(0, 0, 0, 1, 0, 0) = L_{30} = \begin{bmatrix} 0100 \\ 0001 \end{bmatrix} \]
\[ 4 = \Pi(0, 0, 0, 0, 1, 0) = L_6 = \begin{bmatrix} 1000 \\ 0001 \end{bmatrix} \]
\[ 5 = \Pi(0, 0, 0, 0, 0, 1) = L_{28} = \begin{bmatrix} 0100 \\ 0010 \end{bmatrix} \]
\[ \vdots \]
\[ 34 = \Pi(0, 1, 1, 1, 1, 1) = L_{26} = \begin{bmatrix} 1101 \\ 0011 \end{bmatrix} \]

PG(3,2) has the following low weight Pluecker lines:

\[ L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} = \Pi(1, 0, 0, 0, 0, 0) \]
\[ L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} = \Pi(0, 0, 1, 0, 0, 0) \]
\[ L_6 = \begin{bmatrix} 1000 \\ 0001 \end{bmatrix} = \Pi(0, 0, 0, 1, 0) \]
\[ L_{28} = \begin{bmatrix} 0100 \\ 0010 \end{bmatrix} = \Pi(0, 0, 0, 0, 1) \]
\[ L_{30} = \begin{bmatrix} 0100 \\ 0001 \end{bmatrix} = \Pi(0, 0, 0, 1, 0) \]
\[ L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} = \Pi(0, 1, 0, 0, 0) \]
The planes of PG(3, 2)

PG(3, 2) has 15 2-subspaces:

\[ L_0 = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \end{bmatrix} \]

\[ L_1 = \begin{bmatrix} 1000 \\ 0100 \\ 0011 \end{bmatrix} \]

\[ \vdots \]

\[ L_{14} = \begin{bmatrix} 0100 \\ 0010 \\ 0001 \end{bmatrix} \]

The polynomial rings associated with PG(3, 2)

<table>
<thead>
<tr>
<th>( h )</th>
<th>monomial</th>
<th>vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( X_0 )</td>
<td>(1, 0, 0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>( X_1 )</td>
<td>(0, 1, 0, 0)</td>
</tr>
<tr>
<td>2</td>
<td>( X_2 )</td>
<td>(0, 0, 1, 0)</td>
</tr>
<tr>
<td>3</td>
<td>( X_3 )</td>
<td>(0, 0, 0, 1)</td>
</tr>
</tbody>
</table>
4.2 Indexing Points and Lines

The enumerator for points establishes a bijection between the set of points and the integers on the interval $[0, \theta_n(q) - 1]$, where

$$\theta_n(q) = \frac{q^{n+1} - 1}{q - 1}.$$ 

In order to facilitate the bijection, Orbiter enumerates representative vectors for the one-dimensional subspaces. The conditions on the vectors are summarized below:

1. The vector is not the zero vector.
2. The rightmost nonzero entry in the vector is one. If it is not, we normalize the vector so that the rightmost nonzero vector is indeed one. This operation does not change the projective point which is associated with the vector.

The second condition ensures that we list each projective point exactly once. We require two functions, \textsc{Rank} and \textsc{Unrank}. The function \textsc{Rank} takes a vector $x \in \mathbb{F}_q^n$, not zero, and maps it to the element in $\mathbb{Z}_N$ representing the projective point $P(x)$. A frame in $\text{PG}(n, q)$ is a set of $n + 2$ points, no $n + 1$ in a hyperplane. We assume that the coordinates of a vector are indexed by the elements of $\mathbb{Z}_n$. Also, we let $e_i$ be the $i$-th unit vector. A frame for $\text{PG}(n, q)$ is

$$e_0, \ldots, e_{n-1}, e_0 + \cdots + e_{n-1}.$$ 

This is the standard frame. We start the labeling of points with the standard frame. After these $n + 2$ points, we list the remaining points in lexicographic ordering (utilizing right-normalized representative). Thus, for $\text{PG}(2, 2)$ the ordering is

$$(1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 1), (1, 1, 0), (1, 0, 1), (0, 1, 1).$$ 

Let us describe the two functions rank and unrank to perform the actual mappings between $\text{PG}(n, q)$ and $\mathbb{Z}_N$, where $N = \theta_n(q)$. For this, assume that ranking and unranking functions have already been defined for the elements of the finite field $\mathbb{F}_q$. Thus, we assume that for $x \in \mathbb{F}_q$, \textsc{Rank}($\mathbb{F}_q, x$) is a number $b$ in $\mathbb{Z}_q$. Also, for $b \in \mathbb{Z}_q$, we assume that \textsc{Unrank}($\mathbb{F}_q, b$) is the corresponding $x \in \mathbb{F}_q$. So, we assume that \textsc{Rank} and \textsc{Unrank} are mutually inverse functions. Consider the group $\text{PGL}(3, 2)$ acting on $\text{PG}(2, 2)$, for instance. The points of $\text{PG}(2, 2)$ are listed in 4.1.

Let us look at an example. The following command computes the rank of $P(3, 3, 1) = P(\omega + 1, \omega + 1, 1)$

in $\text{PG}(2, 4)$:

```
PG_2_4_rank_point:
▷ $(\text{ORBITER\_PATH})\text{or}biter\_ou}t\cdot-v\cdot2\$
▷ ▷ -define.F:-finite_field:-q\cdot4:-end\$
▷ ▷ -with.F:-do:-finite_field_activity\$
▷ ▷ ▷ -rank_point_in_PG\cdot2\cdot"3,3,1"\cdot-end
```

80
Algorithm 1 Rank

1: procedure Rank(vector : x, field : \( \mathbb{F}_q \), int : n)
2:     assert x is a nonzero vector in \( \mathbb{F}_q^n \).
3:     if x = e, then
4:         return i
5:     if x = 1 then
6:         return n
7:     i ← max\{j ∈ \mathbb{Z}_n | x_j \neq 0\}
8:     x ← \frac{1}{x_i} x
9:     a := 0
10:    for j = i − 1, . . . , 1, 0 do
11:        a ← a + Rank(\mathbb{F}_q, x_j)
12:        if j > 0 then
13:            a ← a · q
14:    if i = n − 1 and a ≥ \sum_{j=0}^{i−1} q^j then
15:        a ← a − 1
16:    a ← a + n − i + \sum_{j=0}^{i−1} q^j
17:    return a

\( a = \text{Rank}(x) \) | \( x = \text{Unrank}(a) \) \\
--- | ---
0 | (1, 0, 0)
1 | (0, 1, 0)
2 | (0, 0, 1)
3 | (1, 1, 1)
4 | (1, 1, 0)
5 | (1, 0, 1)
6 | (0, 1, 1)

Table 4.1: Representatives of the points of PG(2, 2)
Algorithm 2 Unrank

1: procedure Unrank(int : \(a\), field : \(\mathbb{F}_q\), int : \(n\))
2:     assert \(a \in \mathbb{Z}_N\) where \(N = \theta_{n-1}(q)\).
3: if \(a < n\) then
4:     return \(e_a\)
5: \(a \leftarrow a - n\)
6: if \(a = 0\) then
7:     return 1
8: \(a \leftarrow a - 1\)
9: \(x \leftarrow 0\)
10: for \(i = 1, \ldots, n - 1\) do
11:     if \(a \geq \sum_{j=1}^{i-1} q^j\) then
12:         \(a \leftarrow a - \sum_{j=1}^{i-1} q^j\)
13:     else
14:         \(x_i \leftarrow 1\)
15:     break
16: for \(k = i + 1, \ldots, n - 1\) do
17:     \(x_k \leftarrow 0\)
18: \(a \leftarrow a + 1\)
19: if \(i = n - 1\) and \(a \geq \sum_{j=0}^{i-1} q^j\) then
20:     \(a \leftarrow a + 1\)
21: \(j \leftarrow 0\)
22: while \(a > 0\) do
23:     \(r \leftarrow a \mod q\)
24:     \(x_j \leftarrow Unrank(\mathbb{F}_q, r)\)
25:     \(j \leftarrow j + 1\)
26:     \(a \leftarrow (a - r)/q\)
27: for \(h = j, \ldots, i - 1\) do
28:     \(x_h \leftarrow 0\)
29: return \(x\)
The rank turns out to be 20.

It is possible to export the incidence matrix of a projective space to a file. The following example creates PG(2, 8) and exports the incidence matrix to a csv file. After that, a graphical representation is produced.

PG_2_8_incidence_matrix:
   $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2\$
   $\cdot-define\cdot-F\cdot-\text{finite\_field}\cdot-q.8\cdot-end\cdot$
   $\cdot-define\cdot-P\cdot-\text{projective\_space}\cdot-2\cdot-F\cdot-end\cdot$
   $\cdot-\text{with}\cdot-P\cdot-\text{do}\cdot-\text{projective\_space\_activity}\cdot$
   $\cdot-\text{export\_point\_line\_incidence\_matrix}\cdot$
   $\cdot-end$
   $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2\$
   $\cdot-define\cdot\text{all\_one}\cdot-\text{vector}\cdot-\text{repeat}\cdot-1.73\cdot-end\cdot$
   $\cdot-draw\cdot\text{matrix}\cdot$
   $\cdot-\text{input\_csv\_file}\cdot\text{PG\_n2\_q8\_incidence\_matrix.csv}\cdot$
   $\cdot-box\cdot\text{width}\cdot-20\cdot-\text{bit\_depth}\cdot-8\cdot$
   $\cdot-partition\cdot-3\cdot$
   $\cdot-all\cdot\text{one}\cdot-all\cdot\text{one}\cdot$
   $\cdot-end$
   open\cdot\text{PG\_n2\_q8\_incidence\_matrix\_draw.bmp}

The incidence matrix is shown in Figure 4.2. The rows and columns correspond to points and lines, respectively. The Orbiter indexing of points and lines determines the ordering of rows and columns.
Figure 4.2: Incidence matrix of PG(2, 8) in Orbiter ordering
4.3 Finite Desarguesian Projective Planes

The projective spaces $\text{PG}(2, q)$ deserve special attention. They are examples of a more general structure called projective planes. The $\text{PG}(2, F)$, $F$ a field, are distinguished in the class of projective planes by the fact that the theorem of Desargues always holds. They are called the desarguesian projective planes. For other projective planes, see Section 12.2.

The points in the desarguesian projective plane $\text{PG}(2, q)$ have the coordinates $P(x, y, z)$, with $x, y, z \in F_q$. We can distinguish one line, for instance $z = 0$, and call it the line at infinity. The points not on that line form an affine plane $\text{AG}(2, q)$.

The command

```
PG_2_16:
```

```
$\text{(ORBITER_PATH)}$orbiter.out:
```

```
\text{-draw_options-xin.20000-yin.20000-}
```

```
\text{-radius.200-line_width.0.3-nodes_empty-end-}
```

```
\text{-define-F-finite_field-q.16-end-}
```

```
\text{-define-P-projective_space-2-F-end-}
```

```
\text{-with-P-do-projective_space_activity-}
```

```
\text{-cheat_sheet-}
```

```
\text{-end}
```

```
pdflatex PG_2_16.tex
```

```
open PG_2_16.pdf
```

produces the drawing of $\text{PG}(2, 16)$ shown in Figure 4.3. The \text{-nodes_empty} command is used to suppress the drawing of the nodes. The \text{-xin 20000} and \text{-yin 20000} options double the input coordinate system (recall from Table 16.2 that the default values are 10000), which has the effect that the text appears smaller relative to the grid.

Projective spaces has a special property. They admit a cyclic group action on points and hyperplanes. Such a group is often called a Singer cycle. It is generated from a projectivity defined by the companion matrix of an irreducible polynomial. Let us look at an example. The following command creates a Singer cycle of $\text{PG}(2, 4)$

```
PG_2_4_with_decomposition:
```

```
$\text{(ORBITER_PATH)}$orbiter.out-v.2:
```

```
\text{-define-F-finite_field-q.4-end-}
```

```
\text{-define-P-projective_space-2-F-end-}
```

```
\text{-with-P-do-projective_space_activity-}
```

```
\text{-cheat_sheet_for_decomposition_by_element_PG-}
```

```
1:0.0.0.1,2,1,1,0
```

```
"PG_2_4_singer-}
```

```
\text{-end}
```

```
pdflatex PG_2_4_singer.tex
```

```
open PG_2_4_singer.pdf
```

85
Figure 4.3: The plane PG(2, 16)

The output is shown below:

Considering the cyclic group generated by

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\omega & 1 & 1
\end{bmatrix}_0 =
\begin{bmatrix}
010 \\
001 \\
211
\end{bmatrix}_0
\]

The group is transitive on points and on lines.

Orbits on points:
There are 1 orbits, the orbit lengths are 21

Orbits on lines:
There are 1 orbits, the orbit lengths are 21

Fixed points:
Fixed lines:
Row scheme:

\[
\begin{array}{c|c}
\rightarrow & 21 \\
21 & 5
\end{array}
\]

Column scheme:
The command produces a csv file containing the cyclic incidence matrix, which can be visualized using the following command:

```bash
PG_2.4_incma_cyclic:
$ (ORBITER_PATH)orbiter.out --v 4
 -define R -vector -repeat 1 21 -end
 -define C -vector -repeat 1 21 -end
 -draw_matrix
 -input_csv_file PG_2.4_singer_incma_cyclic.csv
 -box_width 40 -bit_depth 24
 -partition 3 R C
 -end
open PG_2.4_singer_incma_cyclic_draw.bmp
```

The command produces a csv file containing the cyclic incidence matrix, which can be visualized using the following command:

The cyclic incidence matrix is shown in Figure 4.4.
If the number of points is not a prime, the group acts imprimitively. By considering various subgroups, tactical decompositions are created. For instance, for PG(2, 4), with 21 points, we can consider a subgroup the Singer cycle of order 3, which induces a partition with 7 classes of size 3 on both points and lines:

```
PG_2_4_incma_singer_sub_3:
  $(ORBITER_PATH)orbiter.out.-v.4.
  -list_arguments.
  -define.R.-vector.-repeat.3.7.-end.
  -define.C.-vector.-repeat.3.7.-end.
  -draw_matrix.
  -input_csv_file.PG_2_4_singer_incma_subgroup_index_3.csv.
  -box_width.40.-bit_depth.24.
  -partition.3.R.C.
  -end
  open.PG_2_4_singer_incma_subgroup_index_3.draw.bmp
```

The tactical decomposition of the incidence matrix is shown in Figure 4.5.
4.4 The Grassmannian

Let $V$ be a finite dimensional vector space and let $G_r^k(V)$ be the Grassmannian of $k$-dimensional subspaces of $V$. If $\dim(V) = n$, the notation $G_r^k_n$ is used for $G_r^k(V)$. If $V = \mathbb{F}_q^n$, the notation $G_r^k_{n,q}$ is used for $G_r^k(V)$. The order of the set $G_r^k_{n,q}$ can be computed as

$$\left[ \begin{array}{c} n \\ k \end{array} \right]_q = \prod_{i=0}^{k-1} \frac{q^{n-i} - 1}{q^{k-i} - 1},$$

using the $q$-binomial coefficient.

Orbiter has an enumerator for the Grassmannian. The purpose of this enumerator is to establish a bijection between the Grassmannian and the integers in the interval $[0, N - 1]$, where $N = \left[ \begin{array}{c} n \\ k \end{array} \right]_q$. In order to do so, Orbiter picks a basis for each subspace. By writing the elements of the basis in the rows of a matrix, a $k \times n$ matrix is obtained. In order to make the matrix unique, we assume it to be in RREF. In coding theory, such a matrix is called a generator matrix.

The Orbiter cheat sheets for PG($n,q$) (see Section 4.1) contain lists of all Grassmannians, provided they are not too big. It is also possible to create cheat sheets specifically for one Grassmannian. For instance, the command

```
GR_3_2_2:
  > $(\text{ORBITER\_PATH})\text{orbiter.out}\$
  > > -define \text{F: finite field q:2: end} \$
  > > -with \text{F: do finite field activity} \$
  > > > -cheat_sheet_Gr.3.2.end
  > pdf\text{latex}Gr.3.2.2.tex
  > open Gr.3.2.2.pdf
```

produces a list of 2-dimensional subspaces of $\mathbb{F}_2^3$, i.e. the lines of PG(2, 2):

$$L_0 = \begin{bmatrix} 100 \\ 010 \end{bmatrix}$$
$$L_1 = \begin{bmatrix} 100 \\ 011 \end{bmatrix}$$
$$L_2 = \begin{bmatrix} 100 \\ 001 \end{bmatrix}$$
$$L_3 = \begin{bmatrix} 101 \\ 010 \end{bmatrix}$$
$$L_4 = \begin{bmatrix} 101 \\ 011 \end{bmatrix}$$
$$L_5 = \begin{bmatrix} 110 \\ 001 \end{bmatrix}$$
$$L_6 = \begin{bmatrix} 010 \\ 001 \end{bmatrix}$$

The following command illustrates how to rank lines. In the example, we consider lines in PG(3, 3). The lines are given as vectors of length 8. Three lines are given in v1 and three lines are given in v2.
rank_lines:
  ▶ $(\text{ORBITER\_PATH})\text{orbiner.out} -v.2\$
  ▶ ▶ -define\_v1\_vector\_format.3\$
  ▶ ▶ ▶ -dense "1,0,2,2,0,1,1,2,1,0,2,0,0,1,1,2,1,0,2,2,0,1,2,1"\$
  ▶ ▶ ▶ -end\$
  ▶ ▶ -define\_v2\_vector\_format.3\$
  ▶ ▶ ▶ -dense "1,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,1,0,0,0,2,1"\$
  ▶ ▶ ▶ -end\$
  ▶ ▶ -define\_F\_finite\_field\_q.3\_end\$
  ▶ ▶ -define\_P\_projective\_space\_3\_F\_end\$
  ▶ ▶ -with\_P\_do\$
  ▶ ▶ -projective\_space\_activity\$
  ▶ ▶ ▶ -rank\_lines\_in\_PG\_v1\$
  ▶ ▶ ▶ -end\$
  ▶ ▶ -with\_P\_do\$
  ▶ ▶ -projective\_space\_activity\$
  ▶ ▶ ▶ -rank\_lines\_in\_PG\_v2\$
  ▶ ▶ -end
4.5 Algebraic Sets

A set of points $V$ in $\text{PG}(n, q)$ is algebraic if there is a set of homogeneous polynomials $p_1, \ldots, p_r$ whose roots over $\mathbb{F}_q$ are the given set. In this case, we write $V = \mathbf{v}(p_1, \ldots, p_r)$. The set $V$ is often called the variety of $p_1, \ldots, p_r$.

Conversely, given a set of points $V$ in $\text{PG}(n, q)$, the ideal $I(V)$ is the set of homogeneous polynomials in $\mathbb{F}_q[X_0, \ldots, X_n]$ which vanish on all of $V$. This set is an ideal in the polynomial ring. In $\text{PG}(n, q)$, every set is algebraic of degree at most $(n + 1)(q - 1)$ [29]. The associated polynomial is unique and known as the algebraic normal form of the set.

In order to work with algebraic sets, polynomial rings are required. Orbiter offers homogeneous polynomials in a finite number of variables. There are two orderings of the monomials which can be chosen. The partition ordering is grouping terms according to the partition that results from the degrees of the variables first, and then applies the lexicographic ordering as a tie breaker. The lexicographic ordering orders the monomials lexicographically. Table 4.2 shows the monomials in the partition ordering for degrees 1, 2, 3 and 4 in a plane.

![Table 4.2: The partition ordering of monomials of degree 1, 2, 3 and 4 in a plane](image)

\[
X_0^3 + 3X_2^3 + 10X_1^2X_2 + X_0X_2^2 = 0.
\]
Using the indexing of monomials from Table 4.2, we record the coefficient vector of the equation as sequence

\[(1,0,3,0,0,0,10,1,0,0,0).\]

The Orbiter command

```
EC_11_EQUATION="1,0,3,0,0,0,10,1,0,0"
```

creates the algebraic set associated to the cubic curve \( y^2 = x^3 + x + 3 \) in PG(2,11). It turns out that there are exactly 18 points over \( \mathbb{F}_{11} \) (cf. Figure 4.6). Suppose we want to create the Hirschfeld surface with equation

\[ X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 = 0. \]

Table 4.3 shows the Orbiter monomial orderings for degrees 2 and 3 in PG(3,q). Based on the partition ordering, the equation is coded as coefficient vector

\[(0,0,0,0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,0).\]

The following command can be used to create the variety over \( \mathbb{F}_4 \):

```
HIRSCHFELD_SURFACE_EQUATION="0,0,0,0,0,1,0,1,0,0,1,0,1,0,0,0,0,0,0,0,0"
```

Hirschfeld_surface_q4.txt:

```
> $(ORBITER_PATH)orbiter.out -v 2
> -define:F:-finite_field:-q4:-end
> -define:R:-polynomial_ring:-field:F
> -number_of_variables:4
> -homogeneous_of_degree:3
> -end
```
Table 4.3: The Orbiter ordering of monomials of degree 1, 2 and 3 in PG(3, q)

<table>
<thead>
<tr>
<th>h</th>
<th>mon</th>
<th>vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$X_0^3$</td>
<td>(3, 0, 0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>$X_1^3$</td>
<td>(0, 3, 0, 0)</td>
</tr>
<tr>
<td>2</td>
<td>$X_2^3$</td>
<td>(0, 0, 3, 0)</td>
</tr>
<tr>
<td>3</td>
<td>$X_3^3$</td>
<td>(0, 0, 0, 3)</td>
</tr>
<tr>
<td>4</td>
<td>$X_0^2X_1$</td>
<td>(2, 1, 0, 0)</td>
</tr>
<tr>
<td>5</td>
<td>$X_0^2X_2$</td>
<td>(2, 0, 1, 0)</td>
</tr>
<tr>
<td>6</td>
<td>$X_0^2X_3$</td>
<td>(2, 0, 0, 1)</td>
</tr>
<tr>
<td>7</td>
<td>$X_0X_1^2$</td>
<td>(1, 2, 0, 0)</td>
</tr>
<tr>
<td>8</td>
<td>$X_0X_2^2$</td>
<td>(1, 0, 2, 0)</td>
</tr>
<tr>
<td>9</td>
<td>$X_0X_3^2$</td>
<td>(0, 2, 0, 1)</td>
</tr>
<tr>
<td>10</td>
<td>$X_1X_2^2$</td>
<td>(0, 1, 2, 0)</td>
</tr>
<tr>
<td>11</td>
<td>$X_1X_3^2$</td>
<td>(0, 0, 2, 1)</td>
</tr>
<tr>
<td>12</td>
<td>$X_2X_3^2$</td>
<td>(1, 0, 0, 2)</td>
</tr>
<tr>
<td>13</td>
<td>$X_2X_3^2$</td>
<td>(0, 0, 1, 2)</td>
</tr>
<tr>
<td>14</td>
<td>$X_2X_3^2$</td>
<td>(0, 0, 1, 2)</td>
</tr>
<tr>
<td>15</td>
<td>$X_0X_1X_2$</td>
<td>(1, 1, 1, 0)</td>
</tr>
<tr>
<td>16</td>
<td>$X_0X_1X_3$</td>
<td>(1, 1, 0, 1)</td>
</tr>
<tr>
<td>17</td>
<td>$X_0X_2X_3$</td>
<td>(1, 0, 1, 1)</td>
</tr>
<tr>
<td>18</td>
<td>$X_1X_2X_3$</td>
<td>(0, 1, 1, 1)</td>
</tr>
</tbody>
</table>

Table 4.3: The Orbiter ordering of monomials of degree 1, 2 and 3 in PG(3, q)
Figure 4.6: Elliptic curve $y^2 \equiv x^3 + x + 3 \mod 11$

```bash
> > -define:P:-projective_space-3:F:-end:\n> > -define:H4:-geometric_object-P:\n> > > -projective_variety-R:\n> > > > "Hirschfeld_surface_q4":\n> > > > "Hirschfeld\_surface\_q4":\n> > > > $(HIRSCHFELD_SURFACE_EQUATION)\n> > > > -end:\n> > > -with-H4:-do:-combinatorial_object_activity:-save:\n> > -end
```

A file called `Hirschfeld_surface_q4.txt` is created. The file contains the Orbiter ranks of the 45 points on the surface.

The next command creates the Endrass surface over $\mathbb{F}_7$. The surface is defined as a makefile variable in sparse form.

ENDRASS_SPARSE="\
6,0,4,4,2,7,5,9,6,20,6,23,1,25,3,30,1,32,3,34,4,56,6,59,1,61,6,66,\n2,68,6,70,3,77,2,79,6,83,6,120,2,123,5,125,3,130,1,132,3,134,3,141,\n2,143,6,147,3,156"

Endrass_F7.txt:
```bash
> $(ORBITER_PATH)orbiter.out-v.2\n> -define:F:-finite_field-q.7:-end\n> -define:R:-polynomial_ring:-field:F\n```
Suppose we want to create the monomials of degree 8 in 4 variables. We use an integer program to do so. The following command also applies the unix sort command to sort the monomials:

```
octic_prepare:
> $(ORBITER_PATH)orbiter.out -v 4
> -define A vector -format 1 dense "1,1,1,1" -end
> -define D diophant
> -define label octic_monomials
> -define coefficient_matrix A
> -define RHS "8,8,1"
> -define x_min_global 0 -x_max_global 8
> -end
> -with D do
> -diophant_activity solve mckay
> -end
> sort -r octic_monomials.sol > octic_monomials_sorted.txt
```

There are 165 monomials. They are listed in the file `octic_monomials_sorted.txt`. 
4.6 The Klein Quadric and the Pl"ucker Map

Orbiter can work with Grassmannians over finite field. In particular, Orbiter offers indexing for these sets. For the Grassmannian $\mathfrak{G}_{r,2}(V)$, additional functionality is possible. The Pl"ucker coordinates allow to identify $\mathfrak{G}_{r,2}(V)$ with the $Q^+(5, q)$ quadric.

The command

```
GR_4_2_2:
▷ $(ORBITER_PATH)orbiter.out\ -v\ .2\ 
▷ ▷ -define F::finite_field::q:2::end\ 
▷ ▷ -with F::do--finite_field_activity\ 
▷ ▷ ▷ -cheat_sheet_Gr_4_2_2\ -end 
▷ pdfflatex\ Gr_4_2_2.tex 
▷ open\ Gr_4_2_2.pdf
```

creates the elements of $\mathfrak{G}_{4,2,2}$ and lists them together with their Pl"ucker coordinates. The following output is shortened:

There are 35 lines:

\[
L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} = \text{Pl}(1, 0, 0, 0, 0, 0) \quad L_2 = \begin{bmatrix} 1000 \\ 0101 \end{bmatrix} = \text{Pl}(1, 0, 0, 0, 1, 0) \\
L_1 = \begin{bmatrix} 1000 \\ 0110 \end{bmatrix} = \text{Pl}(1, 0, 1, 0, 0, 0) \quad : \quad L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} = \text{Pl}(0, 1, 0, 0, 0, 0)
\]

The Pl"ucker coordinates satisfy

\[ p_{12}p_{34} - p_{13}p_{24} + p_{14}p_{23} = 0 \]

and hence belong to the quadric $Q^+(5, q)$. This quadric is also known as the Klein quadric. Orthogonal spaces and quadrics will be discussed in Section 4.7. Orbiter has a labeling of points of quadrics that can be used to enumerate the points of $Q^+(5, q)$. Using the inverse Pl"ucker map, this gives a second way to label the lines of $\text{PG}(3, q)$. In the example of $\text{PG}(3, 2)$ this yields the following list (output shortened):
\[ 0 = \mathbf{P}_1(1, 0, 0, 0, 0, 0) = L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \]
\[ 1 = \mathbf{P}_1(0, 1, 0, 0, 0, 0) = L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} \]
\[ 2 = \mathbf{P}_1(0, 0, 1, 0, 0, 0) = L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} \]
\[ \vdots \]
\[ 34 = \mathbf{P}_1(0, 1, 1, 1, 1, 1) = L_{26} = \begin{bmatrix} 1101 \\ 0011 \end{bmatrix} \]
<table>
<thead>
<tr>
<th>Type</th>
<th>Quadratic Form</th>
<th># Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q^+(n, q)$</td>
<td>$\frac{n-1}{2} \sum_{i=0}^{\frac{n-1}{2}-1} X_{2i}X_{2i+1}$</td>
<td>$\frac{(q^{(n+1)/2} - 1)(q^{(n-1)/2} + 1)}{q-1}$</td>
</tr>
<tr>
<td>Hyperbolic ($n$ is odd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q^-(n, q)$</td>
<td>$p(X_{n-1}, X_n) + \frac{n-1}{2} \sum_{i=0}^{\frac{n-1}{2}-1} X_{2i}X_{2i+1}$</td>
<td>$\frac{(q^{(n+1)/2} + 1)(q^{(n-1)/2} - 1)}{q-1}$</td>
</tr>
<tr>
<td>Elliptic ($n$ is odd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q(n, q)$</td>
<td>$X_n^2 + \sum_{i=0}^{\frac{n}{2}-1} X_{2i}X_{2i+1}$</td>
<td>$\frac{q^n - 1}{q-1}$</td>
</tr>
<tr>
<td>Parabolic ($n$ is even)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Nondegenerate Quadrics in $\text{PG}(n, q)$ and the canonical form adopted in Orbiter

### 4.7 Orthogonal Spaces

Orbiter can create and work with orthogonal spaces and their groups. An orthogonal space is created by a quadratic form. We assume that the form is nondegenerate. There are three types of nondegenerate quadratic forms in $\text{PG}(n, q)$. Two when $n$ is odd (hyperbolic and elliptic) and one if $n$ is even (parabolic). Basic information about these quadrics and their representative quadratic forms in Orbiter is given in Table 4.4. Here, $p(X, Y) = c_1X^2 + c_2XY + c_3Y^2 \in F_q[X, Y]$ is irreducible over $F_q$. To create an orthogonal space, the `-orthogonal_space $\epsilon$ $d$ $q$ -end` command can be used. Here, $d = n + 1$, $q$ is the order of the finite field, and

$$
\epsilon = \begin{cases} 
1 & \text{hyperbolic type } Q^+(d-1, q), \ d \text{ even} \\
0 & \text{elliptic type } Q(d-1, q), \ d \text{ odd} \\
-1 & \text{hyperbolic type } Q^-(d-1, q), \ d \text{ even}
\end{cases}
$$

In order to create an object of type orthogonal space, the `-orthogonal_space` command is used inside a `-definition .. -end` command sequence. In Table 4.5, Orbiter command options for creating orthogonal spaces are shown.

For instance, the following command creates $Q(3, 2)$ together with its group $PGO^+(4, 2)$:

```
Op_4.2:
  $\$(ORBITER_PATH)\operatorout\.-v.2\$
  $\$ -define:F:-finite_field:-q.2:-end\$
  $\$ -define:O:-orthogonal_space:1.4:F:-without_group:-end\$
  $\$ -with:O:-do:-orthogonal_space_activity\$
  $\$ -cheat_sheet_orthogonal:-end
  $\$ pdflatex:0.1.4.2_report.tex
  $\$ open:0.1.4.2_report.pdf
```
### Modifier | Arguments | Meaning
--- | --- | ---
-label_txt | L | Set the ascii-label of the space. The label is used for things like file names etc. A default label will be used if this option is not given.
-label_tex | L | Set the tex-label of the space. The label is used within latex reports. A default label will be used if this option is not given.
-without_group | | Do not create the orthogonal group.

Table 4.5: Command options to create an orthogonal space

The next command creates $Q(4,2)$ together with its group $PGO(5,2)$. There are 15 points and 15 lines. The geometry is a configuration $15_3$ which is also known as the Cremona-Richmond configuration.

```
O_5_2_incidence_matrix.csv:
> $(ORBITER_PATH)orbiter.out -v 2 \n>   -define F:-finite_field:-q 2 -end \n>   -define 0:-orthogonal_space:0 5 F:-without_group:-end \n>   -with 0:-do:-orthogonal_space_activity\n>   -export_point_line_incidence_matrix\n>   -end
> $(ORBITER_PATH)orbiter.out -v 2 \n>   -define all one r:-vector:-repeat 1 15 -end \n>   -define all one c:-vector:-repeat 1 15 -end \n>   -draw_matrix\n>   -input_csv_file:O_5_2_incidence_matrix.csv\n>   -box_width:20 -bit_depth:8\n>   -partition:2\n>   -end
> open O_5_2_incidence_matrix_draw.bmp
```

The command also creates a bitmap drawing of the incidence matrix between points and lines of $Q(4,2)$. The incidence matrix is shown in Figure 4.7. The Orbiter indexing for points and lines of quadrics is used to order the rows and columns.

By default, the orthogonal space is created together with the orthogonal group $PGO(n+1,q)$. When $q$ is prime, the group $PGO(n+1,q)$ is created instead (the groups are isomorphic in this case, and $PGO(n+1,q)$ is a bit more efficient). For large orthogonal spaces, creating the group is expensive in terms of time and memory. The a command `-without_group` can be used to prevent the group from being created. For instance

```
-define 0 -orthogonal_space 1 6 2 -end
```
Figure 4.7: Incidence matrix of $Q(4, 2)$

creates an object $O$ of type $Q^+(5, 2)$. In Table 4.6, Orbiter activities for orthogonal spaces are shown.

The command

```
0p_6_2:
  $(\text{ORBITER\_PATH})\text{orbiter.out\,-v\,-2}\$
  $\gg \text{\textendash\-define\textendash F\,-finite\_field\,-q\,2\,-end\textendash }$
  $\gg \text{\textendash\-define\textendash O\,-orthogonal\_space\,1\,\textendash 6\,-F\,-without\_group\,-end\textendash }$
  $\gg \text{\textendash\-with\,-do\,-orthogonal\_space\_activity\textendash }$
  $\gg \text{\textendash\-cheat\_sheet\_orthogonal\,-end}$
  $\text{pdflatex\,-O\,1\,\_6\,2\,-report\,-tex}$
  $\gg \text{open\,-O\,1\,\_6\,2\,-report\,-pdf}$
```

produces a cheat sheet for the quadric $Q^+(5, 2)$. This is the Klein quadric from Section 4.6. Orbiter produces the following output. At the top is the tactical decomposition of the incidence matrix between points and lines with respect to a hyperbolic pair. After that, the points and lines are listed (output shortened):
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cheat_sheet_orthogonal</td>
<td></td>
<td>Create a latex report of the orthogonal space. If the group has been created, the report will contain information about the group also.</td>
</tr>
<tr>
<td>-unrank_line_through_two_points</td>
<td>p1 p2</td>
<td>Determine the rank of the line through p1 and p2.</td>
</tr>
<tr>
<td>-perp</td>
<td>L</td>
<td>Determine the common perp of a set of points. The point ranks are given in the list L.</td>
</tr>
<tr>
<td>-create_BLT_set</td>
<td>descr</td>
<td>Creates a BLT-set of Q(4,q). See Section 12.4.</td>
</tr>
</tbody>
</table>

Table 4.6: Activities related to orthogonal spaces

<table>
<thead>
<tr>
<th>→</th>
<th>9</th>
<th>36</th>
<th>18</th>
<th>18</th>
<th>6</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>↓</td>
<td>9</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The number of points is 35 points:

- \( P_0 = (1,0,0,0,0,0) \)
- \( P_1 = (0,1,0,0,0,0) \)
- \( P_2 = (0,0,1,0,0,0) \)
- \( P_3 = (1,0,1,0,0,0) \)
- \( P_4 = (0,1,1,0,0,0) \)
- \( P_5 = (0,0,0,1,0,0) \)
- \( P_6 = (1,0,0,1,0,0) \)
Orbiter has enumerators for points and lines in orthogonal spaces. For small spaces, the cheat sheet lists points and lines in the Orbiter ordering. Creating the groups can be expensive. For large spaces, it may be necessary to disable the group using the **-without_group** option.

The command

```
Op_6_64_line_rank:
```

```
> $(ORBITER_PATH)orbiter.out -v 2
```
computes the Orbiter rank of the line through the points with rank 15447347 and 15225451, respectively. The rank of the line is 16767254. These ranks refer to the orthogonal geometry. They are different from the ranks of points and lines in projective spaces.

It is possible to create reports for orthogonal spaces without group. In this case, the group information will be skipped. For instance, the following command creates a report for $Q(5, 64)$:

```
Qp_6.64_report:
  $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -4$
  -define:F:\_finite_field:q\_64:\ -end$
  -define:O:\_orthogonal_space:1\_6:F:\ -without\_group:\ -end$
  -with:O:\_do:\_orthogonal_space\_activity$
  -unrank\_line\_through\_two\_points:15447347\_15225451$
  -end
```

The report does not show information about the group. However, it still contains the tactical decomposition with respect to a hyperbolic pair. The printing of points is restricted to small spaces only.

The group is not available.

The quadratic form is:

$$X_0X_1 + X_2X_3 + X_4X_5 = 0$$

<table>
<thead>
<tr>
<th></th>
<th>16769025</th>
<th>1090252800</th>
<th>532350</th>
<th>532350</th>
<th>130</th>
<th>4225</th>
<th>4225</th>
</tr>
</thead>
<tbody>
<tr>
<td>16511040</td>
<td>65</td>
<td>4160</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>266175</td>
<td>0</td>
<td>4096</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>266175</td>
<td>0</td>
<td>4096</td>
<td>0</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4225</td>
<td>3969</td>
<td>0</td>
<td>126</td>
<td>126</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4225</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4225</td>
</tr>
</tbody>
</table>
To study BLT-sets in $Q(4, q)$, see Section 12.4.

<table>
<thead>
<tr>
<th>↓</th>
<th>16769025</th>
<th>1090252800</th>
<th>532350</th>
<th>532350</th>
<th>130</th>
<th>4225</th>
<th>4225</th>
</tr>
</thead>
<tbody>
<tr>
<td>16511040</td>
<td>64</td>
<td>63</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>266175</td>
<td>0</td>
<td>1</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>63</td>
<td>0</td>
</tr>
<tr>
<td>266175</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>63</td>
</tr>
<tr>
<td>4225</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>65</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The number of points is 17047617
Too many points to print.
The number of lines is 1108095105
4.8 Hermitian Varieties

Orbiter has enumerators for points of the hermitian variety $H(k, Q)$. Here, $Q$ is a square, and so $q = \sqrt{Q}$ is an integer. The equation of the variety is

$$\sum_{i=0}^{k} X_i^{q+1} = 0.$$  

The command

```
H_2_4:
▷ $(ORBITER\_PATH)orbiter.out\cdot-v\cdot2\\
▷ ▷ -define\_F\_finite\_field-q\_4\_end\\
▷ ▷ -with\_F\_do\_finite\_field\_activity\\
▷ ▷ ▷ -cheat\_sheet\_hermitian\_2\_end
▷ pdflatex\_H_2_4.tex
▷ open\_H_2_4.pdf
```

produces a cheat sheet for the variety $H(2, 4)$:

```
The Hermitian variety $H(2, 4)$ contains 9 points:

\[
\begin{align*}
    P_0 &= (1, 1, 0) = 4 & P_5 &= (3, 0, 1) = 9 \\
    P_1 &= (2, 1, 0) = 5 & P_6 &= (0, 1, 1) = 10 \\
    P_2 &= (3, 1, 0) = 6 & P_7 &= (0, 2, 1) = 13 \\
    P_3 &= (1, 0, 1) = 7 & P_8 &= (0, 3, 1) = 17 \\
    P_4 &= (2, 0, 1) = 8
\end{align*}
\]

All points: ( 4, 5, 6, 7, 8, 9, 10, 13, 17 )
```

The command

```
H_3_4:
▷ $(ORBITER\_PATH)orbiter.out\cdot-v\cdot2\\
▷ ▷ -define\_F\_finite\_field-q\_4\_end\\
▷ ▷ -with\_F\_do\_finite\_field\_activity\\
▷ ▷ ▷ -cheat\_sheet\_hermitian\_3\_end
▷ pdflatex\_H_3_4.tex
▷ open\_H_3_4.pdf
```

produces a cheat sheet for the variety $H(3, 4)$. 

105
The Hermitian variety $H(3, 4)$ contains 45 points:

$P_0 = (1, 1, 0, 0) = 5 \quad P_{23} = (3, 3, 1, 1) = 52$

$P_1 = (2, 1, 0, 0) = 6 \quad P_{24} = (0, 0, 1, 1) = 38$

$P_2 = (3, 1, 0, 0) = 7 \quad P_{25} = (1, 1, 2, 1) = 58$

$P_3 = (1, 0, 1, 0) = 8 \quad P_{26} = (2, 1, 2, 1) = 59$

$P_4 = (2, 0, 1, 0) = 9 \quad P_{27} = (3, 1, 2, 1) = 60$

$P_5 = (3, 0, 1, 0) = 10 \quad P_{28} = (1, 2, 2, 1) = 62$

$P_6 = (0, 1, 1, 0) = 11 \quad P_{29} = (2, 2, 2, 1) = 63$

$P_7 = (0, 2, 1, 0) = 15 \quad P_{30} = (3, 2, 2, 1) = 64$

$P_8 = (0, 3, 1, 0) = 19 \quad P_{31} = (1, 3, 2, 1) = 66$

$P_9 = (1, 0, 0, 1) = 23 \quad P_{32} = (2, 3, 2, 1) = 67$

$P_{10} = (2, 0, 0, 1) = 24 \quad P_{33} = (3, 3, 2, 1) = 68$

$P_{11} = (3, 0, 0, 1) = 25 \quad P_{34} = (0, 0, 2, 1) = 53$

$P_{12} = (0, 1, 0, 1) = 26 \quad P_{35} = (1, 1, 3, 1) = 74$

$P_{13} = (0, 2, 0, 1) = 30 \quad P_{36} = (2, 1, 3, 1) = 75$

$P_{14} = (0, 3, 0, 1) = 34 \quad P_{37} = (3, 1, 3, 1) = 76$

$P_{15} = (1, 1, 1, 1) = 4 \quad P_{38} = (1, 2, 3, 1) = 78$

$P_{16} = (2, 1, 1, 1) = 43 \quad P_{39} = (2, 2, 3, 1) = 79$

$P_{17} = (3, 1, 1, 1) = 44 \quad P_{40} = (3, 2, 3, 1) = 80$

$P_{18} = (1, 2, 1, 1) = 46 \quad P_{41} = (1, 3, 3, 1) = 82$

$P_{19} = (2, 2, 1, 1) = 47 \quad P_{42} = (2, 3, 3, 1) = 83$

$P_{20} = (3, 2, 1, 1) = 48 \quad P_{43} = (3, 3, 3, 1) = 84$

$P_{21} = (1, 3, 1, 1) = 50 \quad P_{44} = (0, 0, 3, 1) = 69$

$P_{22} = (2, 3, 1, 1) = 51$

All points: (5, 6, 7, 8, 9, 10, 11, 15, 19, 23, 24, 25, 26, 30, 34, 4, 43, 44, 46, 47, 48, 50, 51, 52, 38, 58, 59, 60, 62, 63, 64, 66, 67, 68, 53, 74, 75, 76, 78, 79, 80, 82, 83, 84, 69)

Coincidentally, this Hermitian variety is the Hirschfeld cubic surface over $\mathbb{F}_4$. 

106
4.9 Advanced Topics

The Orbiter commands associated with projective space objects are summarized in Tables 4.7-4.9.

Table 4.10 lists Orbiter global commands related to projective geometries. These commands do not need an object of type projective space in order to be invoked.

Suppose we want to study the fix structure of a collineation in projective space. Suppose we want to do so for the element

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

which is a Baer collineation. It fixes a subspace PG(3, 2). The command

```
fix_structure_2A:
  ▷ $(ORBITER_PATH)orbiter.out-\text{-v.2}\$
  ▷ ▷ -define-F--finite_field--q.4--end\$
  ▷ ▷ -define-P--projective_space-3.F--end\$
  ▷ ▷ -with-P--do\$
  ▷ ▷ -projective_space_activity\$
  ▷ ▷ ▷ -cheat_sheet_for_decomposition_by_element_PG.1\$
  ▷ ▷ ▷ "1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,1,0,1"\$
  ▷ ▷ ▷ fix_structure_2A\$
  ▷ ▷ -end\$
  ▷ pdflatex-fix_structure_2A.tex
  ▷ open-fix_structure_2A.pdf
```

can be used.

Suppose we are looking for a projectivity of PG(3, 16) fixing the plane \(v(X_3)\) pointwise and mapping a pair of skew lines not in that plane to another pair of skew lines not in that plane. For instance, suppose we want to map

\[
M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mapsto N_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
M_2 = \begin{bmatrix} 1 & 1 & 0 & \delta \\ 0 & 0 & 1 & 0 \end{bmatrix} \mapsto N_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

The command
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-export_point_line_incidence_matrix</td>
<td></td>
<td>Create a csv file of the point line incidence matrix.</td>
</tr>
<tr>
<td>-table_of_cubic_surfaces_compute_properties</td>
<td>fname $q_0$ col-offset</td>
<td>See Section 7.5.</td>
</tr>
<tr>
<td>-cubic_surface_properties_analyze</td>
<td>fname $q_0$</td>
<td>See Section 7.5.</td>
</tr>
<tr>
<td>-canonical_form_of_code</td>
<td>label $m$ $n$ matrix</td>
<td>Compute the automorphism group of a linear code using Nauty. See Section 10.</td>
</tr>
<tr>
<td>-map</td>
<td>label parameters</td>
<td>evaluate a formula using the given parameters</td>
</tr>
<tr>
<td>-analyze_del_Pezzo_surface</td>
<td>label parameters</td>
<td></td>
</tr>
<tr>
<td>-cheat_sheet_for_decomposition_by_element_PG</td>
<td>power elt fname</td>
<td>Analyzes the orbit structure of the cyclic group generated by the given element in the action on $\text{PG}(n,q)$.</td>
</tr>
<tr>
<td>-cheat_sheet_for_decomposition_by_subgroup</td>
<td>label descr</td>
<td>Analyzes the orbit structure of the subgroup $H$ in the action on $\text{PG}(n,q)$. The subgroup must be a linear group, and the description of $H$ must come from the commands from Section 5.2.</td>
</tr>
<tr>
<td>-define_surface</td>
<td>label descr</td>
<td>To create a cubic surface and add it to the symbol table under the given label. See Section 7.1.</td>
</tr>
<tr>
<td>-table_of_quartic_curves</td>
<td></td>
<td>Export the classification of quartic curves to a csv file.</td>
</tr>
<tr>
<td>-table_of_cubic_surfaces</td>
<td></td>
<td>Export the classification of cubic surfaces to a csv file.</td>
</tr>
<tr>
<td>-define_quartic_curve</td>
<td>label descr</td>
<td>To create a quartic curve and add it to the symbol table under the given label. See Section 7.2.</td>
</tr>
<tr>
<td>-classify_surfaces_with_double_sixes</td>
<td>label control</td>
<td>Classify cubic surfaces using the double six approach. See Section 7.3.</td>
</tr>
</tbody>
</table>

Table 4.7: Projective Space Activities (Part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-classify_surfaces_through_arcs_and_two_lines</code></td>
<td></td>
<td>See Section 7.3.</td>
</tr>
<tr>
<td><code>-test_nb_Eckardt_points</code></td>
<td>nbE</td>
<td>See Section 7.3.</td>
</tr>
<tr>
<td><code>-classify_surfaces_through_arcs_and_trihedral_pairs</code></td>
<td></td>
<td>See Section 7.3.</td>
</tr>
<tr>
<td><code>-sweep</code></td>
<td>fname</td>
<td></td>
</tr>
<tr>
<td><code>-sweep_4</code></td>
<td>fname surface-descr</td>
<td></td>
</tr>
<tr>
<td><code>-sweep_4_27</code></td>
<td>fname surface-descr</td>
<td></td>
</tr>
<tr>
<td><code>-six_arcs_not_on_conic</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>-filter_by_nb_Eckardt_points</code></td>
<td>nbE</td>
<td></td>
</tr>
<tr>
<td><code>-surface_quartic</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>-surface_clebsch</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>-surface_codes</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>-trihedral_control</code></td>
<td>poset-control</td>
<td></td>
</tr>
<tr>
<td><code>-trihedra2_control</code></td>
<td>poset-control</td>
<td></td>
</tr>
<tr>
<td><code>-control_six_arcs</code></td>
<td>poset-control</td>
<td></td>
</tr>
<tr>
<td><code>-make_gilbert_varshamov_code</code></td>
<td>n d</td>
<td>See Section 10.8.</td>
</tr>
</tbody>
</table>

Table 4.8: Projective Space Activities (Part 2)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-spread_classify</code></td>
<td>$k$ control</td>
<td>See Section 12.1.</td>
</tr>
<tr>
<td><code>-classify_semifields</code></td>
<td>descr</td>
<td></td>
</tr>
<tr>
<td><code>-cheat_sheet</code></td>
<td>fname-mask $N$</td>
<td>Produce a cheat sheet for PG($n,q$)</td>
</tr>
<tr>
<td><code>-classify_quartic_curves_nauty</code></td>
<td>fname-mask $N$ $k$ $d$ fname</td>
<td>Classify quartic curves using Nauty.</td>
</tr>
<tr>
<td><code>-classify_quartic_curves_with_substructure</code></td>
<td>fname-mask $N$ $k$ $d$ fname</td>
<td>Classify quartic curves using substructure algorithm.</td>
</tr>
<tr>
<td><code>-set_stabilizer</code></td>
<td>$k$ fname-mask $N$ col-label</td>
<td>Compute canonical form of sets using the substructure algorithm.</td>
</tr>
<tr>
<td><code>-lift_skew_hexagon</code></td>
<td>text</td>
<td>Lift a skew-hexagon.</td>
</tr>
<tr>
<td><code>-lift_skew_hexagon_with_polarity</code></td>
<td>polarity</td>
<td>Lift a skew-hexagon with a given polarity.</td>
</tr>
<tr>
<td><code>-arc_with_given_set_as_s_lines_after_dualizing</code></td>
<td>$sz$ $d$ $d_{\text{min}}$ $s$</td>
<td>Finds arcs with the given set as $s$-lines.</td>
</tr>
<tr>
<td><code>-arc_with_two_given_sets_of_lines_after_dualizing</code></td>
<td>$sz$ $d$ $d_{\text{min}}$ $s$ $t$ $T$</td>
<td>Finds arcs with the two given sets as $s$-lines and $t$-lines, respectively.</td>
</tr>
<tr>
<td><code>-arc_with_three_given_sets_of_lines_after_dualizing</code></td>
<td>$sz$ $d$ $d_{\text{min}}$ $s$ $t$ $T$ $u$ $U$</td>
<td>Finds arcs with the three given sets as $s$-lines and $t$-lines and $u$-lines, respectively.</td>
</tr>
<tr>
<td><code>-dualize_hyperplanes_to_points</code></td>
<td></td>
<td>Turns ranks of hyperplanes into ranks of points.</td>
</tr>
<tr>
<td><code>-dualize_points_to_hyperplanes</code></td>
<td></td>
<td>Turns ranks of points into ranks of hyperplanes.</td>
</tr>
</tbody>
</table>

Table 4.9: Projective Space Activities (Part 3)
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-classify_cubic_curves</td>
<td>q</td>
<td>Classifies cubic curves in PG(2, q). Requires -control_arcs. See Section 6.6.</td>
</tr>
<tr>
<td>-control_arcs</td>
<td>description</td>
<td>Poset classification control for arcs used during the classification of cubic curves. See Table 6.2.</td>
</tr>
<tr>
<td>-create_points_on_quartic</td>
<td>( \epsilon )</td>
<td>Creates a table of points on a specific quartic curve. Consecutive points are no more than ( \epsilon ) apart.</td>
</tr>
<tr>
<td>-create_points_on_parabola</td>
<td>( \epsilon, a, b, c )</td>
<td>Creates a table of points on the parabola ( y = ax^2 + bx + c ). Consecutive points are no more than ( \epsilon ) apart.</td>
</tr>
<tr>
<td>-smooth_curve</td>
<td>( \epsilon, N, b, t_{\text{min}}, t_{\text{max}}, \text{function} )</td>
<td>Creates at least ( N ) points on a continuous curve given by “function”. Consecutive points are no more than ( \epsilon ) apart. The function must be in terms of a parameter ( t ). The values of ( t ) are taken from the interval ([t_{\text{min}}, t_{\text{max}}]).</td>
</tr>
<tr>
<td>-create_spread</td>
<td>description</td>
<td>Creates a spread according to the description. See Section 12.1.</td>
</tr>
<tr>
<td>-make_table_of_surfaces</td>
<td></td>
<td>Produces a latex table summarizing the surfaces in the Orbiter catalogue.</td>
</tr>
</tbody>
</table>

Table 4.10: Orbiter commands related to projective geometries
trans:
  >> $(ORBITER_PATH)orbiter.out:-v.5-
  >> > -define:F:-finite_field:-q.16:-end-
  >> > -define:P:-projective_space:3:F:-end-
  >> > -with:P:-do-
  >> > -projective_space_activity-
  >> > > -move_two_lines_in_hyperplane_stabilizer_text-
  >> > > > "1,0,0,0,0,0,0,1"."1,1,0,2,0,0,1,0".\n  >> > > > "1,0,0,0,0,0,0,1"."0,1,0,1,0,0,1,0".\n  >> > > -end

computes a projectivity (transvection) to do so:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\delta^{14} & 0 & 0 & \delta^{14}
\end{bmatrix}
\]

Here, \(\delta\) is the primitive element in the built-in field \(\mathbb{F}_{16}\), satisfying \(\delta^4 = \delta^3 + 1\).

It is possible to define algebraic varieties directly from an algebraic equation. We distinguish between managed variables and arbitrary variables. We require that the polynomial is homogeneous in the managed variables. The other variables can be used to represent scalar parameters, for instance. Here is an example. Suppose we want to study the del Pezzo surfaces

\[f_3: \ w^2 = x^4 + y^4 + z^4 + 8x^2y^2 + 8x^2z^2 + 8y^2z^2, \quad f_4: \ w^2 = x^4 + y^4 + z^4 - x^2y^2.\]

Orbiter assumes that the equation has \(w^2\) on the left hand side. Therefore, only the right hand side of the equation needs to be given. We translate the equation into simplified notation as follows:

\[x*x*x*x+y*y*y*y+z*z*z*z+8*x*x*y*y+8*x*x*z*z+8*y*y*z*z\]

for \(f_3\) and

\[x*x*x*x+y*y*y*y+z*z*z*z-x*x*y*y\]

for \(f_4\). The following command can be used to produce a report on the two surfaces over the field \(\mathbb{F}_{13}\).

del_Pezzo_F13ab_report:
  >> $(ORBITER_PATH)orbiter.out:-v.3-
  >> > -define:F:-finite_field:-q.13:-end-

112
The third argument after the `-formula` command specifies the managed variables, which are $x, y, z$. The command `-collection` is used to group objects together. In this case, both surfaces are group together under new name. That way, we can issue the `-analyze_del_Pezzo_surface` once, and it applies to both surfaces.
4.10 Geometric Objects

Orbiter can create objects in projective space. To do so, define an object of type \texttt{-geometric\_object}. The definition of a geometric object requires a projective geometry object. For this reason, the definition requires an extra argument, which is the label of a previously created projective geometry object. After that, one of the commands shown in Tables 4.11 and 4.12 can be issued. Modifier options as shown in Table 4.13 apply. For instance, the command sequence

```plaintext
elliptic\_curve\_b1\_c3\_q11.txt:
  \texttt{$(ORBITER\_PATH)\textbackslash orbiter.out\textbackslash -v.2\textbackslash $
  \texttt{-define\textbar F\textbar -finite\_field\textbar -q.11\textbar -end\textbackslash$
  \texttt{-define\textbar P\textbar -projective\_space\textbar 2\textbar F\textbar -end\textbackslash$
  \texttt{-define\textbar EC\textbar -geometric\_object\textbar P\textbar$
  \texttt{-elliptic\_curve\textbar 1\textbar 3\textbar$
  \texttt{-end\textbar$
  \texttt{-with\textbar EC\textbar -do\textbar -combinatorial\_object\_activity\textbar -save\textbar$
  \texttt{-end}
```

creates the elliptic curve

\[ y^2 \equiv x^3 + x + 3 \mod 11 \]

over the field $\mathbb{F}_{11}$. The curve has 18 points, whose orbiter ranks are saved to the file

`elliptic\_curve\_b1\_c3\_q11.txt`.

The following command creates an elliptic quadric ovoid on $\text{PG}(3,8)$:

```plaintext
elliptic\_quadric\_ovoid\_q8:
  \texttt{$(ORBITER\_PATH)\textbackslash orbiter.out\textbackslash -v.2\textbackslash $
  \texttt{-define\textbar F\textbar -finite\_field\textbar -q.8\textbar -end\textbar$
  \texttt{-define\textbar P\textbar -projective\_space\textbar 3\textbar F\textbar -end\textbar$
  \texttt{-define\textbar O\textbar -geometric\_object\textbar P\textbar$
  \texttt{-elliptic\_quadric\_ovoid\textbar$
  \texttt{-end\textbar$
  \texttt{-with\textbar O\textbar -do\textbar -combinatorial\_object\_activity\textbar -save\textbar$
  \texttt{-end}
```

The next command creates the Suzuki-Tits ovoid in $\text{PG}(3,8)$:

```plaintext
ovoid\_ST\_q8:
  \texttt{$(ORBITER\_PATH)\textbackslash orbiter.out\textbackslash -v.2\textbackslash$
  \texttt{-define\textbar F\textbar -finite\_field\textbar -q.8\textbar -end\textbar$
  \texttt{-define\textbar P\textbar -projective\_space\textbar 3\textbar F\textbar -end\textbar$
  \texttt{-define\textbar O\textbar -geometric\_object\textbar P\textbar$
```

114
<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-hyperoval</td>
<td></td>
<td>To create a hyperoval</td>
</tr>
<tr>
<td>-subiaco_oval</td>
<td>f_short</td>
<td>Create the Subiaco oval</td>
</tr>
<tr>
<td>-subiaco_hyperoval</td>
<td></td>
<td>Create the Subiaco hyperoval</td>
</tr>
<tr>
<td>-adelaide_hyperoval</td>
<td></td>
<td>Create the Adalaide hyperoval</td>
</tr>
<tr>
<td>-translation</td>
<td>i</td>
<td>Create the translation hyperoval with exponent i</td>
</tr>
<tr>
<td>-Segre</td>
<td></td>
<td>Create the Segre hyperoval</td>
</tr>
<tr>
<td>-Payne</td>
<td></td>
<td>Create the Payne hyperoval</td>
</tr>
<tr>
<td>-Cherowitzo</td>
<td></td>
<td>Create the Cherowitzo hyperoval</td>
</tr>
<tr>
<td>-OKeefe_Penttila</td>
<td></td>
<td>Create the O’Keefe, Penttila hyperoval</td>
</tr>
<tr>
<td>-BLT_database</td>
<td>k</td>
<td>Create the kth BLT-set of order q from the database (k = 0, 1, ...)</td>
</tr>
<tr>
<td>-elliptic_quadric_</td>
<td></td>
<td>Create an elliptic quadric ovoid in PG(3,q).</td>
</tr>
<tr>
<td>ovoid</td>
<td></td>
<td>Create the Suzuki Tits ovoid in PG(3,q). Here, q = 2^{2r+1}.</td>
</tr>
<tr>
<td>-Baer</td>
<td></td>
<td>Create the (standard) Baer subgeometry</td>
</tr>
<tr>
<td>-orthogonal</td>
<td>ε</td>
<td>Create the Qε(n,q) quadric</td>
</tr>
<tr>
<td>-hermitian</td>
<td></td>
<td>Create the Hermitian variety given by ( \sum_{i=0}^{n} X_i^{\sqrt{q}+1} = 0 )</td>
</tr>
<tr>
<td>-cuspidal_cubic</td>
<td></td>
<td>Create the cuspidal cubic ((s^3, ts^2, t^3)) in PG(2,q)</td>
</tr>
<tr>
<td>-twisted_cubic</td>
<td></td>
<td>Create a twisted cubic ((s^3, s^2t, st^2, t^3)) in PG(3,q)</td>
</tr>
<tr>
<td>-elliptic_curve</td>
<td>a b</td>
<td>Create the elliptic curve ( y^2 = x^3 + ax + b )</td>
</tr>
<tr>
<td>-ttp_construction_A</td>
<td></td>
<td>Create the twisted tensor product code of type A [8]</td>
</tr>
<tr>
<td>-ttp_construction_A_h</td>
<td></td>
<td>Create the twisted tensor product code of type A [8]</td>
</tr>
<tr>
<td>-ttp_construction_B</td>
<td></td>
<td>Create the twisted tensor product code of type B [8]</td>
</tr>
</tbody>
</table>

Table 4.11: Orbiter Objects (Part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-unital_XXq_YZq_ZYq</td>
<td></td>
<td>Create the unital with equation (XX^q + YZ^q + ZY^q = 0)</td>
</tr>
<tr>
<td>-desarguesian_line_spread_in_PG_3_q</td>
<td></td>
<td>Create the desarguesian line spread in PG(3, q) as a set of 2-subspaces</td>
</tr>
<tr>
<td>-Buekenhout_Metz</td>
<td></td>
<td>Create the Buekenhout Metz unital</td>
</tr>
<tr>
<td>-Uab</td>
<td>(a \ b)</td>
<td>Create the Buekenhout Metz unital in the form of Barwick and Ebert [5]</td>
</tr>
<tr>
<td>-whole_space</td>
<td></td>
<td>Create the whole space</td>
</tr>
<tr>
<td>-hyperplane</td>
<td>(pt)</td>
<td>Create the hyperplane given by dual coordinates associated with the given point</td>
</tr>
<tr>
<td>-segre_variety</td>
<td>(a \ b)</td>
<td>Create the Segre variety</td>
</tr>
<tr>
<td>-Maruta_Hamada_arc</td>
<td></td>
<td>Create the Maruta Hamada arc</td>
</tr>
<tr>
<td>-projective_variety</td>
<td>(lab_\text{ascii} \ lab_\text{tex} \ d \ \text{coeffs})</td>
<td>Create a projective variety of degree (d) from an equation. By default, the coefficients of the equation are listed in the partition ordering. A different ordering can be specified. A label for the variety in ascii and in tex is required. See Section 4.5.</td>
</tr>
<tr>
<td>-intersection_of_zariski_open_sets</td>
<td>(l \ d \ n \ C_1 \ldots C_n)</td>
<td>Create the intersection of the Zariski open sets given by equations (C_1, \ldots C_n) of degree (d) with label (l), see Section 4.5.</td>
</tr>
<tr>
<td>-projective_curve</td>
<td>(l \ r \ d \ C)</td>
<td>Create the projective curve of degree (d) with label (l), with coefficient vector (C) in (r) variables</td>
</tr>
</tbody>
</table>

Table 4.12: Orbiter Objects (Part 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-embedded_in_PG_4_q</td>
<td></td>
<td>Create the BLT-set with ranks in PG(4, q) instead of orthogonal point ranks</td>
</tr>
<tr>
<td>-BLT_in_PG</td>
<td></td>
<td>Select lexicographic ordering of coefficients in an algebraic equation.</td>
</tr>
<tr>
<td>-monomial_type_LEX</td>
<td></td>
<td>Select partition ordering of coefficients in an algebraic equation (default).</td>
</tr>
</tbody>
</table>

Table 4.13: Orbiter Objects: Modifiers
The Edge curve is given by the equation

\[ X^4 - Y^4 - Z^4 + 2f^2Y^2Z^2 + 4fX^2YZ = 0 \]

where \( f \) is a primitive element of \( \mathbb{F}_q \). Let us pick \( q = 17 \). The next example creates the Edge curve in \( \text{PG}(2, 17) \) and saves it to file. The equation is encoded using the ordering of quartic monomials from Table 4.2.

\[ \text{EDGE\_CURVE\_Q17\_EQUATION} = "1,16,16,0,0,0,0,0,0,0,0,1,12,0,0" \]

The following command computes the line type of the Edge curve:

\[ \text{echo \$\{FILE\_Q17\}:edge\_q17.csv} \]
The line type is

$$(4^6, 2^{30}, 1^{132}, 0^{139})$$

This means that there are 6 4-secants, 30 2-secants, 132 tangent lines, and 139 external lines to the curve.
Chapter 5

Group Theory

5.1 Permutation Groups

Permutation groups can be represented on a computer using the technique of stabilizer chains, or Sims chains (cf. [35, 59]). The stabilizer chain is defined with respect to a sequence of points in the permutation domain called a base. A set of generators which allows to generate each group along the chain is called a strong generating set. Many algorithms for permutation groups rely on knowing a base and strong generating set. In Orbiter, permutation groups can be created from a base and strong generating set. Many types of groups come with their own built-in base and strong generating set. On the other hand, it is also possible to create groups from generating sets which are either not strong or for which a base is not known. For efficiency purposes, it is best to work with small bases.

In order to establish the permutation representation of a group, the technique of indexing is used. Indexing sets up a fixed bijection between the permutation domain (the set we act on) and the integer interval $[0, n - 1]$ for some $n$. The integer associated to an element in the permutation domain is called the rank. Conversely, given an integer in $[0, n - 1]$, the element in the permutation domain associated with it is obtained by the unrank function. The process of converting integers to elements of the permutation domain and vice-versa is indexing. We have seen indexing for projective points in Section 4.1.

In Section 5.2, we will discuss matrix groups over finite fields. The enumerators for projective points from Section 4.1 are used to realize the permutation domain. This enumerator relies on an enumerator for finite fields, as discussed in Sections 3.2 and 3.3. For extension fields, the enumerator for finite fields in turn depends on the choice of the irreducible polynomial which is used to create the field. For affine groups, a different enumerator is used to describe the permutation domain. This enumerator uses the base-$q$ representation of integers, which associates a vector over $\mathbb{F}_q$ of length $n$ with an integer in $[0, q^n - 1]$.

Group elements can be defined using a compact representation as integer vectors. For instance, for linear groups, the coding of elements consists of the entries of the associated matrix (for projective matrix groups, the coding is not unique as scalar multiples of the matrix describe the same group element). For semilinear matrix groups, an extra integer is used to
describe the associated field automorphism as a power of the generator of the group of field automorphisms (the Frobenius endomorphism as transformation). For affine groups, the coding consists of a matrix, a vector and possible a integer describing a field automorphism. Generating sets of groups can be specified by listing generators in coded form.

We start with an example of an explicit permutation group using a known base and strong generating set, using the \texttt{bsgs} command. Here is the cyclic group of order 13 acting on the permutation domain $[0, 12]$. The base is (0). When creating a group, we supply a label in ascii text and in tex. Then we specify the degree of the action, and the group order. After that, we specify the number of generators and the generators themselves. The labels will be used in reports about the group, for instance.

\begin{verbatim}
GEN_C13="1,2,3,4,5,6,7,8,9,10,11,12,0"
#(0,1,2,3,4,5,6,7,8,9,10,11,12)

C13:
▷ $(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_10\_\backslash
▷ ▷ -define.gens.-vector.-dense\$(\text{GEN}\_\text{C13})\_\text{-end}\_\backslash
▷ ▷ -define.G.-permutation_group\_\backslash
▷ ▷ ▷ -bsgs\text{C13-C\{}13\}\_13\_13\_0\_1\_\backslash
▷ ▷ ▷ ▷ gens\_\backslash
▷ ▷ ▷ ▷ -with.G.-do\_\backslash
▷ ▷ ▷ ▷ -group_theoretic_activity\_\backslash
▷ ▷ ▷ ▷ -export_orbiter\_\backslash
▷ ▷ ▷ ▷ -end\_\backslash
▷ ▷ ▷ -with.G.-do\_\backslash
▷ ▷ ▷ -group_theoretic_activity\_\backslash
▷ ▷ ▷ ▷ -export_group_table\_\backslash
▷ ▷ ▷ ▷ -end\_\backslash
▷ ▷ ▷ ▷ -with.G.-do\_\backslash
▷ ▷ ▷ ▷ -group_theoretic_activity\_\backslash
▷ ▷ ▷ ▷ ▷ -report\_\backslash
▷ ▷ ▷ ▷ ▷ -end\_\backslash
▷ ▷ ▷ ▷ -with.G.-do\_\backslash
▷ ▷ ▷ ▷ -group_theoretic_activity\_\backslash
▷ ▷ ▷ ▷ ▷ -save_elements_csv\"C13\_elts.csv\"\_\backslash
▷ ▷ ▷ ▷ ▷ -end
▷ pdflatex:C13\_report.tex
▷ open:C13\_report.pdf
\end{verbatim}

The makefile variable \texttt{GEN\_C13} is used to define the generator of the group, which is the cycle $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$. 

120
The generator is given in list notation, which is the second row in the array
\[
\begin{bmatrix}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 0
\end{bmatrix}.
\]

The command creates the group from the known base 0. After that, several activities are invoked. Specifically, these are group theoretic activities. They will be discussed in more detail in Section 5.6.

Let us take a closer look at the three activities performed in this example. The \texttt{-export_orbiter} command exports the group in Orbiter makefile format. The file \texttt{C13.makefile} is generated, which can be used to recreate the permutation group in an Orbiter makefile. Here is the content of the file:

```
GENERATOR_C13_0=\n    "1,2,3,4,5,6,7,8,9,10,11,12,0"

C13:
    $(ORBITER\PATH)orbiter.out\-v\-2\n    \-define-gens\-vector\-file\C13 gens.csv\-end\n    \-define\G\-permutation\group\n    \-bsgs\C13:\"C\{13\}".13.13.0.1.gens\-end\n```

The activity \texttt{-report} produces a report for the cyclic group, shown below:

**Stabilizer chain**

<table>
<thead>
<tr>
<th>Level</th>
<th>Base pt</th>
<th>Orbit length</th>
<th>Subgroup order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>
Basic orbit 0 has size 13
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

The command `-save_elements_csv` creates a csv file containing all group elements. Each group element is listed one-by-one, using the list notation of permutations. The csv file C13_elts.csv has the following content:

Row,Element
0,"0,1,2,3,4,5,6,7,8,9,10,11,12"
1,"1,2,3,4,5,6,7,8,9,10,11,12,0"
2,"2,3,4,5,6,7,8,9,10,11,12,0,1"
3,"3,4,5,6,7,8,9,10,11,12,0,1,2"
4,"4,5,6,7,8,9,10,11,12,0,1,2,3"
5,"5,6,7,8,9,10,11,12,0,1,2,3,4"
6,"6,7,8,9,10,11,12,0,1,2,3,4,5"
7,"7,8,9,10,11,12,0,1,2,3,4,5,6"
8,"8,9,10,11,12,0,1,2,3,4,5,6,7"
9,"9,10,11,12,0,1,2,3,4,5,6,7,8"
10,"10,11,12,0,1,2,3,4,5,6,7,8,9"
11,"11,12,0,1,2,3,4,5,6,7,8,9,10"
12,"12,0,1,2,3,4,5,6,7,8,9,10,11"

END

Let us look at a symmetric group. The following command creates Sym(3):

Symmetric_3:
> $(ORBITER_PATH)orbiter.out:-v.5:\
The report is shown below:

\begin{verbatim}
\$ (ORBITER_PATH) orbiter.out -v 2 \\
  -define all_one_r -vector -repeat 1:6 -end \\
  -define all_one_c -vector -repeat 1:6 -end \\
  -draw matrix \\
  -input_csv_file Perm3_group_table.csv \\
  -box_width 50 -bit_depth 24 \\
  -partition 3 \\
  all_one_r all_one_c \\
- end \\
\$ (ORBITER_PATH) orbiter.out -v 2 \\
  -define M3 -vector -load_csv_data_column \\
  Symmetric3_elts.csv 1 -end \\
  save_matrix_csv M3 \\
- end \\
\$ (ORBITER_PATH) orbiter.out -v 2 \\
  -define all_one_r -vector -repeat 1:6 -end \\
  -define all_one_c -vector -repeat 1:3 -end \\
  -draw matrix \\
  -input_csv_file M3_matrix.csv \\
  -box_width 50 -bit_depth 8 \\
  -partition 3 \\
  all_one_r all_one_c \\
- end \\
# pdflatex Perm3_report.tex \\
# open Perm3_report.pdf
\end{verbatim}

123
The command also produces a list of group elements in graphical form. Likewise, the group table is produced, again graphically. Both diagrams are shown in Figure 5.1.
Figure 5.1: The elements of Sym(3) and the group table

For comparison, let us have a look at a linear group. Suppose we want to create PGL(4, 2) in the action on points. We use the following Orbiter command to create the group:

```
PGL_4_2_export:
  · $(ORBITER_PATH)orbiter.out::v.2:
    · define F::finite_field::q.2::end:
    · define G::linear_group::PGL.4:F::end:
    · with G::do:
    · -group_theoretic_activity::
      · -report:
      · -end:
    · -with G::do:
    · -group_theoretic_activity::
      · -export_orbiter:
      · -end
  · pdflatex PGL_4_2_report.tex
  · open PGL_4_2_report.pdf
```

The command invokes two activities. The first creates a latex report for the group in the file `PGL_4_2_report.tex`. The second activity exports the permutation representation in Orbiter makefile format. The file `PGL_4_2.makefile` is created:

```
PGL_4_2:
  · $(ORBITER_PATH)orbiter.out::v.2:
    · define gens::vector::file PGL_4_2_gens.csv::end:
    · define G::permutation_group:
    · bsgs PGL_4_2::{\rm PGL}(4,2)".15.20160."0,1,2,3".6 gens::end:
```

125
This command can be used to recreate the group as permutation group directly. This group will be considered again in Section 5.2 below. The permutation representation itself is stored in the file `PGL_4_2_gens.csv`:

```
Row, C0, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14
0, 0, 1, 2, 9, 14, 5, 6, 7, 8, 3, 11, 10, 13, 12, 4
1, 0, 1, 2, 10, 13, 5, 6, 7, 8, 11, 3, 9, 14, 4, 12
2, 0, 1, 2, 12, 11, 5, 6, 7, 8, 13, 14, 4, 3, 9, 10
3, 0, 1, 3, 2, 4, 5, 9, 10, 11, 6, 7, 8, 12, 13, 14
4, 0, 2, 1, 3, 4, 6, 5, 7, 8, 9, 12, 13, 10, 11, 14
5, 1, 0, 2, 3, 4, 5, 7, 6, 8, 10, 9, 11, 12, 14, 13
```

It is possible to create a permutation group as a subgroup of the symmetric group, using the known base for the symmetric group. Because the base of the symmetric group is large, this way of creating the group is less efficient than creating the group with a known (small) base. Here is an example. We create $C_{13}$ as a subgroup of $\text{Sym}(13)$.

```
C13_as_subgroup:
  ▶ $(\text{ORBITER\_PATH})\text{orbiter.out} \cdot -v\cdot 10\cdot \backslash$
  ▶ ▶ -defineG:-permutation\_group:-symmetric\_group:13\cdot \backslash$
  ▶ ▶ ▶ -subgroup\_by\_generatorsC13:13\cdot 1\cdot $(\text{GEN\_C13})\cdot -end\cdot \backslash$
  ▶ ▶ -withG:-do\cdot \backslash$
  ▶ ▶ -group\_theoretic\_activity\cdot \backslash$
  ▶ ▶ ▶ -export\_orbiter\cdot \backslash$
  ▶ ▶ -end\cdot \backslash$
  ▶ ▶ -withG:-do\cdot \backslash$
  ▶ ▶ -group\_theoretic\_activity\cdot \backslash$
  ▶ ▶ -report\cdot \backslash$
  ▶ ▶ -end\cdot \backslash$
  ▶ ▶ -withG:-do\cdot \backslash$
  ▶ ▶ -group\_theoretic\_activity\cdot \backslash$
  ▶ ▶ ▶ -save\_elements\_csv\cdot "C13\_elts.csv"\cdot \backslash$
  ▶ ▶ -end
  ▶ #pdflatex\cdot Perm13\_Subgroup\_C13:13\_report.tex
  ▶ #open\cdot Perm13\_Subgroup\_C13:13\_report.pdf
```

The `subgroup_by_generators` command will be discussed in more detail in Section 5.3.
5.2 Linear Groups

Orbiter provides support for matrix groups and their various permutation representationes. For background information about the classical groups of matrices over finite fields, see cf. \cite{64}. Any group in Orbiter is associated with a permutation action. There can be multiple actions for the same group though. Using homomorphisms of permutation groups, new actions can be formed from old actions. Basic group actions are projective, affine, and general linear, as well as orthogonal, unitary and tensor product. Product actions can be defined also. In order to establish a permutation representation, the elements (aka points) of the permutation domain need to be made available. One way would be to make a table of all elements in the permutation domain. However, this would be time and memory intensive. For this reason, a different technique is used that creates points only when needed. The way this works is that the permutation domain is encoded implicitly, using a fixed bijection to a suitable integer interval (zero based), called the domain. Whenever we want the \( i \)th point in the domain, we can call a function that produces it. Conversely, whenever we have a point, we can call a function that tells us what the associated index in the domain. This is facilitated by two mutually inverse functions. The rank function turns a point into an index. The unrank function turns an index in the domain into a point. Rank and unrank functions are helpful because they eliminate the need for tables of all objects. The ranks lead to rather compact storage of objects in files. The objects can be reconstructed from the ranks.

Let \( V \simeq \mathbb{F}^n_q \) be a finite dimensional vector space over \( \mathbb{F}_q \). The set of subspaces of \( V \) form the projective geometry \( \text{PG}(n - 1, q) \).

Let \( \pi \) be a projective space. A collineation of a projective space \( \pi \) is a bijective mapping from the points of \( \pi \) to themselves which preserves collinearity. That is, a collineation \( \varphi \) maps any three collinear points \( P, Q, R \) to another collinear triple \( \varphi(P), \varphi(Q), \varphi(R) \). The collineations form a group with respect to composition, the collineation group. If \( M \) is the matrix of an endomorphism, then \( \Psi_M \) is the induced map on projective space. By considering the homomorphism \( M \mapsto \Psi_M \), the group \( \text{GL}(n + 1, q) \) of invertible endomorphisms becomes a subgroup of the group of collineations of \( \text{PG}(n, q) \). This is the projectivity group \( \text{PGL}(n + 1, q) \). It is isomorphic to \( \text{GL}(n + 1, q)/\mathbb{F}_q^* \). Another source of collineations is this: Let \( \Phi \in \text{Aut}(\mathbb{F}_q) \) be a field automorphism. Then \( \Phi \) acts on projective space by sending \( P(x) \) to \( P(x\Phi) \). This map is another type of collineation, called automorphic collineation. This way, \( \text{Aut}(\mathbb{F}_q) \) gives rise to a group of collineations. If \( q = p^h \) for some prime \( p \) and some integer \( h \) then

\[ \Phi_0 : \mathbb{F}_q \to \mathbb{F}_q, \quad x \mapsto x^p \]

is a generator for the cyclic group \( C_h \simeq \text{Aut}(\mathbb{F}_q) \). The collineation group of \( \text{PG}(n, q) \) \((n \geq 2)\) is isomorphic to the semidirect product of the projectivity group and the automorphism group of the field. The collineation group is \( \text{PGL}(n + 1, q) = \text{PGL}(n + 1, q) \rtimes \text{Aut}(\mathbb{F}_q) \). We use the following notation for elements of \( \text{PGL}(n + 1, q) \). Let \( \Phi_0 \) be a generator for \( \text{Aut}(\mathbb{F}_q) \) and let \( M \in \text{GL}(n + 1, q) \). The map

\[ (\Psi_M, \Phi_0^k) : \text{PG}(n, q) \to \text{PG}(n, q), \quad P(x) \mapsto P(y), \quad y = (x \cdot M)^{\Phi_0^k} \]
is denoted as

\[ M_k. \] (5.1)

The identity element is \( I_0 \), where \( I \) is the identity matrix and 0 is the residue class modulo \( h \). The rules for multiplication and inversion in the collineation group are given as

\[ M_k \cdot N_l = (M \cdot N^{\Phi_{-k}})_{k+l}, \] (5.2)

\[ (M_k)^{-1} = (M^{-1})^{\Phi_k}_{-k}. \] (5.3)

The affine group \( AGL(n, q) \) is the semidirect product of \( GL(n, q) \) with \( F_q^n \). The affine semilinear group \( AGL(n, q) \) is the semidirect product of \( AGL(n, q) \) with \( \text{Aut}(F_q) \). The elements of \( AGL(n, q) \) are triples

\[ M_{a,k} := (M, a, k) \in \text{GL}(n, q) \times F_q^n \times \text{Aut}(F_q), \]

which act on \( F_q^n \):

\[ (x, (M, a, k)) \mapsto (x \cdot M + a)^{\Phi_k}. \]

The multiplication in \( AGL(n, q) \) is

\[ M_{a,k} \cdot N_{b,l} = (MN)_{a_N^{\Phi_{-k}} + b^{\Phi_{-k}}, k+l}. \]

The inverse of an element is

\[ (M_{a,k})^{-1} = (M^{-1})_{a^{\Phi_k}M^{-1}, -k}. \]

A correlation is a one-to-one mapping between the set of points and the set of hyperplanes which reverses incidence. So, if \( \rho \) is a correlation and \( P \) is a point and \( \ell \) is a hyperplane then \( P^\rho \) is a hyperplane and \( \ell^\rho \) is a point and

\[ \ell^\rho \in P^\rho \iff P \in \ell. \]

A correlation of order two is called polarity. The standard polarity is the map

\[ \rho: \mathcal{P} \leftrightarrow \mathcal{L}, \ P(x) \leftrightarrow [x]. \]

A group \( G \) can act on \( V \) in one of the types listed in Table 5.1. One can create a matrix group over a finite field \( F_q \) is created as described in in two steps. In the first step, the field \( F_q \) is created as described in Sections 3.2 and 3.3. The field is stored in the symbol table. Then, the group is created using the symbolic label for the field. The basic types of matrix groups in Orbiter are listed in Table 5.2.

For instance,
The Group PGL(4, 2)

The order of the group PGL(4, 2) is 20160
The group acts on a set of size 15

Table 5.1: Basic actions

<table>
<thead>
<tr>
<th>Type</th>
<th>Perm. Domain</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>General linear GL(n, q)</td>
<td>all vectors of V</td>
<td>q^n</td>
</tr>
<tr>
<td>Affine AGL(n, q)</td>
<td>all vectors of V</td>
<td>q^n</td>
</tr>
<tr>
<td>Projective PGL(n, q)</td>
<td>G r_1(V)</td>
<td>( \frac{q^n-1}{q-1} )</td>
</tr>
<tr>
<td>Wreath product GL(d, q)≀Sym(n)</td>
<td>G r_1((\mathbb{F}_q^d)^\otimes n) extended</td>
<td>n + nq^d + \frac{q^n-1}{q-1}</td>
</tr>
<tr>
<td>Orthogonal PGO(n, q)</td>
<td>Q(V)</td>
<td>( \frac{q^{n-1}-1}{q-1} )</td>
</tr>
<tr>
<td>Orthogonal PGO+(n, q)</td>
<td>Q+(V)</td>
<td>( \frac{(q^{n/2}-1)(q^{(n-2)/2}+1)}{q-1} )</td>
</tr>
<tr>
<td>Orthogonal PGO-(n, q)</td>
<td>Q-(V)</td>
<td>( \frac{(q^{n/2}+1)(q^{(n-2)/2}-1)}{q-1} )</td>
</tr>
</tbody>
</table>

PGL_4.2:

\$ORBITER\_PATH\$orbi.\text{t.out} -v.2 -d

\$define F -finite_field -q.2 -end -d

\$define G -linear_group -PGL.4.2 -end -d

\$with G -do -d

\$group_theoretic_activity -d

\$report -d

\$-end

pdflatex PGL_4.2_report.tex

open PGL_4.2_report.pdf

creates the group PGL(4, 2) acting on the 15 elements of \( \mathfrak{g} r_1(\mathbb{F}_2^4) \). At first, the field \( \mathbb{F}_2 \) is created. Secondly, the group \( G = PGL(3, 2) \) is created using the previously created field \( \mathbb{F}_2 \), and a report is generated. The report gives information about the permutation group action, including the underlying field and the projective geometry.
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>-GL</td>
<td>$n \ q$</td>
<td>$GL(n,q)$</td>
</tr>
<tr>
<td>-GGL</td>
<td>$n \ q$</td>
<td>$\Gamma L(n,q)$</td>
</tr>
<tr>
<td>-SL</td>
<td>$n \ q$</td>
<td>$SL(n,q)$</td>
</tr>
<tr>
<td>-SSL</td>
<td>$n \ q$</td>
<td>$\Sigma L(n,q)$</td>
</tr>
<tr>
<td>-PGL</td>
<td>$n \ q$</td>
<td>$PGL(n,q)$</td>
</tr>
<tr>
<td>-PGGL</td>
<td>$n \ q$</td>
<td>$P\Gamma L(n,q)$</td>
</tr>
<tr>
<td>-PSL</td>
<td>$n \ q$</td>
<td>$PSL(n,q)$</td>
</tr>
<tr>
<td>-PSSL</td>
<td>$n \ q$</td>
<td>$P\Sigma L(n,q)$</td>
</tr>
<tr>
<td>-AGL</td>
<td>$n \ q$</td>
<td>$AGL(n,q)$</td>
</tr>
<tr>
<td>-AGGL</td>
<td>$n \ q$</td>
<td>$A\Gamma L(n,q)$</td>
</tr>
<tr>
<td>-ASL</td>
<td>$n \ q$</td>
<td>$A SL(n,q)$</td>
</tr>
<tr>
<td>-ASSL</td>
<td>$n \ q$</td>
<td>$A\Sigma L(n,q)$</td>
</tr>
<tr>
<td>-PGO</td>
<td>$n \ q$</td>
<td>$PGO(n,q)$</td>
</tr>
<tr>
<td>-PGOp</td>
<td>$n \ q$</td>
<td>$PGO^+(n,q)$</td>
</tr>
<tr>
<td>-PGOm</td>
<td>$n \ q$</td>
<td>$PGO^-(n,q)$</td>
</tr>
<tr>
<td>-PGGO</td>
<td>$n \ q$</td>
<td>$P\Gamma O(n,q)$</td>
</tr>
<tr>
<td>-PGGOp</td>
<td>$n \ q$</td>
<td>$P\Gamma O^+(n,q)$</td>
</tr>
<tr>
<td>-PGGOm</td>
<td>$n \ q$</td>
<td>$P\Gamma O^-(n,q)$</td>
</tr>
<tr>
<td>-GL_d_q_wr_Sym_n</td>
<td>$d \ q \ n$</td>
<td>$GL(d,q) \wr \ Sym(n)$</td>
</tr>
</tbody>
</table>

Table 5.2: Basic types of Orbiter matrix groups
Strong generators for a group of order 20160:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix},
\begin{bmatrix}
0,0,0,0,1,0,0,0,1,0,0,1,0,0,0,1 \\
1,0,0,0,0,1,0,0,0,1,0,0,1,0,0,0 \\
0,1,0,0,0,0,0,1,0,0,1,0,0,0,1,0 \\
0,0,1,0,0,0,0,0,1,0,0,0,1,0,0,0 \\
0,0,0,1,0,0,0,0,0,1,0,0,0,1,0,0 \\
\end{bmatrix},
\begin{bmatrix}
1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0 \\
1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0 \\
1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0 \\
1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0 \\
0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0 \\
0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,1 \\
\end{bmatrix}
\]

The Action

Group action PGL(4,2) of degree 15
We act on the following set:

0 = ( 1, 0, 0, 0 )
1 = ( 0, 1, 0, 0 )
2 = ( 0, 0, 1, 0 )
3 = ( 0, 0, 0, 1 )
4 = ( 1, 1, 1, 1 )
5 = ( 1, 1, 0, 0 )
6 = ( 1, 0, 1, 0 )
7 = ( 0, 1, 1, 0 )
8 = ( 1, 1, 1, 0 )
9 = ( 1, 0, 0, 1 )
10 = ( 0, 1, 0, 1 )
11 = ( 1, 1, 0, 1 )
12 = ( 0, 0, 1, 1 )
13 = ( 1, 0, 1, 1 )
14 = ( 0, 1, 1, 1 )

The group is a matrix group.
The group acts on projective space PG(3,2)

q = 2
p = 2
e = 1
n = 3
Number of points = 15
Number of lines = 35
Number of lines on a point = 7
Number of points on a line = 3

The finite field $\mathbb{F}_2$

$$Z_i = \log_\alpha(1 + \alpha^i)$$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\gamma_i$</th>
<th>$-\gamma_i$</th>
<th>$\gamma_i^{-1}$</th>
<th>$\log_\alpha(\gamma_i)$</th>
<th>$\alpha^i$</th>
<th>$Z_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 = 0</td>
<td>0</td>
<td>DNE</td>
<td>DNE</td>
<td>1</td>
<td>DNE</td>
</tr>
<tr>
<td>1</td>
<td>1 = 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>DNE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\cdot$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$1^0 \equiv 1$
$1^1 \equiv 1$

Base and Stabilizer Chain

Group order 20160
tl=15, 14, 12, 8,
Base: (0, 1, 2, 3)

Strong generators for a group of order 20160:

$$\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}$$

$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,$
$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,$
$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1,$
$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1,$

132
Stabilizer chain

<table>
<thead>
<tr>
<th>Level</th>
<th>Base pt</th>
<th>Orbit length</th>
<th>Subgroup order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>15</td>
<td>20160</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1344</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>12</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Basic Orbit 0

Basic orbit 0 has size 15
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Basic Orbit 1

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

Basic Orbit 2

2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14
Basic orbit 3

Basic orbit 3 has size 8
3, 4, 9, 10, 11, 12, 13, 14

GAP export:

Generators in GAP format are:
G := Group([(4, 10)(5, 15)(11, 12)(13, 14),
            (4, 11)(5, 14)(10, 12)(13, 15),
            (4, 13)(5, 12)(10, 14)(11, 15),
            (3, 4)(7, 10)(8, 11)(9, 12),
            (2, 3)(6, 7)(11, 13)(12, 14),
            (1, 2)(7, 8)(10, 11)(14, 15)]);

Magma export:

G := GeneralLinearGroup(4, GF(2));
H := sub< G | [1,0,0,0, 0,1,0,0, 0,0,1,0, 1,0,0,1],
         [1,0,0,0, 0,1,0,0, 0,0,1,0, 0,1,0,1],
         [1,0,0,0, 0,1,0,0, 0,0,1,0, 0,0,1,1],
         [1,0,0,0, 0,1,0,0, 0,0,0,1, 0,0,1,0],
         [1,0,0,0, 0,0,1,0, 0,1,0,0, 0,0,0,1],
         [0,1,0,0, 1,0,0,0, 0,0,1,0, 0,0,0,1] >;

Compact form:

Generators in compact permutation form are:
6 15
0 1 2 9 14 5 6 7 8 3 11 10 13 12 4
0 1 2 10 13 5 6 7 8 11 3 9 14 4 12
0 1 2 12 11 5 6 7 8 13 14 4 3 9 10
0 1 3 2 4 5 9 10 11 6 7 8 12 13 14
The base has length 4
The basic orbits are:
Basic orbit 0 is orbit of 0 of length 15
Basic orbit 1 is orbit of 1 of length 14
Basic orbit 2 is orbit of 2 of length 12
Basic orbit 3 is orbit of 3 of length 8

The command

L_{5,3}:
\begin{verbatim}
 $(ORBITER_PATH)orbiter.out -v.2 -define F:finite_field -q.3 -end -define G:linear_group -PSL.5:F -end -with G: -do -group_theoretic_activity -report -end
\end{verbatim}
pdflatex PSL_{5,3}_report.tex
open PSL_{5,3}_report.pdf
\end{verbatim}

creates PSL(5, 3) of order 237783237120.

The command

PSP_{4,4}:
\begin{verbatim}
 $(ORBITER_PATH)orbiter.out -v.2 -define F:finite_field -q.4 -end -define G:linear_group -PGL.4:F -symplectic_group -end -with G: -do -group_theoretic_activity -report -end
\end{verbatim}
pdflatex PGL_{4,4}_Sp_{4,4}_report.tex
open PGL_{4,4}_Sp_{4,4}_report.pdf
\end{verbatim}

creates the symplectic group PSp(4, 4) of order 979200.

The command
PGO_5_2:
▷ $(\text{ORBITER\_PATH})\text{or}\text{b}\text{e}r.out\text{-v.2}\$
▷ ▷ $\text{-define F:\text{-finite_field:\text{-q.2\text{-end}}}\$}
▷ ▷ $\text{-define G:\text{-linear_group:\text{-PGO.5.F\text{-end}}}\$}
▷ ▷ $\text{-with G\text{-do}}\$
▷ ▷ $\text{-group_theoretic_activity}\$
▷ ▷ ▷ $\text{-report}\$
▷ ▷ $\text{-end}\$
▷ pdflatex PGO_5_2_report.tex
▷ open PGO_5_2_report.pdf

creates the group PGO(5, 2) acting on the 15 points of the $Q(4, 2)$ quadric. The following latex report is produced:

The Group PGO(5, 2)

The order of the group PGO(5, 2) is 720
The group acts on a set of size 15
Strong generators for a group of order 720:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}.\]
The Action

Group action PGO(5, 2) of degree 15
We act on the following set:

\[
\begin{align*}
0 &= (0, 1, 0, 0, 0) & 8 &= (0, 1, 1, 1, 1) \\
1 &= (0, 0, 1, 0, 0) & 9 &= (1, 1, 1, 0, 0) \\
2 &= (0, 0, 0, 1, 0) & 10 &= (1, 1, 1, 1, 0) \\
3 &= (0, 1, 0, 1, 0) & 11 &= (1, 1, 1, 0, 1) \\
4 &= (0, 0, 1, 1, 0) & 12 &= (1, 0, 0, 1, 1) \\
5 &= (0, 0, 0, 0, 1) & 13 &= (1, 0, 0, 1, 1) \\
6 &= (0, 1, 0, 0, 1) & 14 &= (1, 0, 1, 1, 1) \\
7 &= (0, 0, 1, 0, 1)
\end{align*}
\]

The group is a matrix group.
The base action is on projective space \( \text{PG}(4, 2) \)
\( q = 2 \)
\( p = 2 \)
\( e = 1 \)
\( n = 4 \)
Number of points = 31
Number of lines = 155
Number of lines on a point = 15
Number of points on a line = 3

The finite field \( \mathbb{F}_2 \)
\( Z_i = \log_\alpha (1 + \alpha^i) \)

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
i & \gamma_i & -\gamma_i & \gamma_i^{-1} & \log_\alpha (\gamma_i) & \alpha^i & Z_i \\
\hline
0 & 0 = 0 & 0 & DNE & DNE & 1 & DNE \\
1 & 1 = 1 & 1 & 1 & 1 & 1 & DNE \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c|}
+ & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0 \\
\hline
\cdot & 1 & \\
1 & 1 & \\
\hline
\end{array}
\]
Base and Stabilizer Chain

Group order 720

$t_{l}=15, 8, 3, 1, 1, 2,$

Base: $(0, 1, 2, 3, 4, 5)$

Strong generators for a group of order 720:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

Stabilizer chain

<table>
<thead>
<tr>
<th>Level</th>
<th>Base pt</th>
<th>Orbit length</th>
<th>Subgroup order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>15</td>
<td>720</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Basic Orbit 0

Basic orbit 0 has size 15
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

Basic Orbit 1

Basic orbit 1 has size 8
1, 4, 7, 8, 9, 10, 11, 14
Basic Orbit 2

Basic orbit 2 has size 3
2, 5, 12

Basic Orbit 3

Basic orbit 3 has size 1
3

Basic Orbit 4

Basic orbit 4 has size 1
4

Basic Orbit 5

Basic orbit 5 has size 2
5, 12
GAP export:
Generators in GAP format are:
\[ G := \text{Group}([\text{(6, 13)(7, 14)(8, 15)(9, 12)}, \text{(3, 13)(4, 14)(5, 15)(9, 11)}, \text{(2, 12)(3, 14)(4, 13)(8, 10)}, \text{(2, 8, 9, 10, 12, 15)(3, 14, 7)(4, 13, 6)(5, 11)}, \text{(1, 10)(4, 11)(7, 12)(9, 14)}, \text{(1, 7)(3, 5)(4, 9)(10, 12)(11, 14)(13, 15)])]; \]

Magma export:

Compact form:

Generators in compact permutation form are:
\[
\begin{align*}
6 & 15 \\
0 & 1 2 3 4 12 13 14 11 9 10 8 5 6 7 \\
0 & 1 12 13 14 5 6 7 10 9 8 11 2 3 4 \\
0 & 11 13 12 4 5 6 9 8 7 10 1 3 2 14 \\
0 & 7 13 12 10 3 2 8 9 11 4 14 5 6 1 \\
9 & 1 2 10 4 5 11 7 13 0 3 6 12 8 14 \\
6 & 1 4 8 2 5 0 7 3 11 13 9 14 10 12 \\
-1
\end{align*}
\]

The base has length 6

The basic orbits are:

- Basic orbit 0 is orbit of 0 of length 15
- Basic orbit 1 is orbit of 1 of length 8
- Basic orbit 2 is orbit of 2 of length 3
- Basic orbit 3 is orbit of 3 of length 1
- Basic orbit 4 is orbit of 4 of length 1
- Basic orbit 5 is orbit of 5 of length 2

The symplectic group PSp(6, 2) can be created using the following command:

```
PSP_6_2:
\[
\begin{align*}
\text{!! $(\text{ORBITER\_PATH})\text{orbiter.out}\!-\!v\!-\!2\!-\!} \\
\text{!! -define F\!-\!finite\!-\!field\!-\!q\!-\!2\!-\!end\!} \\
\text{!! -define G\!-\!linear\!\_\!group\!-\!PGL\!\_\!6\!\_\!F\!-\!symplectic\!\_\!group\!-\!end} \\
\text{!! -with G\!-\!do} \\
\text{!! -group\!\_\!theoretic\!activity} \\
\text{!! -report} \\
\text{!! -end}
\end{align*}
\]
The group PGO(7, 2), isomorphic to PSp(6, 2), can be created using the following command:

```
PGO_7_2:
  $(ORBITER_PATH)orbiter.out -v 2 \n  -define F -finite_field -q 2 -end \n  -define G -linear_group -PGO 7 F -end \n  -with G -do \n  -group_theoretic_activity \n  -report \n  -end
```

```
pdflatex PGO_7_2_report.tex
open PGO_7_2_report.pdf
```
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Janko1</td>
<td></td>
<td>first Janko group, needs PGL(7, 11)</td>
</tr>
<tr>
<td>-monomial</td>
<td></td>
<td>subgroup of monomial matrices</td>
</tr>
<tr>
<td>-diagonal</td>
<td></td>
<td>subgroup of diagonal matrices</td>
</tr>
<tr>
<td>-null_polarity_group</td>
<td></td>
<td>null polarity group</td>
</tr>
<tr>
<td>-symplectic_group</td>
<td></td>
<td>symplectic group</td>
</tr>
<tr>
<td>-singer</td>
<td>k</td>
<td>subgroup of index $k$ in the Singer cycle</td>
</tr>
<tr>
<td>-singer_and_frobenius</td>
<td>$k$</td>
<td>subgroup of index $k$ in the Singer cycle, extended by the Frobenius automorphism of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td>-borel_upper</td>
<td></td>
<td>Borel subgroup of upper triangular matrices</td>
</tr>
<tr>
<td>-borel_lower</td>
<td></td>
<td>Borel subgroup of lower triangular matrices</td>
</tr>
<tr>
<td>-identity_group</td>
<td></td>
<td>identity subgroup</td>
</tr>
<tr>
<td>-subgroup_from_file</td>
<td>$fl$</td>
<td>read subgroup from file $f$ and give it the label $l$</td>
</tr>
<tr>
<td>-orthogonal</td>
<td>$\epsilon$</td>
<td>orthogonal group $O^{\epsilon}(n,q)$, with $\epsilon \in {\pm 1}$ when $n$ is even</td>
</tr>
<tr>
<td>-subgroup_by_generators</td>
<td>$lons_1 \ldots s_n$</td>
<td>Generate a subgroup from generators. The label “l” is used to denote the subgroup; $o$ is the order of the subgroup, $n$ is the number of generators and $s_1, \ldots, s_n$ are the generators for the subgroup in vector form.</td>
</tr>
</tbody>
</table>

Table 5.3: Commands for creating subgroups

5.3 Subgroups

There are many ways to create subgroups of a group. Table 5.3 lists some commands to do so. For instance, the command

J1:

```
$\text{ORBITER}\_\text{PATH}/\text{orbiter.out}\_\text{v.3}\cdot$
\text{define}\cdot G\cdot \text{-linear_group}\cdot \text{-PGL}\cdot 7\cdot 11\cdot \text{-Janko1}\cdot \text{-end}$
\text{with}\cdot G\cdot \text{-do}$
\text{-group_theoretic_activity}$
\text{-report}$
\text{-end}$
\text{pdflatex PGL_7_11_Subgroup_Janko1_report.tex}$
\text{open PGL_7_11_Subgroup_Janko1_report.pdf}$
```

144
creates the first Janko group as a subgroup of $\text{PGL}(7, 11)$.

The command

```
PGL_3.11_singer:
> $(ORBITER_PATH)orbiter.out -v 3 \n> \> -define G -linear_group -PGL.3.11 -singer.19 -end \n> \> -with G -do \n> \> -group_theoretic_activity \n> \> -report \n> \> -end
> pdflatex PGL_3.11_Singer_3.11_19_report.tex
> open PGL_3.11_Singer_3.11_19_report.pdf
```

creates a subgroup of the Singer cycle of order 7. The Singer cycle in $\text{GL}(d, q)$ is a generator for a subgroup of order $q^d - 1$. It induces an element of order $\frac{q^d - 1}{q - 1}$ on the associated projective geometry $\text{PG}(d - 1, q)$. The additional integer parameter $k$ after the `-singer` command is used to create the subgroup of index $k$ of the Singer cycle.

The command

```
PGL_3.11_singer_and_frobenius:
> $(ORBITER_PATH)orbiter.out -v 3 \n> \> -define G -linear_group -PGL.3.11 -singer_and_frobenius.19 -end \n> \> -with G -do \n> \> -group_theoretic_activity \n> \> -report \n> \> -end
> pdflatex PGL_3.11_Singer_and_Frob3.11_19_report.tex
> open PGL_3.11_Singer_and_Frob3.11_19_report.pdf
```

creates a subgroup of index 19 of the Singer cycle of $\text{PG}(2, 11)$, extended by a group of order 3 that arises from the field extension $\mathbb{F}_3^{11}$ over $\mathbb{F}_{11}$. The group created by this command has order 21.

The quaternion group is a group of order 8 generated by the following matrices over $\mathbb{R}$:

$$
i = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad j = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}, \quad k = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.
$$

It is isomorphic to a subgroup of $\text{SL}(2, 3)$. The Orbiter command

```
quaternion:
> $(ORBITER_PATH)orbiter.out -v 30 \n> \> -define G -linear_group -SL.2.3 \n```

145
creates the group. The command produces the list of group elements shown below.

<table>
<thead>
<tr>
<th>Element</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/8</td>
<td>1</td>
</tr>
<tr>
<td>1/8</td>
<td>4</td>
</tr>
<tr>
<td>2/8</td>
<td>2</td>
</tr>
<tr>
<td>3/8</td>
<td>4</td>
</tr>
</tbody>
</table>

Element 0 / 8 of order 1:

\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]

(0)(1)(2)(3)(4)(5)(6)(7)(8)

Element 1 / 8 of order 4:

\[
\begin{bmatrix}
2 & 1 \\
1 & 1
\end{bmatrix}
\]

(0)(1,5,2,7)(3,4,6,8)

Element 2 / 8 of order 2:

\[
\begin{bmatrix}
2 & 0 \\
0 & 2
\end{bmatrix}
\]

(0)(1,2)(3,6)(4,8)(5,7)

Element 3 / 8 of order 4:

\[
\begin{bmatrix}
1 & 2 \\
2 & 2
\end{bmatrix}
\]

(0)(1,7,2,5)(3,8,6,4)
Element 4 / 8 of order 4:
\[
\begin{bmatrix}
1 & 1 \\
1 & 2
\end{bmatrix}
\]
(0)(1,4,2,8)(3,7,6,5)

Element 5 / 8 of order 4:
\[
\begin{bmatrix}
0 & 1 \\
2 & 0
\end{bmatrix}
\]
(0)(1,3,2,6)(4,5,8,7)

Element 6 / 8 of order 4:
\[
\begin{bmatrix}
2 & 2 \\
2 & 1
\end{bmatrix}
\]
(0)(1,8,2,4)(3,5,6,7)

Element 7 / 8 of order 4:
\[
\begin{bmatrix}
0 & 2 \\
1 & 0
\end{bmatrix}
\]
(0)(1,6,2,3)(4,7,8,5)

The group table is created as csv file:

```
Row,C0,C1,C2,C3,C4,C5,C6,C7
0,0,1,2,3,4,5,6,7
1,1,2,3,0,5,6,7,4
2,2,3,0,1,6,7,4,5
3,3,0,1,2,7,4,5,6
4,4,7,6,5,2,1,0,3
5,5,4,7,6,3,2,1,0
6,6,5,4,7,0,3,2,1
7,7,6,5,4,1,0,3,2
END
```

The group of the cube can be created over the field $\mathbb{F}_3$:
cube_group:
```
$ORBITER\_PATH$oriter.out: -v 3
```
The tetrahedral subgroup can be created as well:

```latex
\begin{verbatim}
tetra_group:
  $(ORBITER_PATH)orbiter.out-v-3-
  -define G=-linear_group-GL-3-3-
  -subgroup_by_generators "tetra"."12".2-
  -gens-
  -end-
  -with G=-do-
  -group_theoretic_activity-
  -print_elements.tex-
  -report-
  -end
\end{verbatim}
```

The Hesse group of order 216 extended by the automorphism group of the field can be created in PG(3,4)

```latex
\begin{verbatim}
GENERATORS_HESSE_GROUP="\n 3,0,0,0,3,0,0,0,3,0,\n 2,0,0,0,2,0,1,2,3,0,\n 1,0,0,0,1,0,0,1,1,1,\n```

148
The group has order 432.

The Weyl group of type $E_8$ can be generated as a subgroup of $GL(8,3)$ using the following command:

```plaintext
GENERATORS_WEYL_GROUP_E8="
-1,-1,-1,-1,0,0,0,0,\n0,0,1,0,0,0,0,0,\n1,0,0,0,0,0,0,0,\n0,0,1,0,0,0,0,0,\n0,1,0,1,0,0,0,0,\n0,0,0,0,0,0,1,0,\n0,0,0,0,0,0,1,0,\n-1,0,-1,-1,-1,-1,-1,-1,\n0,1,0,1,1,1,1,1,\n1,0,0,0,0,0,0,0,\n0,0,1,0,0,0,0,0,\n0,0,0,1,0,0,0,0,\n0,0,0,0,1,0,0,0,\n0,0,0,0,0,1,0,0,\n0,0,0,0,0,1,0,0"
```

Weyl_E8:
A latex report is generated in the file \texttt{GL\_8\_3\_Subgroup\_Weyl\_E8\_696729600\_report.tex}. This command uses generators found by Gabi Nebe:

5.4 Linear Groups, Advanced Topics

It is sometimes necessary to control the finite field that is used in the construction of a matrix group. For prime fields, this is not an issue. For extension fields, the choice of polynomial does matter, as the generators depend on specific choices made for the finite field. Magma and GAP use Conway polynomials, which are difficult to compute. Orbiter has a built-in table of primitive polynomials. As explained in Section 3.3, Orbiter allows to specify the polynomial that should be used to create the finite field. The next example shows an instance where choosing the polynomial is important. We are recreating a group from the electronic Atlas on finite simple groups [67].

The electronic Atlas of finite simple groups [67] lists generators for $U_3(3)$ as $3 \times 3$ matrices over the field $\mathbb{F}_9$ using the following short Magma [14] program:

```magma
F<w>:=GF(9);
x:=CambridgeMatrix(1,F,3,[
  "164",
  "506",
  "851"]);
y:=CambridgeMatrix(1,F,3,[
  "621",
  "784",
  "066"]);
G<x,y>:=MatrixGroup<3,F|x,y>;
```

The generators are given using the Magma command `CambridgeMatrix`, which allows for more efficient coding of field elements. The field elements are coded as base-3 integers (like in Orbiter) with respect to the Magma version of $\mathbb{F}_9$. The polynomial for $\mathbb{F}_9$ can be determined using the following Magma command (which can be typed into the Magma online calculator at [62])

```magma
F<w>:=GF(9);
print DefiningPolynomial(F);
```

which results in

```
$.1^2 + 2*$.1 + 2
```

which is the Magma way of printing the polynomial $X^2 + 2X + 2$. If $\alpha$ is a root of the polynomial over $\mathbb{F}_3$, then

$$\alpha^2 = \alpha + 1.$$

Regarding the coefficient vector of the polynomial $(1, 2, 2)$ as in integer written in base-3, we obtain

$$1 \cdot 3^2 + 2 \cdot 3 + 2 = 17.$$

The command

```
-finite_field -q 9 -override_polynomial "17" -end
```
can be used to create $\mathbb{F}_9$ using this polynomial. The command

```plaintext
-define F -finite_field -q 9 -override_polynomial "17" -end
```

creates a symbolic variable $F$ for this specific field $\mathbb{F}_9$. In order to create the linear group over this field, the command

```plaintext
-linear_group -PGL 3 F -end
```

can be used, where the second argument after the `-PGL` command references the field $\mathbb{F}_9$ that we just created through its symbolic name. The desired subgroup can now be created using the command

```plaintext
U_3.3:
  ▷ $(ORBITER_PATH)orbiter.out -v.3\n  ▷ -define F -finite_field -q 9 -override_polynomial "17" -end\n  ▷ -define G -linear_group -PGL 3 F\n  ▷ -subgroup_by_generators "U_3.3" "6048" -2\n  ▷ "1,6,4,5,0,6,8,5,1,\n  ▷ 6,2,1,7,8,4,0,6,6" -end\n  ▷ -with G -do\n  ▷ -group_theoretic_activity\n  ▷ -report\n  ▷ -end\n  ▷ pdflatex PGL_3_9_Subgroup_U_3_3_6048_report.tex\n  ▷ open PGL_3_9_Subgroup_U_3_3_6048_report.pdf
```

Group theoretic activities will be discussed in Section 5.6.

As an example of a large group, consider the Conway group Co3. Following [61], the group can be generated using two matrices of dimension 22 over $\mathbb{F}_2$. We use the makefile variables to give each generator in compact form. Then we define vectors for each of the generators. Finally, we concatenate the two generators to form one long vector of generators, which is passed to the `-subgroup_by_generators` command. Finally, we create a report for the group.

```plaintext
CONWAY_GEN1="110111000100001010000\n  111101011111010000010111\n  0000001000000100010101\n  11111011011001001110\n  010101000000010011101\n  0000010000000100010101\n  0010000000000100010101\n  001000011000000111111\n  11101001001101000100111
```

152
CONWAY

Co3:

$(ORBITER\_PATH)orbiter.out -v 6$

$\_define\_F:\_finite\_field\_q 2\_end$

$\_define\_g1:\_vector\_field\_F:\_format\_22\_compact\_$(CONWAY\_GEN1)$\_end$

$\_define\_g2:\_vector\_field\_F:\_format\_22\_compact\_$(CONWAY\_GEN2)$\_end$

$\_define\_gens:\_vector\_concatenate\_g1\_concatenate\_g2\_end$

$\_define\_G:\_linear\_group\_PGL 22 2$

$\_subgroup\_by\_generators\_Co3\_495766656000 2\_gens$
The next example creates the Ree group in 7 dimensions over the field \( \mathbb{F}_{27} \). Again, we use makefile variables to specify the two generators as 7 \( \times \) 7 matrices over \( \mathbb{F}_{27} \) and concatenate them, before passing them to the `-subgroup_by_generators` command.

```plaintext
Ree_gen1="21,5,1,6,17,1,1,3,13,5,21,6,6,18,21,3,21,21,22,6,14,14,18,1,5,13,6,7,3,3,2,1,24,16,3,17,3,22,10,16,24,26,21,21,6,18,20,2,5"
Ree_gen2="16,3,11,5,16,22,20,24,6,18,24,7,1,26,9,23,17,18,23,20,13,9,7,2,15,17,5,11,3,3,6,21,4,24,16,25,8,6,24,21,12,7,24,15,2,13,11,14,24"
```

Ree_27:

```plaintext
# $(ORBITER_PATH)orbiter.out -v 6
-define F: finite_field -q 27 -override_polynomial "34" -end
-define g1: vector -field F -format 7 -dense $(Ree_gen1) -end
-define g2: vector -field F -format 7 -dense $(Ree_gen2) -end
-define gens: vector -concatenate g1 -concatenate g2 -end
-define G: linear_group -PGL 7: F
-subgroup_by_generators "Ree_27" "1007344472" 2: gens:
- -end
- -with G: do
- group_theoretic_activity
- -report
- -end
```

154
### Table 5.4: Commands for creating new actions from old

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-wedge</td>
<td></td>
<td>action on the exterior square</td>
</tr>
<tr>
<td>-wedge_detached</td>
<td></td>
<td>action on the exterior square. Unlike -wedge, this command does not establish the homomorphism to the original group. Instead, the group is created as subgroup of the larger general linear group.</td>
</tr>
<tr>
<td>-PGL2OnConic</td>
<td></td>
<td>induced action of PGL(2, q) on the conic in the plane PG(2, q)</td>
</tr>
<tr>
<td>-subfield_structure_action</td>
<td>$s$</td>
<td>action by field reduction to the subfield of index $s$</td>
</tr>
<tr>
<td>-on_k_subspaces</td>
<td>$k$</td>
<td>induced action on $k$ dimensional subspaces</td>
</tr>
<tr>
<td>-on_tensors</td>
<td></td>
<td>induced action of GL($d$, q) $\wr$ Sym($n$) on the tensor space</td>
</tr>
<tr>
<td>-on_rank_one_tensors</td>
<td></td>
<td>induced action of GL($d$, q) $\wr$ Sym($n$) on the tensor space</td>
</tr>
<tr>
<td>-restricted_action</td>
<td>$s$</td>
<td>restricted action on the set $s$</td>
</tr>
</tbody>
</table>

#### 5.5 Induced Actions

It is possible to create new group actions from old. Table 5.4 lists Orbiter commands to do so. For instance, the command

```bash
T3_on_tensors:
    $ $(ORBITER_PATH)orbiter.out -v 4 \
    -define G: \
    -linear_group -GL_d_q_wr_Sym_n 2 2 3 -on_tensors -end: \
    -with G: -do: \
    -group_theoretic_activity: \
    -report: \
    -end \
    pdflatex GL_2_2_wreath_Sym3_report.tex \
    open GL_2_2_wreath_Sym3_report.pdf
```

creates the group GL(2, 2)$\wr$Sym(3) acting on the 255 elements of PG(7, 2) which are identified with the tensors of type (2, 2, 2) over $\mathbb{F}_2$. Elements of this group are denoted in the notation of the semidirect product. A vector of elements in the linear group is followed by a permutation of the components.
The Group $\text{GL}(2, 2) \wr \text{Sym}(3)$

The order of the group $\text{GL}(2, 2) \wr \text{Sym}(3)$ is 1296.
The group acts on a set of size 255.

The Action

Group action $\text{GL}(2, 2) \wr \text{Sym}(3)\text{res}255$ of degree 255.

Base and Stabilizer Chain

Group order 1296
$tl=3, 2, 1, 3, 2, 3, 2, 3, 2,$
Strong generators for a group of order 1296.

$$(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}; \text{id}), (\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}; \text{id}), (\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \text{id}),$$

$$(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; (1,2)), (\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; (0,1))$$

$0,1,2,0,1,0,1,0,0,1,1,0,1,1,,$
$0,1,2,1,0,0,1,1,0,0,1,1,0,1,0,,$
$0,1,2,1,0,0,1,1,0,1,1,1,0,0,1,,$
$0,1,2,1,0,0,1,1,0,1,1,1,0,0,1,,$
$0,1,2,1,0,0,1,1,0,0,1,1,0,0,1,,$
$0,1,2,1,0,1,1,1,0,0,1,1,0,0,1,,$
$0,2,1,1,0,0,1,1,0,0,1,1,0,0,1,,$
$1,0,2,1,0,0,1,1,0,0,1,1,0,0,1,$
### Stabilizer chain

<table>
<thead>
<tr>
<th>Level</th>
<th>Base pt</th>
<th>Orbit length</th>
<th>Subgroup order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1296</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>432</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>216</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>216</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>3</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

It is also possible to restrict the action on all rank-one tensors, as the following example shows:

T3r1:
```
-define G:
-linear_group -GL_dq_wr_Sym_n:2:2:3: -on_rank_one_tensors: -end:
-with G: -do:
-group_theoretic_activity:
-report:
-end
```
```
pdflatex GL_2_2_wreath_Sym3_report.tex
open GL_2_2_wreath_Sym3_report.pdf
```

This creates an action of degree 27. Because the degree is small, the Orbiter report shows all points in the permutation domain.

### The Group \( \text{GL}(2, 2) \ltimes \text{Sym}(3) \)

The order of the group \( \text{GL}(2, 2) \ltimes \text{Sym}(3) \) is 1296
The group acts on a set of size 27

### The Action

Group action \( \text{GL}(2, 2) \ltimes \text{Sym}(3) \)res27 of degree 27
We act on the following set:
The group of a conic is isomorphic to the group of the projective line. This isomorphism from the group of the projective line to the group of the conic can be realized using the command \texttt{PGL2OnConic}. The action is changed using the induced action on the Veronese.
variety. The group elements are still represented as $2 \times 2$ matrices. Here is an example. We create the collineation group $P\Gamma L(2, 8)$ of $PG(1, 8)$ and act on $PG(2, 8)$:

```
PGGL_2_8_on_conic:
  $\$(ORBITER_PATH)orbiter.out.-v.4.\$
  $\$ -define G:\$
  $\$ -linear_group-PGGL.2-8.-PGL2OnConic.-end.\$
  $\$ -with G.-do.\$
  $\$ -group_theoretic_activity.\$
  $\$ -report.\$
  $\$ -end
  pdflatex PGGL_2_8_OnConic_2_8_report.tex
  open PGGL_2_8_OnConic_2_8_report.pdf
```

This produces the following report. The generators are elements of $P\Gamma L(2, 8)$ acting on $PG(2, 8)$. The first basic orbit is the conic itself and all other basic orbits are subsets of it.

### The Group $P\Gamma L(2, 8)OnConic(2, 8)$

The order of the group $P\Gamma L(2, 8)OnConic(2, 8)$ is 1512

The group acts on a set of size 73

Strong generators for a group of order 1512:

\[
\begin{bmatrix}
  1 & 0 \\
  0 & 1
\end{bmatrix}, \begin{bmatrix}
  \gamma & 0 \\
  0 & 1
\end{bmatrix}, \begin{bmatrix}
  1 & 0 \\
  1 & 1
\end{bmatrix}, \begin{bmatrix}
  1 & 0 \\
  \gamma & 1
\end{bmatrix}, \begin{bmatrix}
  \gamma & 0 \\
  0 & 1
\end{bmatrix}, \begin{bmatrix}
  \gamma^2 & 1 \\
  1 & 0
\end{bmatrix}
\]

1,0,0,1,1,
1,0,0,6,0,
1,0,1,1,0,
1,0,2,1,0,
1,0,4,1,0,
0,1,1,0,0,

### The Action

Group action $P\Gamma L(2, 8)OnConic$ of degree 73

We act on the following set:
0 = (1, 0, 0) 5 = (2, 1, 0)
1 = (0, 1, 0) 72 = (7, 7, 1)
2 = (0, 0, 1) :
3 = (1, 1, 0)
4 = (1, 1, 0)

The group is a matrix group.
The base action is on projective space PG(1, 8)
$q = 8$
$p = 2$
$e = 3$
$n = 1$
Number of points = 9
Number of lines = 1
Number of lines on a point = 1
Number of points on a line = 9

The finite field $\mathbb{F}_8$

polynomial: $X^3 + X^2 + 1 = 13$
$Z_i = \log_\alpha(1 + \alpha^i)$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\gamma_i$</th>
<th>$-\gamma_i$</th>
<th>$\gamma_i^{-1}$</th>
<th>$\log_\alpha(\gamma_i)$</th>
<th>$\alpha^i$</th>
<th>$Z_i$</th>
<th>$\phi(\gamma_i)$</th>
<th>$T(\gamma_i)$</th>
<th>$N(\gamma_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>DNE</td>
<td>DNE</td>
<td>1</td>
<td>DNE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 = 1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha = \gamma$</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>$\alpha + 1 = \gamma^5$</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>$\alpha^2 = \gamma^2$</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>$\alpha^2 + 1 = \gamma^3$</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>$\alpha^2 + \alpha = \gamma^6$</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>$\alpha^2 + \alpha + 1 = \gamma^4$</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>DNE</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
\[ + \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 1 & 0 & 3 & 2 & 5 & 4 & 7 \\
2 & 2 & 3 & 0 & 1 & 6 & 7 & 4 \\
3 & 3 & 2 & 1 & 0 & 7 & 6 & 5 \\
4 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\
5 & 5 & 4 & 7 & 6 & 1 & 0 & 3 \\
6 & 6 & 7 & 4 & 5 & 2 & 3 & 0 \\
7 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\
\end{array} \]

\[ - \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
2 & 2 & 4 & 6 & 5 & 7 & 1 & 3 \\
3 & 3 & 6 & 5 & 1 & 2 & 7 & 4 \\
4 & 4 & 5 & 1 & 7 & 3 & 2 & 6 \\
5 & 5 & 7 & 2 & 3 & 6 & 4 & 1 \\
6 & 6 & 1 & 7 & 2 & 4 & 3 & 5 \\
7 & 7 & 3 & 4 & 6 & 1 & 5 & 2 \\
\end{array} \]

\[ 2^0 = 1 \quad \quad 2^5 = 3 \]
\[ 2^1 = 2 \quad \quad 2^6 = 6 \]
\[ 2^2 = 4 \quad \quad 2^7 = 1 \]
\[ 2^3 = 5 \]
\[ 2^4 = 7 \]

**Base and Stabilizer Chain**

Group order 1512

\( \text{tl}=9, 8, 7, 3, \)

**Stabilizer chain**

<table>
<thead>
<tr>
<th>Level</th>
<th>Base pt</th>
<th>Orbit length</th>
<th>Subgroup order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1512</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
<td>168</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Basic Orbit 0

Basic orbit 0 has size 9
0, 1, 2, 3, 4, 5, 6, 7, 8

Basic Orbit 1

Basic orbit 1 has size 8
1, 2, 3, 4, 5, 6, 7, 8
In the following example, we will demonstrate two types of induced actions. One is the action induced on $k$-dimensional subspaces. The second is the restricted action on an invariant subset. The example we show is related to cubic surfaces. At first, we create the Eckardt surface in $PG(3, 13)$ from the arc

$$\{0, 1, 2, 3, 43, 113\}.$$
Then we export the set of 45 tritangent planes to file and we produce a report about the surface and its automorphism group. The next command creates the stabilizer of the surface from the generators given in the report, creates the induced action on planes, and restricts the action to the 45 tritangent planes stored in the file. Here is the fill command sequence, including a makefile variable for the generators of the stabilizer of the surface:

```
SURFACE_q13_STAB="1,0,0,0,12,0,0,0,0,12,0,0,0,0,1,\
1,0,0,0,0,12,0,0,0,0,10,0,0,0,12,\
1,0,0,0,0,12,0,0,0,0,0,0,1,\
0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1"
```

```
surface_q13_stab_on_tritangents_orbits:
  $(ORBITER_PATH)/orbiter.out:-v.30:
  |   -define.F--finite_field-q.13:-end:
  |   -define.P--projective_space.3.F:-end:
  |   -with.P--do:
  |     -projective_space_activity:
  |       -define_surface.S--q.13:
  |       -arc_lifting:"0,1,2,3,43,113":-end:
  |     -end:
  |   -with.S--do:
  |   -cubic_surface_activity:
  |     -report_with_group:
  |     -end:
  |   -with.S--do:
  |   -cubic_surface_activity:
  |     -export_tritangent_planes:
  |     -end
  $(ORBITER_PATH)/orbiter.out:-v.2:
  |   -orbiter_path:$(ORBITER_PATH):
  |   -define.TriP--set--file:
  |     family_Eckardt_q13.a2_b1.tritangent_planes.csv:
  |     -end:
  |   -define.G--linear_group--PGL.4.13:
  |     -subgroup_by_generators."SURF_STAB":
  |     -24.4:$(SURFACE_q13_STAB):
  |     -end:
  |   -define.G.on_planes--modified_group--from.G:
  |     -on_k_subspaces.3:
  |     -end:
  |   -define.Gr--modified_group--from.G.on_planes:
  |     -restricted_action.TriP:
  |     -end:
```

164
5.6 Group Theoretic Activities

Once a group has been created as in Section 5.2, a group theoretic activity can be performed. For this purpose, Orbiter provides the -group_theoretic_activity option. Tables 5.5 and 5.6 list the possible commands that can come after it.

The command

\begin{verbatim}
PGL_3_2_elements:
 $\$(ORBITER_PATH)orbiter.out-v.5-
 $define G -linear_group-PGL-3-2-end-
 $with G-do-
 $group_theoretic_activity-
 $save_elements_csv "PGL_3_2_elements.csv"
$end
\end{verbatim}

creates all elements of PGL(3,2) and writes them into the file PGL_3_2_elements.csv.

The command

\begin{verbatim}
Sym_3_elements:
 $\$(ORBITER_PATH)orbiter.out-v.3-
 $define G -permutation_group-symmetric_group-3-end-
 $with G-do-
 $with G-do-
 $group_theoretic_activity-
 $print_elements_tex-
 $end-
 $\$(ORBITER_PATH)orbiter.out-v.20-
 $draw_options-
 $nodes-
 $embedded-radius.250-
 $xin.10000-yin.10000-
 $xout.1000000-yout.600000-
 $scale.0.3-line_width.1.0-
 $end-
 $tree_draw-filePerm3_elements_tree.txt-end
\end{verbatim}

creates a tree of the elements of Sym(3) (see Fig 5.2). The leaves are ordered lexicographically.

It is possible to compute all powers of a fixed element, as in the following command:
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-apply</td>
<td>(a\ s)</td>
<td>Applies the group element given by the coded vector (s) to the element (a).</td>
</tr>
<tr>
<td>-multiply</td>
<td>(s_1\ s_2)</td>
<td>Multiplies group elements (s_1) and (s_2), assuming the elements are given in coded form. Produces a latex report.</td>
</tr>
<tr>
<td>-inverse</td>
<td>(s)</td>
<td>Computes the inverse of (s), which is given in coded form. Produces a latex report.</td>
</tr>
<tr>
<td>-consecutive_powers</td>
<td>(s\ n)</td>
<td>Computes all powers (s^i) for (i = 1, \ldots, n). (s) is given in coded form. Produces a latex report.</td>
</tr>
<tr>
<td>-raise_to_the_power</td>
<td>(s\ n)</td>
<td>Computes the (n)-th power of (s), which is given in coded form. Produces a latex report.</td>
</tr>
<tr>
<td>-export_orbiter</td>
<td></td>
<td>Exports the group to Orbiter.</td>
</tr>
<tr>
<td>-export_gap</td>
<td></td>
<td>Exports the group to GAP [28].</td>
</tr>
<tr>
<td>-export_magma</td>
<td></td>
<td>Exports the group to Magma [14].</td>
</tr>
<tr>
<td>-search_element_of_order</td>
<td>(i)</td>
<td>Finds all elements of order (i) in the group ((i \in \mathbb{N})).</td>
</tr>
<tr>
<td>-element_rank</td>
<td>(s)</td>
<td>Determines the rank of the group element (s) in the given group. (s) is given in coded form.</td>
</tr>
<tr>
<td>-element_unrank</td>
<td>(r)</td>
<td>Produces the group element whose rank is (r).</td>
</tr>
<tr>
<td>-find_singer_cycle</td>
<td></td>
<td>Finds all Singer cycles whose matrix is a companion matrix.</td>
</tr>
<tr>
<td>-poset_classification_control</td>
<td></td>
<td>Poset classification options. The argument list must be terminated with (-end)</td>
</tr>
<tr>
<td>-classes_based_on_normal_form</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-group_table</td>
<td></td>
<td>Stores the group table as csv-file.</td>
</tr>
<tr>
<td>-report</td>
<td></td>
<td>Produce a latex report about the group.</td>
</tr>
<tr>
<td>-sylow</td>
<td></td>
<td>Include Sylow subgroups in the report (requires (-report)).</td>
</tr>
<tr>
<td>-print_elements</td>
<td></td>
<td>Produces a printout of all group elements.</td>
</tr>
<tr>
<td>-print_elements_tex</td>
<td></td>
<td>Produces a latex report of all group elements.</td>
</tr>
<tr>
<td>-order_of_products</td>
<td>(g_1 \ldots g_n)</td>
<td>Creates a table of the orders of all products (g_i g_j, 1 \leq i, j \leq n).</td>
</tr>
</tbody>
</table>

Table 5.5: Group theoretic activities (Part 1)
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>classify_arcs</code></td>
<td>description</td>
<td>Classify arcs in geometries. See Section 6.5.</td>
</tr>
<tr>
<td><code>linear_codes</code></td>
<td>$d \ n_{\text{max}}$</td>
<td>Classify linear codes with prescribed minimum distance $d$. Assumes that the group is $\text{PGL}(r,q)$ or $\text{PΓL}(r,q)$. For each $n \leq n_{\text{max}}$, the $[n,k,\geq d]$ codes are classified with $n - k = r$. See Section 10.</td>
</tr>
<tr>
<td><code>tensor_classify</code></td>
<td>$d$</td>
<td>Classifies tensors of tensor-rank at most $d$.</td>
</tr>
<tr>
<td><code>tensor_permutations</code></td>
<td></td>
<td>Computes the permutation representation of generators of wreath product.</td>
</tr>
<tr>
<td><code>reverse_iso</code></td>
<td></td>
<td>Given a set of generators of a subgroup of $\text{PGO}^+(6,q)$ as $6 \times 6$ matrixes, compute the inverse image of the generators in $\text{PGL}(4,q)$ (if possible).</td>
</tr>
<tr>
<td><code>classify_cubic_curves</code></td>
<td>descr</td>
<td>Classifies cubic curves. Expects an arc description options as in Table 6.4.</td>
</tr>
</tbody>
</table>

Table 5.6: Group theoretic activities (Part 2)

Figure 5.2: The elements of $\text{Sym}(3)$ in lex-order
We create the 12 powers of the cycle

\[(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11).\]

The output is

<table>
<thead>
<tr>
<th>(i)</th>
<th>((0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)^i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11))</td>
</tr>
<tr>
<td>2</td>
<td>((0, 2, 4, 6, 8, 10)(1, 3, 5, 7, 9, 11))</td>
</tr>
<tr>
<td>3</td>
<td>((0, 3, 6, 9)(1, 4, 7, 10)(2, 5, 8, 11))</td>
</tr>
<tr>
<td>4</td>
<td>((0, 4, 8)(1, 5, 9)(2, 6, 10)(3, 7, 11))</td>
</tr>
<tr>
<td>5</td>
<td>((0, 5, 10, 3, 8, 1, 6, 11, 4, 9, 2, 7))</td>
</tr>
<tr>
<td>6</td>
<td>((0, 6)(1, 7)(2, 8)(3, 9)(4, 10)(5, 11))</td>
</tr>
<tr>
<td>7</td>
<td>((0, 7, 2, 9, 4, 11, 6, 1, 8, 3, 10, 5))</td>
</tr>
<tr>
<td>8</td>
<td>((0, 8, 4)(1, 9, 5)(2, 10, 6)(3, 11, 7))</td>
</tr>
<tr>
<td>9</td>
<td>((0, 9, 6, 3)(1, 10, 7, 4)(2, 11, 8, 5))</td>
</tr>
<tr>
<td>10</td>
<td>((0, 10, 8, 6, 4, 2)(1, 11, 9, 7, 5, 3))</td>
</tr>
<tr>
<td>11</td>
<td>((0, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1))</td>
</tr>
<tr>
<td>12</td>
<td>(id)</td>
</tr>
</tbody>
</table>

The command

\begin{verbatim}
PGL_3_4_singer:
  $\$(ORBITER_PATH)orbiner.out\:v:5$
  $-\$define\:G\:-\$linear\:group\:-\$PGL\:3\:4\:-\$end$
\end{verbatim}
finds all Singer cycles in $\text{PGL}(3, 4)$ whose matrix is the companion matrix of a polynomial. The first one found is

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
2 & 3 & 2
\end{bmatrix}
\]

whose projective order is 21. Here, we are using the numeric form of field elements, so 2 is $\omega$ and 3 is $\omega + 1$.

Suppose we want to multiply two elements in a group. The following command shows an example in $\text{GL}(2, 8)$. We multiply the elements coded by 0,1,2,3 and 4,5,6,7:

```
GL_2_8_multiply:
$ (\text{ORBITER\_PATH})\text{or} \text{b} \text{er} .\text{o} \text{u} \text{t} -v -5 \backslash$
$ -d\text{e} \text{f} \text{i} \text{n} \text{e} \text{G} -\text{linear\_group} -\text{G} \text{L} .2 -8 -\text{e} \text{n} \text{d} \backslash$
$ -d\text{o} \backslash$
$ -g\text{roup\_theoretic\_activity} \backslash$
$ -m\text{i} \text{l} \text{t} \text{i} \text{p} \text{y} -1 \text{0} , 2, 3 \text{"} 4, 5, 6, 7 \text{"} \backslash$
$ -e\text{nd} \backslash$
pdflatex GL_2_8_mult.tex
open GL_2_8_mult.pdf
```

The output is

\[
\begin{bmatrix}
0 & 1 \\
\gamma & \gamma^5
\end{bmatrix}
\begin{bmatrix}
\gamma^2 & \gamma^3 \\
\gamma^6 & \gamma^4
\end{bmatrix} =
\begin{bmatrix}
\gamma^6 & \gamma^4 \\
\gamma & \gamma^5
\end{bmatrix}
\]

0,1,2,3, 4,5,6,7, 6,7,2,3,

Note that the output shows the codings of the three group elements. This way, the result of this computation can be processed further easily. The same example over $\mathbb{F}_7$, noting that $7 \equiv 0 \pmod{7}$ is:
GL\_2\_7\_multiply:
\[
\text{\$\{ORBITER\_PATH\}orbiter.out\,-v\,-p\,-\text{-v}5\,-\text{-end}\,\text{-end}\}
\]
\[
\text{-define\,G\,linear\,group\,-GL\,2\,7\,-end}\,,
\]
\[
\text{-with\,G\,-do}\,,
\]
\[
\text{-group\,theoretic\,activity}\,\text{-do}\,,
\]
\[
\text{-multiply\,"0,1,2,3","4,5,6,0"}\,,
\]
\[
\text{-end}\,,
\]
\[
pdflatex\,\text{GL\,2\,7\,mult\,tex}\,,
\]
\[
\text{open\,GL\,2\,7\,mult.pdf}\,
\]

The output is

\[
\begin{bmatrix}
0 & 1 \\
2 & 3
\end{bmatrix}
\begin{bmatrix}
4 & 5 \\
6 & 0
\end{bmatrix}
= \begin{bmatrix}
6 & 0 \\
5 & 3
\end{bmatrix}
\]

0,1,2,3,
4,5,6,0,
6,0,5,3,

We can compute the inverse of a group element:

GL\_2\_7\_inv:
\[
\text{\$\{ORBITER\_PATH\}orbiter.out\,-v\,-p\,-\text{-v}5\,-\text{-end}\,\text{-end}\}
\]
\[
\text{-define\,G\,linear\,group\,-GL\,2\,7\,-end}\,,
\]
\[
\text{-with\,G\,-do}\,,
\]
\[
\text{-group\,theoretic\,activity}\,\text{-do}\,,
\]
\[
\text{-inverse\,"0,1,2,3"}\,,
\]
\[
\text{-end}\,,
\]
\[
pdflatex\,\text{GL\,2\,7\,inv\,tex}\,,
\]
\[
\text{open\,GL\,2\,7\,inv.pdf}\,
\]

The output is

\[
\begin{bmatrix}
0 & 1 \\
2 & 3
\end{bmatrix}^{-1}
= \begin{bmatrix}
2 & 4 \\
1 & 0
\end{bmatrix}
\]

0,1,2,3,
2,4,1,0,
We can raise a group element to a power:

GL_2.7_power:

\[
\begin{array}{l}
\text{\verb|\$|(ORBITER\_PATH)orbiter.out\-v.5\textbar
\text{\verb|\> \> -define\-G::linear\_group\-GL\-2\-7\-end\-\textbar
\text{\verb|\> \> -with\-G::do\-\textbar
\text{\verb|\> \> -group\_theoretic\_activity\-\textbar
\text{\verb|\> \> \> -raise\_to\_the\_power\"0,1,2,3\"\-2\-\textbar
\text{\verb|\> \> \> -end
\text{\verb|\> pdflatex\-GL\_2\_7\_power.tex
\text{\verb|\> open\-GL\_2\_7\_power.pdf}
\end{array}
\]

The output is

\[
\begin{bmatrix}
0 & 1 \\
2 & 3
\end{bmatrix}^2 = \begin{bmatrix}
2 & 3 \\
6 & 4
\end{bmatrix}
\]

0,1,2,3,
2,3,6,4,
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-classes</td>
<td></td>
<td>Compute a report of the conjugacy classes of elements.</td>
</tr>
<tr>
<td>-centralizer_of_element</td>
<td>label coding</td>
<td>Compute the centralizer of the coded group element, using label to create file names.</td>
</tr>
<tr>
<td>-normalizer_of_cyclic_subgroup</td>
<td>label s</td>
<td>Compute the normalizer of the cyclic subgroup generated by the element s.</td>
</tr>
<tr>
<td>-normalizer</td>
<td></td>
<td>Compute the normalizer of a subgroup in the larger group.</td>
</tr>
</tbody>
</table>

Table 5.7: Group theoretic activities based on Magma

### 5.7 Group Theoretic Activities Based on Magma

Through its interface to Magma [14], Orbiter can perform group theoretic computations. Table 5.7 list the group theoretic commands that rely on Magma. The communication to and from magma happens through files. This is a three step process: An Orbiter session receives a command to compute the conjugacy classes of a group. The Orbiter session writes a magma file. This file is read and executed by Magma. Magma writes a second file containing the conjugacy classes in coded form. Another Orbiter session reads the magma output file, decodes the information and produces the desired list of conjugacy classes. A latex report is written containing the classes, as well as related information regarding centralizers and normalizers.

For instance, the three-step command sequence

```
PGGL_2_4_classes:  
  $ (ORBITER_PATH) orbiter.out -v 3 \  
  $ (MAGMA_PATH) / magma_PGGL_2_4_classes.magma  
```

173
computes the classes of elements in $\text{PGL}(2, 4)$ using Orbiter-Magma-Orbiter. The first Orbiter command produces the file $\text{PGGL}_2_4\_classes.magma$. The magma command reads this file and produces the file $\text{PGGL}_2_4\_classes\_out.txt$. The second Orbiter command reads the file $\text{PGGL}_2_4\_classes\_out.txt$ and produces the latex report $\text{PGGL}_2_4\_classes\_out.tex$.

The report produced by Orbiter is too long to be reproduced here fully. Let us look at just one conjugacy class. Here is the output for class $1/7$ (numbering starts from 0, so this is the second class):

Order of element = 2  
Class size = 10  
Centralizer order = 12  
Normalizer order = 12  
Representing element is

$$c_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

of order 2 and with 3 fixed points. 0,1,1,0,1,1,0,0,2,1,0,1,1,0,1,

The normalizer is generated by:

Strong generators for a group of order 12:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \omega^2 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

1,0,0,1,1, 
1,0,0,2,1, 
0,1,1,0,1, 

The command sequence

```
$\text{pdflatex}\_\text{PGGL}\_2\_4\_classes\_out.tex$
```

```
$\text{open}\_\text{PGGL}\_2\_4\_classes\_out.pdf$
```

```
PGGL\_2\_4\_cent\_2A:
```

```
$\text{eval}(\text{ORBITER\_PATH})\text{orbiter.out}\_\text{-v}\_3\_\text{-}\$
```

```
$\text{-define}\_\text{G}\_\$
```

```
$\text{-linear}\_\text{group}\_\text{-PGGL\_2\_4\_end}\_\$
```

```
$\text{-with}\_\text{G}\_\text{-do}\_\$
```

```
$\text{-group}\_\text{theoretic}\_\text{activity}\_\$
```

```
$\text{-centralizer}\_\text{of}\_\text{element}\_2A\_1,0,0,1,1,1,0,0,2,1,0,1,1,0,1,\$
```

174
computes the centralizer of the Baer involution
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

The centralizer is a group of order 40320, isomorphic to \( \text{PGL}(4,2).Z_2 \). Orbiter produces a list of strong generators, shown below:

\[
\begin{align*}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}, & \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1
\end{bmatrix}, & \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}, & \begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}, & \begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1
\end{bmatrix}, & \begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1
\end{bmatrix}
\end{align*}
\]
The end of the report has a list of generators in coded form. This list can be used to create the centralizer in Orbiter.

Orbiter can compute the normalizer of a cyclic subgroup. For instance, the element

$$\sigma = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 4 \end{bmatrix}$$

generates a cyclic subgroup of PGL(4,5) of order 31. The command

```
PGL_4_5_norm_31:
\$ $(ORBITER_PATH)orbiter.out -v 6 -define G:\
\$ linear_group -PGL 4 5 -end \$
\$ with G -do \$
\$ group_theoretic_activity \$
\$ \$ -normalizer_of_cyclic_subgroup "31" \$
\$ "2,0,0,0,0,1,0,0,0,0,1,0,3,0,4" \$
\$ \$ -end
\$ pdflatex-normalizer_of_31_in_PGL_4_5.tex
\$ open-normalizer_of_31_in_PGL_4_5.pdf
```

computes the normalizer, which is a group of order 372. The following report is produced:

```
The subgroup generated by

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

has order 31

The normalizer has order 372
```
Strong generators for a group of order 372:

\[
\begin{bmatrix}
4 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
4 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 3 & 4 \\
0 & 2 & 3 & 1
\end{bmatrix},
\begin{bmatrix}
2 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 \\
0 & 0 & 0 & 2 \\
0 & 2 & 2 & 1
\end{bmatrix}
\]

1,0,0,0,0,4,0,0,0,0,0,4,0,0,0,0,4,
1,0,0,0,0,3,0,0,0,0,3,0,0,0,0,3,
1,0,0,0,0,4,0,0,0,0,2,1,0,3,2,4,
1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,3,

For general normalizers, the group must be constructed as a subgroup $H$ of a larger group $G$ containing $H$. Then, the normalizer of $H$ in $G$ is computed. Consider this example. The group

\[ H = \langle \begin{bmatrix} \alpha^4 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \rangle \simeq C_2 \times C_2 \]

is a subgroup of $G = \text{PGL}(2,9)$. To compute the normalizer of $H$ in $G$, the following command sequence can be used:

```
Normalizer_of_Z22_in_PGL_2.9:
\$\$(ОРБИТЕР_PATH)orbit.out\$\$ -v 2
\$define G -linear_group -PGL_2.9$
\$-subgroup_by_generators Z22 4 2$
\$"2,0,0,1,0,1,1,0" -end$
\$-with G -do$
\$-group_theoretic_activity$
\$-normalizer$
\$-end$
```

It produces a report showing that the normalizer is a group of order 24 (it is isomorphic to $\text{Sym}(4)$, though the report does not tell us this fact directly):
The group $\text{PGL}(2,9)\text{Subgroup} Z_2$ of order 4 is:

Strong generators for a group of order 4:

\[
\begin{bmatrix}
\alpha^4 & 0 \\
0 & 1
\end{bmatrix},
\begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]

1,0,0,2,
0,1,1,0,

Inside the group of order 720, the normalizer has order 24:

Strong generators for a group of order 24:

\[
\begin{bmatrix}
\alpha^4 & 0 \\
0 & 1
\end{bmatrix},
\begin{bmatrix}
\alpha^2 & 0 \\
0 & 1
\end{bmatrix},
\begin{bmatrix}
\alpha^4 & \alpha^4 \\
\alpha^4 & 1
\end{bmatrix},
\begin{bmatrix}
\alpha^4 & \alpha^6 \\
\alpha^2 & 1
\end{bmatrix}
\]

1,0,0,2,
1,0,0,5,
1,1,1,2,
1,7,5,2,
Chapter 6

Orbit Algorithms

6.1 Schreier Trees

Orbiter provides several different orbit algorithms. The most basic orbit algorithm uses Schreier trees. It is explained in [17, 35, 59]. This algorithm has memory and time complexity proportional to the size of the orbit. It therefore is limited to small problems. More elaborate algorithms exist, provided the set on which we act has additional structure. These algorithms offer sublinear complexity. Orbiter offers posets based algorithms, which exploit an underlying poset structure. They will be discussed in Section 6.2. Orderly generation using canonical forms is discussed in Section 15.2.

The commands discussed in this section are group theoretic activities, see Table 6.1.

Consider the wreath product acting on rank-one tensors from Section 5.5. The following command sequence computes the orbits, exports the Schreier tree, and produces the drawing shown in Figure 6.1.

```
T3r1_orbits:
\$ $(ORBITER_PATH)orbiter.out\ -v\-4\$
\$define G\$
\$ linear_group \-GL\_d\_q\_wr\_Sym\_n\_2\_2\_3\$
\$on\_rank\_one\_tensors \-end\$
\$with\_G \-do\$
\$group\_theoretic\_activity\$
\$report\$
\$orbits\_on\_points\$
\$export\_trees\$
\$end$
```

In the next example, we compute the orbits of the linear group PGL(4, 2) on homogeneous polynomials of degree 3 in 4 variables:
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-orbits_on_subsets</td>
<td>$k$</td>
<td>Compute orbits on $k$-subsets.</td>
</tr>
<tr>
<td>-orbits_on_points</td>
<td></td>
<td>Compute orbits in the action that was created.</td>
</tr>
<tr>
<td>-orbits_of</td>
<td>$i$</td>
<td>Compute orbit of point $i$ in the given action.</td>
</tr>
<tr>
<td>-stabilizer</td>
<td></td>
<td>Compute the stabilizer of the orbit representative (needs -orbits_on_points).</td>
</tr>
<tr>
<td>-orbits_on_set_system_from_file</td>
<td>fname $f$ $l$</td>
<td>Reads the csv file “fname” and extract sets from columns $[f, ..., f + l - 1]$.</td>
</tr>
<tr>
<td>-orbit_of_set_from_file</td>
<td>fname</td>
<td>Reads a set from the text file “fname” and computes orbits on the elements of the set.</td>
</tr>
<tr>
<td>-orbits_on_polynomials</td>
<td>$d$</td>
<td>Computes the orbits of the matrix group on homogeneous polynomials of degree $d$. The number of variables is determined by the degree of the matrix group.</td>
</tr>
<tr>
<td>-conjugacy_class_of</td>
<td>label $s$</td>
<td>Compute the conjugacy class of the group element encoded as $s$ using the given label for file-names. Write a file containing the ranks for all elements in the class. Writes a second file containing the transporter elements for each element in the class. A transporter element maps the class representative to the given element under conjugation.</td>
</tr>
<tr>
<td>-orbits_on_group_elements_under_conjugation</td>
<td>fname-C fname-T</td>
<td>Under the centralizer of the class representative, construct the orbits on the class. For each non-trivial orbit, test whether the group generated by it and the class representative is Klein-four and all nontrivial elements are from the given class. If so, classify these groups and compute the normalizers. The arguments fname-C and fname-T are the files containing the elements of the class and the transporter, respectively.</td>
</tr>
</tbody>
</table>

Table 6.1: Basic Orbit algorithms
orbits_cubic_curves_q2:
\[\text{lisp} \text{ (ORBITER\_PATH)orbiter.out-v.4} \\]
\[\text{lisp} \quad \text{define:G.linear\_group:-PGL\_3.2.-end} \\]
\[\text{lisp} \quad \text{with:G.-do} \\]
\[\text{lisp} \quad \text{-group\_theoretic\_activity} \\]
\[\text{lisp} \quad \text{-orbits\_on\_polynomials.3} \\]
\[\text{lisp} \quad \text{-end} \\]
\[\text{lisp} \quad \text{pdflatex.poly\_orbits.d3.n3.q2.tex} \\]
\[\text{lisp} \quad \text{open-poly\_orbits.d3.n3.q2.pdf} \\]

This command computes the orbits of on all cubic forms in 4 variables, confirming the work of Dickson [23] and an enumerative result of Cooley [20].

The next example computes orbits in an induced action. Induced actions have been described in Section 5.5. One group can have many actions. In particular, Orbiter can work with induced actions without changing the representation of the group elements. This has the advantage that the stabilizers are expressed in terms of the original action. To consider an example, suppose we want to consider the action of the stabilizer of a conic on the points of the plane (this continues an example from Section 5.5). The following command can be used:

PGGL_2.8.on_conic_orbits:
\[\text{lisp} \quad \text{(ORBITER\_PATH)orbiter.out-v.4} \\]
\[\text{lisp} \quad \text{define:G} \\]
\[\text{lisp} \quad \text{-linear\_group:-PGGL\_2.8.-PGL2\_on\_Conic.-end} \\]
\[\text{lisp} \quad \text{-with:G.-do} \\]
The output shown below. First, the orbits are listed. Then for each orbit, the stabilizer is shown, together with the generators in the action on the plane. For the sake of space, some of the output has been shortened. The three orbits correspond to the conic, the nucleus and the remaining points of the plane.

**Group Orbits**

Orbits of the group $\text{PGL}(2,8)\text{OnConic}$:

Strong generators for a group of order 1512:

\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}, \begin{bmatrix}
\gamma & 0 \\
0 & 1
\end{bmatrix}, \begin{bmatrix}
1 & 0 \\
1 & 1
\end{bmatrix}, \begin{bmatrix}
1 & 0 \\
\gamma & 1
\end{bmatrix}, \begin{bmatrix}
1 & 0 \\
\gamma^2 & 1
\end{bmatrix}, \begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]

\[1,0,0,1,1,\]
\[1,0,0,6,0,\]
\[1,0,1,1,0,\]
\[1,0,2,1,0,\]
\[1,0,4,1,0,\]
\[0,1,1,0,0,\]

Considering the orbit length, there are 3 types of orbits:

\[(1,9,63)\]

\[i : \text{orbit length} : \text{number of orbits}\]
\[0 : 1 : 1\]
\[1 : 9 : 1\]
\[2 : 63 : 1\]

Orbits classified:
Set 0 has size 1 : \{1\}
Set 1 has size 1 : \{0\}
Set 2 has size 1 : \{2\}
Orbits of length 1:
Orbit 1: (1)

0 : 1 = ( 0, 1, 0 )

Orbits of length 9:
Orbit 0: ( 0, 2, 3, 29, 48, 38, 55, 60, 67 )

0 : 0 = ( 1, 0, 0 )
1 : 2 = ( 0, 0, 1 )
2 : 3 = ( 1, 1, 1 )
3 : 29 = ( 4, 2, 1 )
4 : 48 = ( 7, 4, 1 )
5 : 38 = ( 5, 3, 1 )
6 : 55 = ( 6, 5, 1 )
7 : 60 = ( 3, 6, 1 )
8 : 67 = ( 2, 7, 1 )

Orbits of length 63:
Orbit 2: ( 4, 5, 18, 7, 57, 25, 11, 37, 56, 10, 8, 33, 66, 45, 32, 41, 34, 14, 64, 9, 30, 47, 68, 52, 59, 71, 62, 6, 49, 65, 26, 21, 72, 54, 39, 13, 20, 43, 70, 50, 61, 17, 22, 44, 35, 23, 46, 40, 51, 28, ...12, 31, 16 )

0 : 4 = ( 1, 1, 0 )
1 : 5 = ( 2, 1, 0 )
2 : 18 = ( 0, 1, 1 )
3 : 7 = ( 4, 1, 0 )

Orbits of length 1:
Orbit 1: (1)

Stabilizer of orbit representative 1:
Strong generators for a group of order 1512:

\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
\gamma & 0 \\
0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
1 & 0 \\
\gamma & 1 \\
\end{bmatrix},
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\end{bmatrix}
\]

1,0,0,1,1,
1,0,0,6,0,
1,0,2,1,0,
Generator 0 / 4 is:
\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]

Generator 1 / 4 is:
\[
\begin{bmatrix}
\gamma & 0 \\
0 & 1
\end{bmatrix}
\]

Generator 2 / 4 is:
\[
\begin{bmatrix}
1 & 0 \\
\gamma & 1
\end{bmatrix}
\]

Generator 3 / 4 is:
\[
\begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]

Orbits of length 9:
Orbit 0: \((0, 2, 3, 29, 48, 38, 55, 60, 67)\)
Stabilizer of orbit representative 0:
Strong generators for a group of order 168:
\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}, \begin{bmatrix}
\gamma^6 & 0 \\
0 & 1
\end{bmatrix}, \begin{bmatrix}
\gamma^4 & 0 \\
0 & 1
\end{bmatrix}, \begin{bmatrix}
\gamma^2 & 0 \\
0 & 1
\end{bmatrix}
\]

1, 0, 0, 1, 1,
1, 0, 0, 2, 0,
Generator 0 / 3 is:
\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix},
\]

Generator 1 / 3 is:
\[
\begin{bmatrix}
\gamma^6 & 0 \\
0 & 1
\end{bmatrix},
\]

Generator 2 / 3 is:
\[
\begin{bmatrix}
\gamma^4 & 0 \\
\gamma^2 & 1
\end{bmatrix},
\]

Orbits of length 63:
Orbit 2: (4, 5, 18, 7, 57, 25, 11, 37, 56, 10, 8, 33, 66, 45, 32, 41, 34, 14, 64, 9, 30, 47, 68, 52, 59, 71, 62, 6, 49, 65, 26, 21, 72, 54, 39, 13, 20, 43, 70, 50, 61, 17, 22, 44, 35, 23, 46, 40, 51, 28, ...12, 31, 16)
Stabilizer of orbit representative 4:
Strong generators for a group of order 24:
\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 \\
\gamma^5 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 \\
\gamma^3 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 \\
\gamma & 1
\end{bmatrix}
\]
Generator 0 / 4 is:
\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]
as permutation: 

Generator 1 / 4 is:
\[
\begin{bmatrix}
1 & 0 \\
\gamma^5 & 1
\end{bmatrix}
\]
as permutation: 

Generator 2 / 4 is:
\[
\begin{bmatrix}
1 & 0 \\
\gamma^3 & 1
\end{bmatrix}
\]
as permutation: 

Generator 3 / 4 is:
\[
\begin{bmatrix}
1 & 0 \\
\gamma & 1
\end{bmatrix}
\]
as permutation: 
6.2 Poset Classification

A partially ordered set (poset) is a set together with a partial order. For instance, the set of subsets of a fixed set form an order structure with respect to set-inclusion. The Hasse diagram is a diagram whose nodes represent the element. Nodes are arranged from top to bottom, and relations are indicated by lines. Transitivity is implied. For instance, Figure 6.2 shows the power set lattice of a four-element subset.

Posets often come with group actions. We say that a group \( G \) acts on a poset \( P \) if for all \( x, y \in P \) and all \( g \in G \),

\[
x \leq y \Rightarrow xg \leq yg.
\]

For background on poset actions, see Plesken [54]. The orbits of \( G \) on \( P \) form another poset, the poset of orbits. The problem of classification of combinatorial objects can often be attacked by using group invariant relations. A layered poset can be decomposed into a series of relations. The layers allow to reduce the classification problem into small steps, namely from on layer to the next. This uses the incidence relation between adjacent layers. By iterating this method, one can form a poset of substructures, and the classification problem reduces to the problem of determining the orbits of the poset, which we will henceforth call the poset classification problem. Many classification problem in Combinatorics reduce to poset classification problems.

Orbiter uses the algorithm of Schmalz [58] to perform poset classification. Two versions are available: one for subset-type posets and one for subspace-type posets. Figure 6.3 shows the subspace lattice of \( V(3, 2) = F_2^3 \). The basis elements are listed, using the enumerator for elements on the projective geometry \( PG(2, 2) \) explained in Section 4.1.
The commands shown in Tables 6.2-6.3 can be used to control the poset classification algorithm. By default, Orbiter will choose the lexicographically least orbit representatives. It is possible to direct Orbiter to choose different orbit representatives. To this end, the nodes in the orbit tree are labeled. The node number is the zero-based number of a given node in the tree, using the breadth first ordering.

Suppose that Orbiter chooses element \( a \) at node \( n \). Suppose we are interested in choosing element \( b \) instead. The command

\[-\text{preferred_choice } n \ a \ b\]

can be used to force Orbiter to choose \( b \) instead of \( a \) at node \( n \).
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-problem_label</td>
<td>str</td>
<td>Use str as a prefix for files that are created.</td>
</tr>
<tr>
<td>-path</td>
<td>p</td>
<td>Use path p for files that are created.</td>
</tr>
<tr>
<td>-depth</td>
<td>d</td>
<td>Set search depth to d.</td>
</tr>
<tr>
<td>-draw_options</td>
<td>options</td>
<td>Drawing options according to Table 16.2.</td>
</tr>
<tr>
<td>-v</td>
<td>v</td>
<td>Set verbosity to v. Larger numbers mean more output.</td>
</tr>
<tr>
<td>-gv</td>
<td>v</td>
<td>Set verbosity for group theoretic operations to v. Larger numbers mean more output.</td>
</tr>
<tr>
<td>-recover</td>
<td>fname</td>
<td>Recover from the given file.</td>
</tr>
<tr>
<td>-lex</td>
<td></td>
<td>Use the lexicographic ordering to speed up the search.</td>
</tr>
<tr>
<td>-w</td>
<td></td>
<td>Save orbits at level d only.</td>
</tr>
<tr>
<td>-W</td>
<td></td>
<td>Save orbits at all levels.</td>
</tr>
<tr>
<td>-write_data_files</td>
<td></td>
<td>Save data to files.</td>
</tr>
<tr>
<td>-t</td>
<td></td>
<td>Write a file containing the search tree at level d.</td>
</tr>
<tr>
<td>-T</td>
<td></td>
<td>Write a file containing the search tree at all levels.</td>
</tr>
<tr>
<td>-write_tree</td>
<td></td>
<td>Write the poset of orbits as a tree file.</td>
</tr>
<tr>
<td>-find_node_by_stabilizer_order</td>
<td>i</td>
<td>Find all nodes whose stabilizer has order i.</td>
</tr>
<tr>
<td>-draw_poset</td>
<td></td>
<td>Produce a drawing of the poset of orbits.</td>
</tr>
<tr>
<td>-draw_full_poset</td>
<td></td>
<td>Produce a drawing of the full poset with elements grouped by orbits.</td>
</tr>
<tr>
<td>-plesken</td>
<td></td>
<td>Compute Plesken matrices Asup and Ainf.</td>
</tr>
<tr>
<td>-print_data_structure</td>
<td></td>
<td>Print the data structure.</td>
</tr>
<tr>
<td>-list</td>
<td></td>
<td>List orbits at level d.</td>
</tr>
</tbody>
</table>

Table 6.2: Options to control the poset classification algorithm (Part 1)
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-list_all</td>
<td></td>
<td>List orbits at all levels.</td>
</tr>
<tr>
<td>-table_of_nodes</td>
<td></td>
<td>Produce a spreadsheet of all orbits.</td>
</tr>
<tr>
<td>-make_relations_with_flag_orbits</td>
<td></td>
<td>Produce a bitmap drawing of the neighboring relations in the poset with flag orbits.</td>
</tr>
<tr>
<td>-Kramer_Mesner_matrix</td>
<td>$t ; k$</td>
<td>Compute the Kramer-Mesner matrix $M_{t,k}$.</td>
</tr>
<tr>
<td>-level_summary_csv</td>
<td></td>
<td>Write a summary of number of orbits at each level to a csv file.</td>
</tr>
<tr>
<td>-orbit_reps_csv</td>
<td></td>
<td>Write orbit representatives to a csv file.</td>
</tr>
<tr>
<td>-report..-end</td>
<td></td>
<td>Produce a latex report. Requires -orbiter_path option from Section 2.2</td>
</tr>
<tr>
<td>-node_label_is_group_order</td>
<td></td>
<td>When drawing the poset of orbits, display the group order in the orbit nodes.</td>
</tr>
<tr>
<td>-node_label_is_element</td>
<td></td>
<td>When drawing the poset of orbits, display the element rank in the orbit nodes.</td>
</tr>
<tr>
<td>-show_orbit_decomposition</td>
<td></td>
<td>Show the orbits of the stabilizers.</td>
</tr>
<tr>
<td>-show_stab</td>
<td></td>
<td>Show the stabilizers.</td>
</tr>
<tr>
<td>-save_stab</td>
<td></td>
<td>Save the stabilizer generators.</td>
</tr>
<tr>
<td>-show_whole_orbits</td>
<td></td>
<td>Show the whole orbits.</td>
</tr>
<tr>
<td>-recognize $L$</td>
<td></td>
<td>Recognize the given object in the classified list and compute a transporter that maps the given object to the canonical form. Here, $L$ must be a list of integers (comma separated and enclosed in double quotes) encoding an object. This option can be repeated.</td>
</tr>
<tr>
<td>-export_schreier_trees</td>
<td></td>
<td>Export all Schreier trees.</td>
</tr>
<tr>
<td>-draw_schreier_trees</td>
<td>args</td>
<td>Draw all Schreier trees.</td>
</tr>
<tr>
<td>-preferred_choice $n ; a ; b$</td>
<td></td>
<td>At node $n$, choose $b$ instead of $a$ as orbit representative.</td>
</tr>
</tbody>
</table>

Table 6.3: Options to control the poset classification algorithm (Part 2)
6.3 Orbits on Subsets

The lattice of subsets of a set $X$ is $\mathcal{P}(X)$, the set of all subsets of $X$, ordered with respect to inclusion. Assume that a group $G$ acts on $X$, and hence on the lattice by means of the induced action on subsets. The orbits of $G$ on subsets form a new poset, the poset of orbits. Poset classification is the process of computing the poset of orbits. Orbiter has an algorithm to perform poset classification. In many cases, we are not interested in the full lattice of subsets $\mathcal{P}(X)$ but rather in a subposet of it. We require that the subposet is closed under the group action and that the following property holds:

$$x, y \in \mathcal{P}(X) \text{ and } x \leq y \Rightarrow \left(y \in \mathcal{P} \rightarrow x \in \mathcal{P}\right).$$

The join of two subsets in the poset may or may not belong to the poset. Let us consider the action of the Singer cycle on $\text{PG}(3, 2)$. The following command computes the orbits of the group $G$ generated by it:

```
PGL_3_2_singer:
X $\langle\text{ORBITER\_PATH}/\text{orbiter\_out}\cdot-v\cdot3\cdot\rangle$
X $\langle\text{orbiter\_path}\cdot\langle\text{ORBITER\_PATH}\rangle\cdot\rangle$
X $\langle\text{define}\cdot G\cdot-\text{linear\_group}\cdot PGL\cdot3\cdot2\cdot-\text{singer}\cdot1\cdot-\text{end}\cdot\rangle$
X $\langle\text{with}\cdot G\cdot-\text{do}\cdot\rangle$
X $\langle\text{group\_theoretic\_activity}\cdot\rangle$
X $\langle\text{poset\_classification\_control}\cdot\rangle$
X $\langle\text{problem\_label}\cdot PGL\cdot3\cdot2\cdot_\text{singer}\cdot1\cdot-\text{W}\cdot-\text{depth}\cdot7\cdot\rangle$
X $\langle\text{draw\_poset}\cdot\rangle$
X $\langle\text{report}\cdot-\text{end}\cdot\rangle$
X $\langle\text{end}\cdot\rangle$
X $\langle\text{report}\cdot\rangle$
X $\langle\text{end}\cdot\rangle$
pdflatex PGL_3_2_singer_1_poset.tex
open PGL_3_2_singer_1_poset.pdf
```

Orbiter can compute orbits of groups acting in various different actions. For instance, the following example computes the orbits of $\text{PGL}(3, 2)$ acting on the lines of $\text{PG}(2, 2)$. All orbits on subsets of lines are classified:

```
PGL_3_2_on_lines:
X $\langle\text{ORBITER\_PATH}/\text{orbiter\_out}\cdot-v\cdot3\cdot\rangle$
X $\langle\text{orbiter\_path}\cdot\langle\text{ORBITER\_PATH}\rangle\cdot\rangle$
X $\langle\text{define}\cdot G\cdot-\text{linear\_group}\cdot PGL\cdot3\cdot2\cdot-\text{end}\cdot\rangle$
X $\langle\text{define}\cdot G\cdot_\text{on\_lines}\cdot-\text{modified\_group}\cdot-\text{from}\cdot G\cdot\rangle$
X $\langle\text{on\_k\_subspaces}\cdot2\cdot\rangle$
X $\langle\text{end}\cdot\rangle$
```

191
The following example computes the orbits of $\text{PGO}(5, 2)$ on the power set lattice of points of $Q(4, 2)$:

```
PGO_5_2_on_subsets:
  $(\text{ORBITER}\_\text{PATH})\text{oriter.out}\_v.3$
  $(\text{ORBITER}\_\text{PATH})\_\text{orbiter_path}$(\text{ORBITER}\_\text{PATH})$
  $(\text{ORBITER}\_\text{PATH})\_\text{define}F\_\text{finite_field}q.2\_\text{end}$
  $(\text{ORBITER}\_\text{PATH})\_\text{define}G\_\text{linear_group}PGO.5.F\_\text{end}$
  $(\text{ORBITER}\_\text{PATH})\_\text{with}G\_\text{do}$
  $(\text{ORBITER}\_\text{PATH})\_\text{group_theoretic_activity}$
  $(\text{ORBITER}\_\text{PATH})\_\text{poset_classification_control}$
  $(\text{ORBITER}\_\text{PATH})\_\text{problem_label}PGO.5.2$
  $(\text{ORBITER}\_\text{PATH})\_\text{depth}15$
  $(\text{ORBITER}\_\text{PATH})\_\text{draw_poset}$
  $(\text{ORBITER}\_\text{PATH})\_\text{w}$
  $(\text{ORBITER}\_\text{PATH})\_\text{end}$
  $(\text{ORBITER}\_\text{PATH})\_\text{orbits_on_subsets}15$
  $(\text{ORBITER}\_\text{PATH})\_\text{report}$
  $(\text{ORBITER}\_\text{PATH})\_\text{end}$
```

The poset of orbits is shown in Figure 6.4.
Figure 6.4: Orbits of PGO(5, 2) on the poset of subsets of $Q(4, 2)$
6.4 Orbits on Subspaces

Orbiter can compute the orbits of a group on the lattice of subspaces of a finite vector space.

The orthogonal group is the stabilizer of a non-degenerate quadric. Suppose we want to classify the subspaces in PG(3, 2) under the action of the orthogonal group. In PG(3, 2) there are two distinct nondegenerate quadrics, $Q^+(3, 2)$ and $Q^-(3, 2)$. The $Q^+(3, 2)$ quadric is a finite version of the quadric given by the equation

$$x_0x_1 + x_2x_3 = 0,$$

and depicted over the real numbers in Figure 6.5. PG(3, 2) has 15 points:

$$P_0 = (1, 0, 0, 0)$$  $$P_4 = (1, 1, 1, 1)$$  $$P_8 = (1, 1, 1, 0)$$  $$P_{12} = (0, 0, 1, 1)$$
$$P_1 = (0, 1, 0, 0)$$  $$P_5 = (1, 1, 0, 0)$$  $$P_9 = (1, 0, 0, 1)$$  $$P_{13} = (1, 0, 1, 1)$$
$$P_2 = (0, 0, 1, 0)$$  $$P_6 = (1, 0, 1, 0)$$  $$P_{10} = (0, 1, 0, 1)$$  $$P_{14} = (0, 1, 1, 1)$$
$$P_3 = (0, 0, 0, 1)$$  $$P_7 = (0, 1, 1, 0)$$  $$P_{11} = (1, 1, 0, 1)$$

The $Q^+(3, 2)$ quadric given by the equation above consists of the nine points

$$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8.$$

The quadric is stabilized by the group PGO$^+(4, 2)$ of order 72. The command

```
subspaces.Op.4.2:
  ▶ $(ORBITER_PATH)orbiter.out-\text{-}v.5\.\$
  ▶ ▶ $-\text{orbit}\text{-}\text{path}$$(ORBITER\_PATH)\$.\$
  ▶ ▶ ▶ $-\text{define}$-linear\_group-PGL.4.2-orthogonal.1.-end\$.\$
  ▶ ▶ ▶ ▶ $-\text{with}$-G.-do\$.\$
```
Figure 6.6: Hasse-diagram for the orbits of the orthogonal group $\text{PGO}^+(4, 2)$ on subspaces of $\text{PG}(3, 2)$

```latex
\begin{verbatim}
\text{-group_theoretic_activity.}\\
\text{-poset_classification_control.}\\
\text{\quad -node_label_is_element.}\\
\text{\quad -draw_poset:-draw_options:-radius:200:-end.}\\
\text{\quad -problem_label:Op_4.2:-W:-depth:4.}\\
\text{\quad -report:-end.}\\
\text{-end.}\\
\text{-orbits_on_subspaces:4.}\\
\text{-report.}\\
\text{-end.}\\
pdflatex\text{-PGL_4.2_Orthogonal_plus_4.2_poset.tex}\\
open\text{-PGL_4.2_Orthogonal_plus_4.2_poset.pdf}
\end{verbatim}
```

produces a classification of all subspaces of $\text{PG}(3, 2)$ under $\text{PGO}^+(4, 2)$. The option `-draw_poset` creates a Hasse diagram of the classification as shown in Figure 6.6. The nodes show the ranks of points in $\text{PG}(3, 2)$ as described in Section 4.1.
<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-q</td>
<td>q</td>
<td>Specify the size of the field $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-d</td>
<td>d</td>
<td>Require that no more than $d$ points lie on a line.</td>
</tr>
<tr>
<td>-n</td>
<td>n</td>
<td>The size of the matrix group.</td>
</tr>
<tr>
<td>-target_size</td>
<td>t</td>
<td>Specify the size of the arc to be $t$.</td>
</tr>
<tr>
<td>-conic_test</td>
<td></td>
<td>Require that no 6 points of the arc lie on a conic.</td>
</tr>
<tr>
<td>-affine</td>
<td></td>
<td>Classify arcs in the affine geometry, assuming that $x_0 = 0$ is the hyperplane at infinity. The condition that no more that $d$ point lie on a line applies to affine lines only.</td>
</tr>
<tr>
<td>-no_arc_testing</td>
<td></td>
<td>Do not test the at most $d$ points per line condition.</td>
</tr>
<tr>
<td>-forbidden_point_set</td>
<td>set</td>
<td>The arc must not contain any of the given points.</td>
</tr>
</tbody>
</table>

Table 6.4: Commands for Classifying Arcs

6.5 Arcs and Caps in Projective Spaces

In $\text{PG}(n, q)$, an arc is a set of points, no $n + 1$ in a hyperplane. A cap is set of points, no three collinear. Here, we restrict our attention to arcs in $\text{PG}(2, q)$. Arcs in higher dimensional projective spaces are equivalent to MDS codes and will be treated in Section 10. Our main examples will be the construction of the Lunelli-Sce hyperoval in $\text{PG}(2, 16)$ (cf. [47]) and the Pellegrino cap in $\text{AG}(4, 3)$. The uniqueness of this cap was proven by Hill [30].

A $(k, d)$-arc in a projective plane $\pi$ is a set $S$ of $k$ points such that very line intersects $S$ in at most $d$ points. Arcs are related to linear codes and other structures. Two arcs $S_1$ and $S_2$ are equivalent if there is a projectivity $\Phi$ such that $\Phi(A) = B$. The problem of classifying arcs is the problem of determining the orbits of the projectivity group on arcs. At times, we consider the larger group of collineations. In that case, the problem of classifying arcs is the problem of determining the orbits of the collineation group on arcs. Orbiter can solve such classification problems, at least for small parameter cases. Table 6.4 list the commands available to classify arcs. Here is an example. A hyperoval in a plane $\text{PG}(2, 2^e)$ is a $(2^e + 2, 2)$-arc. It is interesting to classify the hyperovals up to collineation equivalence under the group $\text{PGL}(3, 2^e)$. The command

```
hyperoval_16_classify:
  $ (\text{ORBITER\_PATH})$oriter.out.;v.4;\n  $ \text{-orbiter\_path};(\text{ORBITER\_PATH});\n  $ \text{-define};F.;\text{-finite\_field};-q.16;.-end;\n  $ \text{-define};P.;\text{-projective\_space};2:F.;.-end;\n```
performs the classification of hyperovals in \( \text{PG}(2,16) \). There are exactly two hyperovals in this plane. Orbiter also finds the stabilizers of these arcs. They have orders 16320 and 144, respectively. The two hyperovals are the regular hyperoval and the Lunelli-Sce hyperoval. Here is the relevant output from the Orbiter report (in the output, the Lunelli-Sce hyperoval is orbit 0, and the regular hyperoval is orbit 1):

### Orbits at Level 18

There are 2 orbits at level 18.

#### Orbit 0 / 2 at Level 18

Node number: 4212

\[
\{0, 1, 2, 3, 52, 67, 89, 106, 126, 141, 159, 176, 184, 199, 220, 235, 245, 262\}_{144}
\]

<table>
<thead>
<tr>
<th>Orbit</th>
<th>Stabilizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(1, 0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>(0, 1, 0)</td>
</tr>
<tr>
<td>2</td>
<td>(0, 0, 1)</td>
</tr>
<tr>
<td>3</td>
<td>(1, 1, 1)</td>
</tr>
<tr>
<td>4</td>
<td>(3, 2, 1)</td>
</tr>
<tr>
<td>5</td>
<td>(2, 3, 1)</td>
</tr>
<tr>
<td>6</td>
<td>(8, 4, 1)</td>
</tr>
<tr>
<td>7</td>
<td>(9, 5, 1)</td>
</tr>
<tr>
<td>8</td>
<td>(13, 6, 1)</td>
</tr>
<tr>
<td>9</td>
<td>(12, 7, 1)</td>
</tr>
<tr>
<td>10</td>
<td>(14, 8, 1)</td>
</tr>
<tr>
<td>11</td>
<td>(15, 9, 1)</td>
</tr>
<tr>
<td>12</td>
<td>(7, 10, 1)</td>
</tr>
<tr>
<td>13</td>
<td>(6, 11, 1)</td>
</tr>
<tr>
<td>14</td>
<td>(11, 12, 1)</td>
</tr>
<tr>
<td>15</td>
<td>(10, 13, 1)</td>
</tr>
<tr>
<td>16</td>
<td>(4, 14, 1)</td>
</tr>
<tr>
<td>17</td>
<td>(5, 15, 1)</td>
</tr>
</tbody>
</table>
Strong generators for a group of order 144:

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\delta^4 & \delta^9 & 1
\end{bmatrix},
\begin{bmatrix}
1 & \delta^7 & \delta^{13} \\
\delta^8 & \delta^9 & \delta^{10} \\
\delta & \delta & 1
\end{bmatrix},
\begin{bmatrix}
\delta^5 & \delta^5 & \delta^5 \\
\delta^5 & \delta^2 & \delta^{11} \\
\delta^5 & \delta^{14} & 1
\end{bmatrix}
\]

1,0,0,1,0,9,5,1,1,
1,7,6,14,5,10,2,15,1,3,
1,1,1,1,15,1,5,10,0,
There are 0 extensions
Number of generators 3

**Orbit 1 / 2 at Level 18**

Node number: 4213

\{0, 1, 2, 3, 52, 70, 83, 109, 127, 139, 156, 174, 186, 199, 217, 229, 256, 264\}_{16320}

0 : 0 = (1, 0, 0) 10 : 156 = (11, 8, 1)
1 : 1 = (0, 1, 0) 11 : 174 = (13, 9, 1)
2 : 2 = (0, 0, 1) 12 : 186 = (9, 10, 1)
3 : 3 = (1, 1, 1) 13 : 199 = (6, 11, 1)
4 : 52 = (3, 2, 1) 14 : 217 = (8, 12, 1)
5 : 70 = (5, 3, 1) 15 : 229 = (4, 13, 1)
6 : 83 = (2, 4, 1) 16 : 256 = (15, 14, 1)
7 : 109 = (12, 5, 1) 17 : 264 = (7, 15, 1)
8 : 127 = (14, 6, 1)
9 : 139 = (10, 7, 1)

Strong generators for a group of order 16320:

\[
\begin{bmatrix}
\delta^6 & 0 & 0 \\
0 & \delta^3 & 0 \\
0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
\delta^9 & 0 & 0 \\
0 & \delta^7 & 0 \\
0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
\delta^2 & 0 & 0 \\
0 & \delta^{11} & 0 \\
\delta^4 & \delta^7 & 1
\end{bmatrix},
\begin{bmatrix}
\delta^{10} & 0 & 0 \\
0 & \delta^3 & 0 \\
\delta & \delta^{11} & 1
\end{bmatrix},
\begin{bmatrix}
\delta & 0 & 0 \\
\delta^{12} & \delta^2 & \delta^5 \\
\delta^{14} & \delta^{10} & 1
\end{bmatrix},
\begin{bmatrix}
\delta^5 & 0 & 0 \\
\delta & \delta & \delta \\
\delta^6 & \delta^8 & 1
\end{bmatrix},
\begin{bmatrix}
\delta^{12} & 1 & \delta^2 \\
\delta^4 & \delta^3 & \delta^7 \\
\delta^6 & \delta^3 & 1
\end{bmatrix},
\begin{bmatrix}
\delta^{11} & \delta^6 & \delta^{10} \\
\delta^{10} & \delta^6 & 1
\end{bmatrix}
\]
1,0,0,0,3,0,0,0,5,2,
1,0,0,0,6,0,0,0,15,1,
1,0,0,0,5,0,4,11,6,3,
1,0,0,0,14,0,15,2,11,3,
1,0,13,2,9,6,5,12,1,
1,0,0,13,13,13,2,8,10,0,
1,8,11,7,15,10,5,15,8,2,
1,6,2,15,2,11,11,2,10,3,
There are 0 extensions
Number of generators 8

In the theory of cubic surfaces, we are interested in non-conical arcs. These are arcs which do not lie on a conic. The following example demonstrates how this can be done in Orbiter:

```
nc_arcs_16:
  $(ORBITER_PATH)oriter.out-v.4\n  -define:F:-finite_field-q.16:-end\n  -define:P:-projective_space-2:F:-end\n  -with:P-do\n  -projective_space_activity\n  -classify_arcs\n  -poset_classification_control\n  -problem_label:nc_arcs_q16_d2\n  -W:-depth:6\n  -report:-end\n  -end\n  -target_size:6\n  -d:2\n  -conic_test\n  -end\n  -end
```

Cubic surfaces are associated with arcs of size 6 (in a many-to-one relationship when considering isomorphism classes). The number of Eckardt points of the surface can be recovered from properties of the arc. For this reason, it is interesting to classify arcs so that the associated cubic surface has a fixed number of Eckardt points. The following command shows how to create all arcs associated with cubic surfaces with 13 Eckardt points over the field $\mathbb{F}_{32}$:

```
nc_arcs_32_E13:
```

199
\$$(\text{ORBITER\_PATH})\text{or}b\text{iter.\_out.-}v.4\backslash$
\$\backslash -\text{or}b\text{iter.\_path}\$(\text{ORBITER\_PATH)}\backslash$
\$\backslash -\text{define}\text{F.\_finite\_field}\text{-}q.32\text{-}end\backslash$
\$\backslash -\text{define}\text{P.\_projective\_space}\text{-}2\text{\_F.\_end}\backslash$
\$\backslash -\text{with}\text{P.\_do}\backslash$
\$\backslash -\text{projective\_space\_activity}\backslash$
\$\backslash \quad -\text{classify\_arcs}\backslash$
\$\backslash \quad -\text{-poset\_classification\_control}\backslash$
\$\backslash \quad \quad -\text{-problem\_label}\text{-}nc\text{-}arcs\text{-}q32\text{-}d2\backslash$
\$\backslash \quad \quad -\text{-W\_depth}\text{-}6\backslash$
\$\backslash \quad \quad -\text{-draw\_poset\_draw\_options\_end}\backslash$
\$\backslash \quad \quad -\text{-report\_end}\backslash$
\$\backslash \quad -\text{-end}\backslash$
\$\backslash \quad -\text{-target\_size}\text{-}6\backslash$
\$\backslash \quad -\text{-test\_nb\_Eckardt\_points}\text{-}13\backslash$
\$\backslash \quad -\text{-d.2}\backslash$
\$\backslash \quad -\text{-conic\_test}\backslash$
\$\backslash \quad -\text{-end}\backslash$
\$\backslash -\text{-end}\backslash$
\$pdflatex\text{-nc\_arcs\_q32\_d2\_poset.tex}\backslash$
\$open\text{-nc\_arcs\_q32\_d2\_poset.pdf}$
6.6 Cubic Curves

Orbiter can classify cubic curves in PG(2, q). To this end, the (9, 3)-arcs in PG(2, q) are classified first. From this classification, the classification of curves is computed. This classification only works for arcs which contain a (9, 3) arc. For very small fields, this is not always the case.

Here is an example. The command

cubic_curves_PG_2.8:
▷ $(ORBITER_PATH)orbiter.out-\text{-v}.3-\text{-define-G-}
▷ ▷ -\text{-define-F-}\text{-finite_field-}q.8-\text{-end-}
▷ ▷ -\text{-define-P-}\text{-projective_space-}F.\text{-end-}
▷ ▷ -\text{-with-P-}\text{-do-}
▷ ▷ -\text{-projective_space_activity-}
▷ ▷ -\text{-classify_cubic_curves-}q.8-\text{-target_size-}9-\text{-n-}3-\text{-d-}3-\text{-}
▷ ▷ ▷ -\text{-poset_classification_control-}
▷ ▷ ▷ ▷ -\text{-problem_label-}cc.8-\text{-W-}\text{-depth-}9-\text{-}
▷ ▷ ▷ ▷ -\text{-draw_options-}\text{-radius-}200-\text{-embedded-}\text{-end-}
▷ ▷ ▷ ▷ -\text{-recognize-"}0,1,2,3,35,28\text{-"-}
▷ ▷ ▷ ▷ -\text{-draw_poset-}
▷ ▷ ▷ ▷ -\text{-Kramer_Mesner_matrix-}6.9-\text{-}
▷ ▷ ▷ -\text{-end-}
▷ ▷ -\text{-end-}
▷ $(ORBITER_PATH)orbiter.out-\text{-v}.2-\text{-draw_matrix-}
▷ ▷ -\text{-input_csv_file-cc.8_KM_6.9.csv-}
▷ ▷ -\text{-box_width-}50-\text{-bit_depth-}8-\text{-end-}
▷ pdflatex\text{-Cubic_curves_q8.tex}
▷ open\text{-Cubic_curves_q8.pdf}
▷ #pdflatex\text{-cc.8_tree_lvl_9.tex}
▷ #open\text{-cc.8_tree_lvl_9.pdf}

classifies the cubic curves in PG(2, 8).
Chapter 7
Cubic Surfaces

7.1 Creation

Orbiter can create, classify and investigate cubic surfaces over small finite fields. In this
section, we describe ways in which surfaces can be created. The following sections will be
about classification and investigation.

Orbiter contains a built-in catalogue of cubic surfaces with 27 lines for small finite fields $\mathbb{F}_q$
(all surfaces in fields $\mathbb{F}_q$, $q \leq 97$, plus some for larger fields). The surfaces in the catalogue
all come with their automorphism group. It is also possible to create surfaces from known
families, or to create surfaces from associated objects like 6-arcs. Some of these constructions
only create the surface, not the automorphism group.

To create a cubic surface, one must first create a projective space object (three-dimensional). Tables 7.1-7.2 summarize the Orbiter commands that can be used to create cubic surfaces.
The commands are applied to the projective space object. Not all of the surfaces created may
have 27 lines, and some of the constructions may yield degenerate surfaces. Let us look at
some examples. The next command creates the unique surface with 27 lines over the field
$\mathbb{F}_4$, the Hirschfeld surface. The surface is pulled from Orbiter’s built-in catalogue of cubic
surfaces. The surface has Orbiter Catalogue Number (OCN) equal to 0.

```bash
surface_4_0:
  $(ORBITER_PATH)orbiter.out-v.3-
  define F -finite_field -q 4-end-
  define P -projective_space 3 F-end-
  with P-do-
  projective_space_activity-
  define_surface S -q 4 -catalogue 0 -end-
  end-
  with S-do-
  cubic_surface_activity-
  report-
  report_with_group-
  end
```

203
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-catalogue</td>
<td>$i$</td>
<td>Create the $i$-th surface in the Orbiter catalogue. Here, $i$ is an index variable used to index all surfaces in PG(3, $q$). The index $i$ is zero-based. The automorphism group is created as well.</td>
</tr>
<tr>
<td>-by_coefficients</td>
<td>list-of-coeff-pairs</td>
<td>Create a surface from a list of coefficient-monomial pairs. The automorphism group is not created.</td>
</tr>
<tr>
<td>-family_Eckardt</td>
<td>$a$ $b$</td>
<td>Create the Eckardt surface with parameters $(a,b)$ as in see [12] (where it is called the Hilbert, Cohn-Vossen surface). The equation is $X_3^3 - b^2(X_0^2 + X_1^2 + X_2^2)X_3 + \frac{b^2}{a^2 + 1}X_0X_1X_2 = 0$. The automorphism group is created as well.</td>
</tr>
<tr>
<td>-family_G13</td>
<td>$a$</td>
<td>Create a member of the $G_{13}$ family with parameter $a$. The surface has 13 Eckardt points.</td>
</tr>
<tr>
<td>-family_F13</td>
<td>$a$</td>
<td>Create a member of the $F_{13}$ family with parameter $a$. The surface has 13 Eckardt points.</td>
</tr>
<tr>
<td>-family_bes</td>
<td>$a$ $c$</td>
<td>Create a member of the bes family with parameter $a$. The surface has 5 Eckardt points. Bes means five in Turkish.</td>
</tr>
<tr>
<td>-family_general_abcd</td>
<td>$a$ $b$ $c$ $d$</td>
<td>Create a member of the general family with parameters $a,b,c,d$.</td>
</tr>
<tr>
<td>-arc_lifting</td>
<td>$A$</td>
<td>Create the surface associated with the arc $A = a_1, \ldots, a_6$ in PG(2, $q$) by means of the Clebsch map. Each of the $a_i$ is the rank of a point in PG(2, $q$). Use the trihedral pair algorithm. Here, $A$ is a comma-separated string containing the numerical ranks of the $P_i$ in PG(2, $q$).</td>
</tr>
</tbody>
</table>

Table 7.1: Commands to create a known cubic surface (Part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-arc_lifting_with_two_lines</td>
<td>$A \ L$</td>
<td>Create the surface associated with the arc $a_1, \ldots, a_6$ in PG($2, q$) by means of the Clebsch map. Each of the $a_i$ is the rank of a point in PG($2, q$). Use the two-lines algorithm. Here, $A$ is a comma-separated string containing the numerical ranks of the $P_i$ in PG($3, q$) and $L$ is a comma-separated string of the numerical ranks of two lines in PG($3, q$). If both of the lines are given as 0, the program will pick a suitable set of lines automatically.</td>
</tr>
<tr>
<td>-select_double_six</td>
<td>$L$</td>
<td>Relabel the lines by choosing the 12 lines in $L$ as new double six. The entries in $L$ are line indices with respect to the old double six. They are integers in the interval $[0, 26]$. This command can be repeated. In each application, the double six refers to the previous labeling.</td>
</tr>
<tr>
<td>-transform</td>
<td>$A$</td>
<td>Transform the surface by the projectivity (or collineation) defined by $A$. This option can be repeated.</td>
</tr>
<tr>
<td>-transform_inverse</td>
<td>$A$</td>
<td>Transform the surface by the inverse projectivity (or collineation) defined by $A$. This option can be repeated.</td>
</tr>
</tbody>
</table>

Table 7.2: Commands to create a known cubic surface (Part 2)
Two reports are created, one with and the other without information about the group.

Another way of creating surfaces is as members of known infinite families. For instance,

Eckardt_13:
\[
\text{\$ORBITER\_PATH/orbiter.out-v.3\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -define F:\text{-finite field}\_q13\_end\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -define P:\text{-projective space}\_3\_F\_end\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -with P:\text{-do} \$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -projective space activity\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -define S_q13\_q13\_family Eckardt\_3\_1\_end\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -end\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -with S_q13\_do\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -cubic surface activity\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -report\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -report_with_group\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ -end\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ pdflatex surface\_family Eckardt\_q13\_a3\_b1\_with group\$

\[\text{\$ORBITER\_PATH/orbiter.out-v.3\$ open surface\_family Eckardt\_q13\_a3\_b1\_with group.pdf}

creates the member of the Eckardt family described in [12] with parameters \((a,b) = (3,1)\) over the field \(F_{13}\).

Let us try the 4-parameter normal form of cubic surfaces with four parameters \(a, b, c, d\). The formula can be encoded as makefile variable:

\[
\begin{align*}
F_{abcd \text{ eqn}} = &-(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*(b-d)*X0*X0*X2 \\
+ &+(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*(a+b-c-d)*X0*X1*X2 \\
+ &+(a*a*c-a*a*d-a*c*c+b*c*c+a*d-b*c)*(b-d)*X0*X1*X3 \\
- &-(a*d-b*c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X0*X2*X2 \\
- &-(a*a*c*d-a*b*c*c-a*a*d+a*b*d+b*c*c-b*c*d)*(b-d)*X0*X2*X3 \\
- &-(a-c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X1*X1*X2 \\
- &-(a-c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X1*X1*X3 \\
+ &+(a*d-b*c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X1*X2*X2 \\
+ &+(1+1)*a*a*b*c*d-a*a*b*d^d-(1+1)*a*a*c*d^d \\
- &-(1+1)*a*b*b*c*c+a*b*b*c*d^d+(1+1)*a*b*c*c*d+a*b*c*d^d \\
- &-b*b*c*c^d-a*a*b*c+a*a*c^d+a*d+a*b*b*c+a*b*c^c \\
- &-(1+1+1)*a*b*b*c^c+a*c*c^d+a*c*d^d+b*b*c^c)*X1*X2*X3 \\
+ &+c*a*(a*d-b*c-a+a+b-c-d)*(b-d)*X1*X3*X3
\end{align*}
\]

\[206\]
The following command parses the formula and creates the surface with \((a,b,c,d) = (4,2,2,4)\):

```plaintext
F_abcd:
▷ $(\text{ORBITER\_PATH})\text{oriter.out\_v\_3}\$
▷ ▷ -define F\_finite_field_q\_7\_end\$
▷ ▷ -with F\_do\$
▷ ▷ -finite_field_activity\$
▷ ▷ -parse_and_evaluate \"F_{abcd}\" \"X0, X1, X2, X3\"\$
▷ ▷ ▷ $(F_{abcd\_eqn})\"a=4, b=2, c=2, d=4\"\$
▷ ▷ -end
```

It is possible to recreate the surface with the generators for the automorphism group. The following command creates two reports about the surface. One with and one without information about the group action.

```plaintext
F_alpha_beta_gamma_delta_q7_override_group:
▷ $(\text{ORBITER\_PATH})\text{oriter.out\_v\_3}\$
▷ ▷ -define F\_finite_field_q\_7\_end\$
▷ ▷ -define P\_projective_space_3\_F\_end\$
▷ ▷ -with P\_do\$
▷ ▷ -projective_space_activity\$
▷ ▷ -define_surface F_{2345\_q\_7}\$
▷ ▷ ▷ -by_equation \"F_{alpha\_beta\_gamma\_delta}\"\$
▷ ▷ ▷ \"D\{alpha, beta, gamma, delta\}D\" \"x0, x1, x2, x3\"\$
▷ ▷ ▷ $(F_{ALPHA\_BETA\_GAMMA\_DELTA})\$
▷ ▷ ▷ \"alpha=2, beta=3, gamma=4, delta=5\"\$
▷ ▷ ▷ \"D\{alpha=2, beta=3, gamma=4, delta=5\}\"\$
▷ ▷ ▷ -override_group 6\_2\$
▷ ▷ ▷ \"1,5,0,0,3,6,0,0,1,1,3,0,5,5,0,3,1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4\"\$
▷ ▷ -end\$
▷ ▷ -end\$
▷ ▷ -cubic_surface_activity\$
▷ ▷ -report\$
▷ ▷ ▷ -report_with_group\$
▷ ▷ -end
▷ pdflatex\_surface\_equation_F_alpha_beta_gamma_delta_q7\_report.tex
▷ open\_surface\_equation_F_alpha_beta_gamma_delta_q7\_report.pdf
▷ pdflatex\_surface\_equation_F_alpha_beta_gamma_delta_q7\_with\_group.tex
▷ open\_surface\_equation_F_alpha_beta_gamma_delta_q7\_with\_group.pdf
```
7.2 Quartic Curves

Cubic surfaces with 27 lines are associated with quartic curves with 28 bitangents (see [31]), which in turn are associated with del Pezzo surfaces. Orbiter can classify quartic curves based on a known classification of cubic surfaces. Orbiter also has a catalogue of quartic curves for small field sizes. Here is an example. Suppose we want to study the (unique) quartic curve for $q = 9$. The following command pulls the curve from the catalogue and produces a report:

```
quartic_curve_9_0_report::
▷ $(ORBITER_PATH) orbiter.out -v 3 \n▷ ▷ -define: F: finite_field: q 9 -end \n▷ ▷ -define: P: projective_space: 2 F: end \n▷ ▷ -with: P: do \n▷ ▷ ▷ -projective_space_activity \n▷ ▷ ▷ -define_quartic_curve: C: q 9 \n▷ ▷ ▷ -catalogue: 0 -end \n▷ ▷ -end \n▷ -with: C: do \n▷ ▷ -quartic_curve_activity \n▷ ▷ ▷ -report \n▷ ▷ -end \n▷ pdflatex quartic_curve_catalogue_q9_is0_report.tex
▷ open quartic_curve_catalogue_q9_is0_report.pdf
```

The report contains the following information:

**The equation**

The equation of the quartic curve is:

$$\alpha^3 X_0^3 X_1 + \alpha^4 X_0^3 X_2 + \alpha^7 X_0 X_1^3 + \alpha^6 X_1^3 X_2 + \alpha^2 X_2^3 + X_1 X_2^3$$

$$(0, 0, 0, 8, 2, 4, 5, 7, 1, 0, 0, 0, 0, 0, 0)$$

**The gradient**

The gradient of the quartic curve is:

$$\alpha^7 X_1^3 + \alpha^2 X_2^3$$

$$(0, 4, 7, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$\alpha^3 X_0^3 + X_2^3$$
\[ \alpha^4 X_0^3 + \alpha^6 X_1^3 \]
\[ (2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0) \]

**General information**

<table>
<thead>
<tr>
<th>Number of bitangents</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of points</td>
<td>28</td>
</tr>
<tr>
<td>Fullness</td>
<td>is full</td>
</tr>
<tr>
<td>Number of Kovalevski points</td>
<td>63</td>
</tr>
<tr>
<td>Bitangent line type ((a_0, a_1, a_2))</td>
<td>((0, 28, 0))</td>
</tr>
<tr>
<td>Number of singular points</td>
<td>0</td>
</tr>
</tbody>
</table>

**All points on the curve**

The surface has 28 points:

The points on the quartic curve are:

0 : \( P_0 = (1, 0, 0) \)
1 : \( P_1 = (0, 1, 0) \)
2 : \( P_2 = (0, 0, 1) \)
3 : \( P_3 = (1, 1, 1) \)
4 : \( P_4 = (1, 1, 0) \)
5 : \( P_5 = (2, 1, 0) \)
6 : \( P_{14} = (3, 0, 1) \)
7 : \( P_{17} = (6, 0, 1) \)
8 : \( P_{24} = (5, 1, 1) \)
9 : \( P_{25} = (6, 1, 1) \)
10 : \( P_{30} = (2, 2, 1) \)
11 : \( P_{32} = (4, 2, 1) \)
12 : \( P_{34} = (6, 2, 1) \)
13 : \( P_{38} = (1, 3, 1) \)
14 : \( P_{41} = (4, 3, 1) \)
15 : \( P_{44} = (7, 3, 1) \)
16 : \( P_{46} = (0, 4, 1) \)
17 : \( P_{51} = (5, 4, 1) \)
18 : \( P_{53} = (7, 4, 1) \)
19 : \( P_{57} = (2, 5, 1) \)
20 : \( P_{58} = (3, 5, 1) \)
21 : \( P_{62} = (7, 5, 1) \)
22 : \( P_{76} = (3, 7, 1) \)
23 : \( P_{77} = (4, 7, 1) \)
24 : \( P_{78} = (5, 7, 1) \)
25 : \( P_{80} = (0, 8, 1) \)
26 : \( P_{83} = (1, 8, 1) \)
27 : \( P_{84} = (2, 8, 1) \)

The points by rank are: \((0, 1, 2, 3, 4, 5, 14, 17, 24, 25, 30, 32, 34, 38, 41, 44, 46, 51, 53, 57, 58, 62, 76, 77, 78, 82, 83, 84)\)

The Kovalevski points are:

0 : \( P_7 = (4, 1, 0) = c_{13} \cap c_{14} \cap c_{36} \cap c_{46} \)
1 : \( P_8 = (5, 1, 0) = a_2 \cap a_4 \cap c_{25} \cap c_{45} \)
2 : \( P_9 = (6, 1, 0) = b_1 \cap b_6 \cap c_{12} \cap c_{26} \)
3 : \( P_{10} = (7, 1, 0) = a_3 \cap b_5 \cap c_{35} \cap d \)
4 : \( P_{11} = (8, 1, 0) = b_2 \cap b_3 \cap c_{24} \cap c_{34} \)
5 : \( P_{12} = (1, 0, 1) = a_3 \cap a_4 \cap c_{23} \cap c_{24} \)
6 : \( P_{13} = (2, 0, 1) = c_{34} \cap c_{36} \cap c_{45} \cap c_{56} \)
7 : \( P_{15} = (4, 0, 1) = b_3 \cap b_6 \cap c_{13} \cap c_{16} \)
8 : \( P_{16} = (5, 0, 1) = a_5 \cap a_6 \cap c_{25} \cap c_{26} \)
9 : \( P_{18} = (7, 0, 1) = a_2 \cap b_1 \cap c_{35} \cap c_{46} \)
10 : \( P_{19} = (8, 0, 1) = b_4 \cap b_5 \cap c_{14} \cap c_{15} \)
11 : \( P_{20} = (0, 1, 1) = a_2 \cap b_3 \cap c_{14} \cap c_{56} \)
12 : \( P_{21} = (2, 1, 1) = b_2 \cap b_4 \cap c_{26} \cap c_{46} \)
13 : \( P_{22} = (3, 1, 1) = a_4 \cap b_5 \cap c_{12} \cap c_{36} \)
14 : \( P_{23} = (4, 1, 1) = a_6 \cap b_1 \cap c_{23} \cap c_{45} \)
15 : \( P_{26} = (7, 1, 1) = c_{16} \cap c_{25} \cap c_{34} \cap d \)
16 : \( P_{27} = (8, 1, 1) = a_3 \cap a_5 \cap c_{13} \cap c_{15} \)
17 : \( P_{28} = (0, 2, 1) = c_{12} \cap c_{13} \cap c_{25} \cap c_{35} \)
18 : \( P_{29} = (1, 2, 1) = b_1 \cap b_5 \cap c_{16} \cap c_{56} \)
19 : \( P_{31} = (3, 2, 1) = a_3 \cap a_6 \cap c_{34} \cap c_{46} \)
20 : \( P_{33} = (5, 2, 1) = a_2 \cap b_4 \cap c_{24} \cap d \)
21 : \( P_{35} = (7, 2, 1) = b_2 \cap b_6 \cap c_{23} \cap c_{36} \)
22 : \( P_{36} = (8, 2, 1) = a_4 \cap b_3 \cap c_{15} \cap c_{26} \)
23 : \( P_{37} = (0, 3, 1) = a_5 \cap b_1 \cap c_{24} \cap c_{36} \)
24 : \( P_{39} = (2, 3, 1) = a_2 \cap a_6 \cap c_{12} \cap c_{16} \)
25 : \( P_{40} = (3, 3, 1) = b_3 \cap b_4 \cap c_{35} \cap c_{45} \)
26 : \( P_{42} = (5, 3, 1) = a_4 \cap b_2 \cap c_{13} \cap c_{56} \)
27 : \( P_{43} = (6, 3, 1) = a_3 \cap b_6 \cap c_{14} \cap c_{25} \)
28 : \( P_{45} = (8, 3, 1) = c_{15} \cap c_{23} \cap c_{46} \cap d \)
29 : \( P_{47} = (1, 4, 1) = a_6 \cap b_2 \cap c_{14} \cap c_{35} \)
30 : \( P_{48} = (2, 4, 1) = b_3 \cap b_5 \cap c_{23} \cap c_{25} \)
31 : \( P_{49} = (3, 4, 1) = a_5 \cap b_6 \cap c_{56} \cap d \)
32 : \( P_{50} = (4, 4, 1) = a_2 \cap a_3 \cap c_{26} \cap c_{36} \)
33 : \( P_{52} = (6, 4, 1) = b_1 \cap b_4 \cap c_{13} \cap c_{34} \)
34 : \( P_{54} = (8, 4, 1) = c_{12} \cap c_{15} \cap c_{24} \cap c_{45} \)
35 : \( P_{55} = (0, 5, 1) = a_4 \cap a_6 \cap b_4 \cap b_6 \)
36 : \( P_{56} = (1, 5, 1) = c_{13} \cap c_{26} \cap c_{45} \cap d \)
37 : \( P_{59} = (4, 5, 1) = c_{24} \cap c_{25} \cap c_{46} \cap c_{56} \)
38 : \( P_{60} = (5, 5, 1) = c_{12} \cap c_{14} \cap c_{23} \cap c_{34} \)
39 : \( P_{61} = (6, 5, 1) = a_2 \cap a_5 \cap b_2 \cap b_5 \)
40 : \( P_{63} = (8, 5, 1) = c_{15} \cap c_{16} \cap c_{35} \cap c_{36} \)
41 : \( P_{64} = (0, 6, 1) = a_1 \cap b_5 \cap c_{26} \cap c_{34} \)
42 : \( P_{65} = (1, 6, 1) = a_1 \cap b_4 \cap c_{25} \cap c_{36} \)
43 : \( P_{66} = (2, 6, 1) = a_1 \cap b_6 \cap c_{24} \cap c_{35} \)
44 : \( P_{67} = (3, 6, 1) = a_1 \cap a_2 \cap c_{13} \cap c_{23} \)
45 : \( P_{68} = (4, 6, 1) = a_1 \cap b_2 \cap c_{12} \cap d \)
46 : \( P_{69} = (5, 6, 1) = a_1 \cap a_3 \cap b_1 \cap b_3 \)
47 : \( P_{70} = (6, 6, 1) = a_1 \cap a_4 \cap c_{16} \cap c_{46} \)
48 : \( P_{71} = (7, 6, 1) = a_1 \cap a_5 \cap c_{14} \cap c_{45} \)
49 : \( P_{72} = (8, 6, 1) = a_1 \cap a_6 \cap c_{15} \cap c_{56} \)
The Kovalevski points by rank are: (7, 8, 9, 10, 11, 12, 13, 15, 16, 18, 19, 20, 21, 22, 23, 26, 27, 28, 29, 31, 33, 35, 36, 37, 39, 40, 42, 43, 45, 47, 48, 49, 50, 52, 54, 55, 56, 59, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 79, 80, 81, 85, 86, 87, 88, 89, 90, 6)

The points off the curve are:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$P_6 = (3, 1, 0)$</td>
</tr>
<tr>
<td>1</td>
<td>$P_7 = (4, 1, 0)$</td>
</tr>
<tr>
<td>2</td>
<td>$P_8 = (5, 1, 0)$</td>
</tr>
<tr>
<td>3</td>
<td>$P_9 = (6, 1, 0)$</td>
</tr>
<tr>
<td>4</td>
<td>$P_{10} = (7, 1, 0)$</td>
</tr>
<tr>
<td>5</td>
<td>$P_{11} = (8, 1, 0)$</td>
</tr>
<tr>
<td>6</td>
<td>$P_{12} = (1, 0, 1)$</td>
</tr>
<tr>
<td>7</td>
<td>$P_{13} = (2, 0, 1)$</td>
</tr>
<tr>
<td>8</td>
<td>$P_{15} = (4, 0, 1)$</td>
</tr>
<tr>
<td>9</td>
<td>$P_{16} = (5, 0, 1)$</td>
</tr>
<tr>
<td>10</td>
<td>$P_{18} = (7, 0, 1)$</td>
</tr>
<tr>
<td>11</td>
<td>$P_{19} = (8, 0, 1)$</td>
</tr>
<tr>
<td>12</td>
<td>$P_{20} = (0, 1, 1)$</td>
</tr>
<tr>
<td>13</td>
<td>$P_{21} = (2, 1, 1)$</td>
</tr>
<tr>
<td>14</td>
<td>$P_{22} = (3, 1, 1)$</td>
</tr>
<tr>
<td>15</td>
<td>$P_{23} = (4, 1, 1)$</td>
</tr>
<tr>
<td>16</td>
<td>$P_{26} = (7, 1, 1)$</td>
</tr>
<tr>
<td>17</td>
<td>$P_{27} = (8, 1, 1)$</td>
</tr>
<tr>
<td>18</td>
<td>$P_{28} = (0, 2, 1)$</td>
</tr>
<tr>
<td>19</td>
<td>$P_{29} = (1, 2, 1)$</td>
</tr>
<tr>
<td>20</td>
<td>$P_{31} = (3, 2, 1)$</td>
</tr>
<tr>
<td>21</td>
<td>$P_{33} = (5, 2, 1)$</td>
</tr>
</tbody>
</table>

The points at rank 1 are: (0, 7, 1) = $a_3 \cap b_2 \cap c_{16} \cap c_{45}$

The points at rank 2 are: (1, 7, 1) = $a_5 \cap b_3 \cap c_{12} \cap c_{46}$

The points at rank 3 are: (2, 7, 1) = $a_4 \cap b_1 \cap c_{14} \cap d$

The points at rank 4 are: (6, 7, 1) = $c_{23} \cap c_{26} \cap c_{35} \cap c_{56}$

The points at rank 5 are: (7, 7, 1) = $a_6 \cap b_5 \cap c_{13} \cap c_{24}$

The points at rank 6 are: (8, 7, 1) = $a_2 \cap b_6 \cap c_{15} \cap c_{34}$

The points at rank 7 are: (3, 8, 1) = $c_{14} \cap c_{16} \cap c_{24} \cap c_{26}$

The points at rank 8 are: (4, 8, 1) = $a_4 \cap a_5 \cap c_{34} \cap c_{35}$

The points at rank 9 are: (5, 8, 1) = $b_5 \cap b_6 \cap c_{45} \cap c_{46}$

The points at rank 10 are: (6, 8, 1) = $a_6 \cap b_3 \cap c_{36} \cap d$

The points at rank 11 are: (7, 8, 1) = $a_3 \cap b_4 \cap c_{12} \cap c_{56}$

The points at rank 12 are: (8, 8, 1) = $b_1 \cap b_2 \cap c_{15} \cap c_{25}$
The lines and their points of contact are:

\[
\begin{align*}
a_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_0 = P(1, 0, 0) \times 4 \\
a_2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_5 = P(1, 5, 0) \times 4 \\
a_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_8 = P(1, 8, 0) \times 4 \\
a_4 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
b_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{62} = P(6, 2, 0) \times 4 \\
b_2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{77} = P(7, 7, 0) \times 4 \\
b_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{41} = P(4, 1, 0) \times 4 \\
b_4 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
b_5 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
b_6 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
c_{12} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
c_{13} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
c_{14} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{32} = P(3, 2, 0) \times 4 \\
\end{align*}
\]
\[
c_{15} = \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & 0 \end{bmatrix}, \quad c_{16} = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^6 \end{bmatrix}, \quad c_{23} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^5 \end{bmatrix}, \quad c_{24} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^7 \end{bmatrix}, \quad c_{25} = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^4 \end{bmatrix}, \quad c_{26} = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^3 \end{bmatrix},
\]
\[
c_{34} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^3 \end{bmatrix}, \quad c_{35} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^4 \end{bmatrix}, \quad c_{36} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^7 \end{bmatrix}, \quad c_{45} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^6 \end{bmatrix}, \quad c_{46} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha \end{bmatrix}, \quad c_{56} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & 1 \end{bmatrix}, 
\]
\[
d = \begin{bmatrix} 1 & \alpha^6 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_1 = P(0, 1, 0) 4\times
\]
\[
c_{16} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^7 \end{bmatrix}, \quad c_{23} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^5 \end{bmatrix}, \quad c_{24} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^7 \end{bmatrix}, \quad c_{25} = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^4 \end{bmatrix}, \quad c_{26} = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^3 \end{bmatrix},
\]
\[
c_{34} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^3 \end{bmatrix}, \quad c_{35} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^4 \end{bmatrix}, \quad c_{36} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^7 \end{bmatrix}, \quad c_{45} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^6 \end{bmatrix}, \quad c_{46} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha \end{bmatrix}, \quad c_{56} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & 1 \end{bmatrix}, 
\]
\[
d = \begin{bmatrix} 1 & \alpha^6 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_2 = P(0, 0, 1) 4\times
\]

Rank of lines: (8, 51, 15, 17, 74, 77, 54, 45, 31, 67, 68, 37, 82, 32, 61, 60, 35, 16, 14, 72, 78, 28, 52, 24, 25, 53, 21, 59)

Line type: 1^{28}

point types: 1^{28}

<table>
<thead>
<tr>
<th></th>
<th>1, 2, 3, 4, 5, 6, 7, 8, 9, 10,</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>11, 12, 13, 14, 15, 16, 17, 18, 19, 20,</td>
</tr>
<tr>
<td></td>
<td>21, 22, 23, 24, 25, 26, 27, 0</td>
</tr>
</tbody>
</table>

213
<table>
<thead>
<tr>
<th>28</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 0</td>
<td></td>
</tr>
</tbody>
</table>

point types for points off the curve: $4^{63}$

<table>
<thead>
<tr>
<th>63</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 0</td>
<td></td>
</tr>
</tbody>
</table>

Lines on points off the curve:
Off point 0 = $P_6 = (3, 1, 0)$ lies on 4 bisecants: \{ 4, 9, 16, 17 \}
Off point 1 = $P_7 = (4, 1, 0)$ lies on 4 bisecants: \{ 13, 14, 23, 25 \}
Off point 2 = $P_8 = (5, 1, 0)$ lies on 4 bisecants: \{ 1, 3, 19, 24 \}
Off point 3 = $P_9 = (6, 1, 0)$ lies on 4 bisecants: \{ 6, 11, 12, 20 \}
Off point 4 = $P_{10} = (7, 1, 0)$ lies on 4 bisecants: \{ 2, 10, 22, 27 \}
Off point 5 = $P_{11} = (8, 1, 0)$ lies on 4 bisecants: \{ 7, 8, 18, 21 \}
Off point 6 = $P_{12} = (1, 0, 1)$ lies on 4 bisecants: \{ 2, 3, 17, 18 \}
Off point 7 = $P_{13} = (2, 0, 1)$ lies on 4 bisecants: \{ 21, 23, 24, 26 \}
Off point 8 = $P_{14} = (4, 0, 1)$ lies on 4 bisecants: \{ 8, 11, 13, 16 \}
Off point 9 = $P_{15} = (5, 0, 1)$ lies on 4 bisecants: \{ 4, 5, 19, 20 \}
Off point 10 = $P_{16} = (7, 0, 1)$ lies on 4 bisecants: \{ 1, 6, 22, 25 \}
Off point 11 = $P_{17} = (8, 0, 1)$ lies on 4 bisecants: \{ 9, 10, 14, 15 \}
Off point 12 = $P_{18} = (0, 1, 1)$ lies on 4 bisecants: \{ 1, 8, 14, 26 \}
Off point 13 = $P_{19} = (2, 1, 1)$ lies on 4 bisecants: \{ 7, 9, 20, 25 \}
Off point 14 = $P_{20} = (3, 1, 1)$ lies on 4 bisecants: \{ 3, 10, 12, 23 \}
Off point 15 = $P_{21} = (4, 1, 1)$ lies on 4 bisecants: \{ 5, 6, 17, 24 \}
Off point 16 = $P_{22} = (7, 1, 1)$ lies on 4 bisecants: \{ 16, 19, 21, 27 \}
Off point 17 = $P_{23} = (8, 1, 1)$ lies on 4 bisecants: \{ 2, 4, 13, 15 \}
Off point 18 = $P_{24} = (0, 2, 1)$ lies on 4 bisecants: \{ 12, 13, 19, 22 \}
Off point 19 = $P_{25} = (1, 2, 1)$ lies on 4 bisecants: \{ 6, 10, 16, 26 \}
Off point 20 = $P_{26} = (3, 2, 1)$ lies on 4 bisecants: \{ 2, 5, 21, 25 \}
Off point 21 = $P_{27} = (5, 2, 1)$ lies on 4 bisecants: \{ 1, 9, 18, 27 \}
Off point 22 = $P_{28} = (7, 2, 1)$ lies on 4 bisecants: \{ 7, 11, 17, 23 \}
Off point 23 = $P_{29} = (8, 2, 1)$ lies on 4 bisecants: \{ 3, 8, 15, 20 \}
Off point 24 = $P_{30} = (0, 3, 1)$ lies on 4 bisecants: \{ 4, 6, 18, 23 \}
Off point 25 = $P_{31} = (2, 3, 1)$ lies on 4 bisecants: \{ 1, 5, 12, 16 \}
Off point 26 = $P_{32} = (3, 3, 1)$ lies on 4 bisecants: \{ 8, 9, 22, 24 \}
| Off point 28 = $P_{32} = (6, 3, 1)$ lies on 4 bisecants: | {3, 7, 13, 26} |
| Off point 29 = $P_{33} = (8, 3, 1)$ lies on 4 bisecants: | {2, 11, 14, 19} |
| Off point 30 = $P_{34} = (1, 4, 1)$ lies on 4 bisecants: | {5, 7, 14, 22} |
| Off point 31 = $P_{35} = (2, 4, 1)$ lies on 4 bisecants: | {8, 10, 17, 19} |
| Off point 32 = $P_{36} = (3, 4, 1)$ lies on 4 bisecants: | {4, 11, 26, 27} |
| Off point 33 = $P_{37} = (4, 4, 1)$ lies on 4 bisecants: | {1, 2, 20, 23} |
| Off point 34 = $P_{38} = (6, 4, 1)$ lies on 4 bisecants: | {6, 9, 13, 21} |
| Off point 35 = $P_{39} = (8, 4, 1)$ lies on 4 bisecants: | {12, 15, 18, 24} |
| Off point 36 = $P_{40} = (0, 5, 1)$ lies on 4 bisecants: | {3, 5, 9, 11} |
| Off point 37 = $P_{41} = (1, 5, 1)$ lies on 4 bisecants: | {13, 20, 24, 27} |
| Off point 38 = $P_{42} = (4, 5, 1)$ lies on 4 bisecants: | {18, 19, 25, 26} |
| Off point 39 = $P_{43} = (5, 5, 1)$ lies on 4 bisecants: | {12, 14, 17, 21} |
| Off point 40 = $P_{44} = (6, 5, 1)$ lies on 4 bisecants: | {1, 4, 7, 10} |
| Off point 41 = $P_{45} = (8, 5, 1)$ lies on 4 bisecants: | {15, 16, 22, 23} |
| Off point 42 = $P_{46} = (0, 6, 1)$ lies on 4 bisecants: | {0, 10, 20, 21} |
| Off point 43 = $P_{47} = (1, 6, 1)$ lies on 4 bisecants: | {0, 9, 19, 23} |
| Off point 44 = $P_{48} = (2, 6, 1)$ lies on 4 bisecants: | {0, 11, 18, 22} |
| Off point 45 = $P_{49} = (3, 6, 1)$ lies on 4 bisecants: | {0, 1, 13, 17} |
| Off point 46 = $P_{50} = (4, 6, 1)$ lies on 4 bisecants: | {0, 7, 12, 27} |
| Off point 47 = $P_{51} = (5, 6, 1)$ lies on 4 bisecants: | {0, 2, 6, 8} |
| Off point 48 = $P_{52} = (6, 6, 1)$ lies on 4 bisecants: | {0, 3, 16, 25} |
| Off point 49 = $P_{53} = (7, 6, 1)$ lies on 4 bisecants: | {0, 4, 14, 24} |
| Off point 50 = $P_{54} = (8, 6, 1)$ lies on 4 bisecants: | {0, 5, 15, 26} |
| Off point 51 = $P_{55} = (0, 7, 1)$ lies on 4 bisecants: | {2, 7, 16, 24} |
| Off point 52 = $P_{56} = (1, 7, 1)$ lies on 4 bisecants: | {4, 8, 12, 25} |
| Off point 53 = $P_{57} = (2, 7, 1)$ lies on 4 bisecants: | {3, 6, 14, 27} |
| Off point 54 = $P_{58} = (3, 7, 1)$ lies on 4 bisecants: | {17, 20, 22, 26} |
| Off point 55 = $P_{59} = (4, 7, 1)$ lies on 4 bisecants: | {5, 10, 13, 18} |
| Off point 56 = $P_{60} = (5, 7, 1)$ lies on 4 bisecants: | {1, 11, 15, 21} |
| Off point 57 = $P_{61} = (6, 7, 1)$ lies on 4 bisecants: | {14, 16, 18, 20} |
| Off point 58 = $P_{62} = (7, 7, 1)$ lies on 4 bisecants: | {3, 4, 21, 22} |
| Off point 59 = $P_{63} = (8, 7, 1)$ lies on 4 bisecants: | {10, 11, 24, 25} |
| Off point 60 = $P_{64} = (5, 8, 1)$ lies on 4 bisecants: | {5, 8, 23, 27} |
| Off point 61 = $P_{65} = (6, 8, 1)$ lies on 4 bisecants: | {2, 9, 12, 26} |
| Off point 62 = $P_{66} = (8, 8, 1)$ lies on 4 bisecants: | {6, 7, 15, 19} |
7.3 Classification

There are several different approaches to classify cubic surfaces with 27 lines over finite fields $\mathbb{F}_q$ in Orbiter. Classification means to determine the non-equivalent surfaces under the action of the collineation group $\text{PGL}(4, q)$ of $\text{PG}(3, q)$. The approach described in [12] relies on Schlaefli’s notion of a double six as a substructure [57]. The approach described in [36] utilizes the relation to non-conical six-arcs in a plane. A third approach is described in [37]. All three approaches are available in Orbiter.

In $\text{PG}(3, 4)$, there is only one type of cubic surfaces with 27 lines. It is a member of the Hirschfeld family, described in [32]. The following Orbiter command can be used to construct this surface and to prove its uniqueness for $\mathbb{F}_4$. The following command utilizes the algorithm of [12] to do so:

```
surface classify_q4:
  > $(ORBITER_PATH)orbiter.out-v-5\n  >   -defineF:-finite_field-q4-end\n  >   -defineP:-projective_space-3F:-end\n  >   -withP:-do\n  >   -projective_space_activity\n  >   > -classify_surfaces_with_double_sixes-Surf27-W:-end\n  >   > -end\n  >   > -withSurf27:-do\n  >   > -classification_of_cubic_surfaces_with_double_sixes_activity\n  >   > > -report:-end\n  >   > > -end\n  >   > -print_symbols
  > pdflatex Surfaces_q4.tex
  > open Surfaces_q4.pdf
```

The -report option creates a latex report. After some redactions, the report contains the following elements.

---

**The semilinear group**

**The Action**

Group action $\text{PGL}(4, 4)$ of degree 85

The group is a matrix group.

The base action is on projective space $\text{PG}(3, 4)$

$q = 4$
The orthogonal group

The Action

Group action $\text{PGL}(4, 4)\text{OnWedge}$ of degree 1365
The group is a matrix group.
The base action is on projective space $\text{PG}(3, 4)$

The group stabilizing the fixed line

The Action

Group action $\text{PGL}(4, 4)\text{OnWedges100}$ of degree 100

Strong generators for a group of order 5529600: :

The classification of five-plus-ones

Poset classification up to depth 5
The Orbits

Number of Orbits By Level

<table>
<thead>
<tr>
<th>Depth</th>
<th>Nb of orbits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Summary of Orbit Representatives

N = node
D = depth or level
O = orbit with a level
Rep = orbit representative
(S,O) = (order of stabilizer, orbit length)
L = number of live points
F = number of flags
Gen = number of generators for the stabilizer of the orbit rep.

Table 7.3: Orbit Representatives

<table>
<thead>
<tr>
<th>N</th>
<th>D</th>
<th>O</th>
<th>Rep</th>
<th>(S,O)</th>
<th>L</th>
<th>F</th>
<th>Gen</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>{}</td>
<td>(5529600, 1)</td>
<td>100</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>{ 0 }</td>
<td>(55296, 100)</td>
<td>64</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>{ 0, 3 }</td>
<td>(1728, 3200)</td>
<td>36</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>{ 0, 3, 56 }</td>
<td>(144, 38400)</td>
<td>16</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>{ 0, 3, 56, 76 }</td>
<td>(288, 19200)</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>{ 0, 3, 56, 77 }</td>
<td>(96, 57600)</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
<td>{ 0, 3, 56, 80 }</td>
<td>(72, 76800)</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>0</td>
<td>{ 0, 3, 56, 76, 96 }</td>
<td>(1440, 3840)</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1</td>
<td>{ 0, 3, 56, 76, 97 }</td>
<td>(96, 57600)</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>2</td>
<td>{ 0, 3, 56, 80, 92 }</td>
<td>(360, 15360)</td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Poset of Orbits in Detail

Classification of 5 + 1 Configurations in PG(3, 4)

The order of the group is 1974067200
The group has 4 orbits on five plus one configurations in PG(3, 4).

Of these, 1 impose 19 conditions.
Of these, 1 are associated with double sixes. They are:
0/1 is orbit 3/4 \{0, 3, 56, 80, 93\}_{120} orbit length 46080
The overall number of five plus one configurations associated with double sixes in PG(3, 4) is: 46080

Flag orbits for double sixes

The number of primary orbits below is 4
The number of primary orbits above is 1
The number of flag orbits is 1
The flag orbits are:

(1) Flag orbit 0 / 1 down=(3,0) up=(0,-1) is ( 0, 3, 56, 80, 93, 16, 340, 38, 61, 156, 0, 16, 340, 38, 61, 156, 165, 72, 54, 25, 356, 0 ) with a stabilizer of order 120

Strong generators for a group of order 120:
$$\begin{bmatrix}
1 & \omega & 0 & 0 \\
0 & 1 & 0 & 0 \\
\omega & \omega & 1 & \omega^2 \\
0 & \omega^2 & 0 & 1
\end{bmatrix}.$$ 

1,0,0,0,0,3,0,0,0,0,3,0,0,0,0,1,1, 
1,0,0,0,0,2,0,0,3,0,2,0,0,3,0,1,1, 
1,0,0,0,3,2,0,0,0,0,2,0,0,0,3,1,1, 
1,0,0,0,3,0,0,0,0,3,0,0,0,1,1,0, 
1,0,0,0,3,2,0,0,2,0,2,0,3,1,3,1,0, 
1,1,0,0,3,0,0,0,0,0,3,0,0,1,0,1, 
1,2,0,0,0,1,0,0,2,2,1,3,0,3,0,1,1, 

nb received = 0

**Double Sixes**

The order of the group is 1974067200 
The group has 1 orbits:

The orbits are:

(1) \(0/1 \{16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0\}_{1440} \) orbit length 1370880 

Strong generators for a group of order 1440:

$$\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
\omega^2 & 0 & \omega^2 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & \omega & 0 & 0 \\
\omega & 0 & \omega & 0 \\
1 & \omega & 0 & 0
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
\omega^2 & 0 & \omega & 0 \\
\omega & \omega & \omega & 1
\end{bmatrix}, \begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & \omega & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega^2 & \omega & \omega & 1
\end{bmatrix}. $$

1,0,0,0,0,3,0,0,0,0,3,0,0,0,0,1,1, 
1,0,0,0,0,3,0,0,3,0,3,0,0,1,0,1, 
1,0,0,0,3,0,2,0,2,2,0,0,3,3,1,1,0, 
1,0,0,0,2,0,2,0,1,2,0,0,2,1,2,1,1, 
0,0,1,0,0,2,1,1,0,3,0,3,1,3,2,0, 
1,1,0,0,3,0,0,0,0,0,3,3,0,0,1,0,1,
The overall number of objects is: 1370880

**Flag orbits for surfaces**

The number of primary orbits below is 1
The number of primary orbits above is 1
The number of flag orbits is 1
The flag orbits are:

(1) Flag orbit 0 / 1 down=(0,0) up=(0,-1) is ( 16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0, 71, 55, 26, 100, 1, 138, 109, 345, 84, 85, 122, 110, 145, 139, 81 )
with a stabilizer of order 1440

Strong generators for a group of order 1440:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & 1 & \omega^2 \\
0 & 1 & 0 & \omega^2
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & 1 & \omega^2 \\
0 & 1 & 0 & \omega^2
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
1 & \omega & 0 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
1 & \omega & 0 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
\omega & \omega & 0 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
\omega & \omega & 0 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
\omega & \omega & 0 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
\omega & \omega & 0 & 0
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & 0 & \omega & 0 \\
\omega & \omega & 0 & 0
\end{bmatrix}
\]

Surfaces

The order of the group is 1974067200
The group has 1 orbits:

The orbits are:

(1) 0/1 \{16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0, 71, 55, 26, 100, 1, 138, 109, 345, 84, 85, 122, 110, 145, 139, 81\}_{51840} orbit length 38080
Strong generators for a group of order 51840:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & \omega^2 \\
\end{bmatrix}, \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
\end{bmatrix}, \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & \omega & 0 \\
1 & 0 & \omega^2 & 1 \\
0 & 1 & 0 & 0 \\
\end{bmatrix}, \\
\begin{bmatrix}
\omega^2 & \omega & \omega^2 & 1 \\
\omega^2 & 0 & 1 & 0 \\
\omega & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
\end{bmatrix}, \\
\begin{bmatrix}
\omega & \omega & 1 & 1 \\
\omega & \omega & 0 & 0 \\
\omega^2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\end{bmatrix}, \\
\begin{bmatrix}
1,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1, \\
1,0,0,0,2,0,0,0,0,2,0,0,0,0,1,0, \\
1,0,0,0,3,0,0,0,0,3,0,1,0,0,1,0, \\
1,0,0,0,1,0,0,1,1,1,0,1,1,0,1,0, \\
1,0,0,0,3,2,2,0,0,0,2,0,1,0,3,1,0, \\
1,0,0,0,1,0,2,0,2,2,0,0,2,2,1,1,0, \\
1,3,1,2,1,0,2,0,3,2,0,0,2,0,0,0,0, \\
1,1,3,3,0,3,0,1,1,2,0,1,0,3,0,0,0, \\
\end{bmatrix}
\]

The overall number of objects is: 38080

**The Group** PGL(4, 4)

The order of the group is 1974067200

**Cubic Surfaces with 27 Lines in** PG(3, 4)

The order of the group is 1974067200

The group has 1 orbits:

The orbits are:
(1) 0/1 {16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0, 71, 55, 26, 100, 1, 138, 109, 345, 84, 85, 122, 110, 145, 139, 81} \_51840 orbit length 38080

Strong generators for a group of order 51840:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
, \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix}
, \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix}
, \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix}
, \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega^2 & \omega & \omega \\
0 & 1 & 0 \\
1 & 0 & 0
\end{bmatrix}
, \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & \omega^2 & 1 \\
0 & 1 & 0 \\
0 & 1 & 0
\end{bmatrix}

1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,  \\
1,0,0,0,0,2,0,0,0,0,2,0,0,0,0,1,0,  \\
1,0,0,0,0,3,0,0,0,0,3,0,1,0,0,1,0,  \\
1,0,0,0,0,1,0,0,1,1,1,0,1,1,0,1,0,  \\
1,0,0,0,3,2,2,0,0,0,2,0,1,0,3,1,0,  \\
1,0,0,0,1,0,2,0,2,0,0,2,1,1,0,  \\
1,3,1,2,1,0,2,0,3,2,0,0,2,0,0,0,0,  \\
1,1,3,3,0,3,0,1,1,2,0,1,0,3,0,0,0,

The overall number of objects is: 38080

**Surface 4#0**

**The equation**

The equation of the surface is:

\[
X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_1^2 = 0
\]

( 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 )

Number of points on the surface 45

The automorphism group of the surface has order 51840
The automorphism group is the following group

Strong generators for a group of order 51840:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & \omega^2
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & \omega & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & \omega^2
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
\omega & \omega^2 & 1 \\
\omega^2 & 0 & 1 \\
\omega & 1 & 0 \\
1 & 0 & 0
\end{bmatrix},
\begin{bmatrix}
\omega & \omega^2 & 1 \\
\omega & 0 & 1 \\
\omega & 1 & 0 \\
0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
\omega & \omega^2 & 1 \\
\omega^2 & 0 & 1 \\
\omega & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\]

1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,2,0,0,0,0,0,1,0,1,0,0,0,0,3,0,0,0,0,3,0,1,0,0,1,0,0,0,0,3,0,0,0,0,3,0,1,0,0,1,0,0,0,0,3,0,2,2,0,0,0,0,2,0,1,0,3,1,0,1,0,0,0,1,0,2,0,2,2,0,0,2,2,1,1,0,1,0,0,0,1,0,2,0,2,0,0,0,2,0,0,0,0,0,1,1,3,3,0,3,0,1,1,2,0,1,0,3,0,0,0,

General information

Points on lines: 5^{27}

Lines on points: 3^{45}

The 27 Lines

\[\ell_0 = a_1 = \begin{bmatrix} 1 & 0 & \omega^2 & 0 \\
0 & 1 & 1 & \omega \end{bmatrix}_{72} = \begin{bmatrix} 1 & 0 & 3 & 0 \\
0 & 1 & 1 & 2 \end{bmatrix}_{72} = \text{Pl}(3,2,3,0,3,1)_{308}\]

\[\ell_1 = a_2 = \begin{bmatrix} 1 & 0 & \omega & 0 \\
0 & 1 & 0 & \omega^2 \end{bmatrix}_{54} = \begin{bmatrix} 1 & 0 & 2 & 0 \\
0 & 1 & 0 & 3 \end{bmatrix}_{54} = \text{Pl}(2,3,0,2,1)_{238}\]
\ell_2 = a_3 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{25} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{25} = \text{Pl}(1,1,0,0,1,1)_{177}

\ell_3 = a_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{356} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{356} = \text{Pl}(0,1,0,0,0,0)_{1}

\ell_4 = a_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}_0 = \text{Pl}(1,0,0,0,0,0)_{0}

\ell_5 = a_6 = \begin{bmatrix} 1 & 0 & \omega^2 & 1 \\ 0 & 1 & 0 & \omega \end{bmatrix}_{155} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 0 & 1 & 0 & 2 \end{bmatrix}_{155} = \text{Pl}(3,2,0,2,3,1)_{314}

\ell_6 = b_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{340} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{340} = \text{Pl}(0,0,0,1,0,0)_{9}

\ell_7 = b_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{38} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{38} = \text{Pl}(0,0,1,1,1,1)_{198}

\ell_8 = b_3 = \begin{bmatrix} 1 & \omega & 0 & 0 \\ 0 & 0 & 1 & \omega^2 \end{bmatrix}_{61} = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix}_{61} = \text{Pl}(0,0,2,3,2,1)_{265}

\ell_9 = b_4 = \begin{bmatrix} 1 & 0 & \omega & 1 \\ 0 & 1 & 1 & \omega \end{bmatrix}_{156} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}_{156} = \text{Pl}(3,0,3,2,3,1)_{335}

\ell_{10} = b_5 = \begin{bmatrix} 1 & \omega & 0 & 1 \\ 0 & 0 & 1 & \omega \end{bmatrix}_{165} = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}_{165} = \text{Pl}(0,2,3,2,3,1)_{337}

\ell_{11} = b_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{16} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{16} = \text{Pl}(0,0,1,0,0,0)_{2}

\ell_{12} = c_{12} = \begin{bmatrix} 1 & 0 & \omega & 1 \\ 0 & 1 & 0 & \omega^2 \end{bmatrix}_{138} = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 3 \end{bmatrix}_{138} = \text{Pl}(2,3,0,3,2,1)_{256}

\ell_{13} = c_{13} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{109} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{109} = \text{Pl}(1,1,0,1,1,1)_{189}

\ell_{14} = c_{14} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{345} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{345} = \text{Pl}(0,1,0,1,0,0)_{13}

\ell_{15} = c_{15} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}_{84} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}_{84} = \text{Pl}(1,0,0,1,0,0)_{10}
\begin{align*}
\ell_{16} &= c_{16} = \begin{bmatrix} 1 & 0 & \omega^2 & 0 \\ 0 & 1 & 0 & \omega \end{bmatrix}_{71} = \begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}_{71} = \text{Pl}(3, 2, 0, 0, 3, 1)_{299} \\
\ell_{17} &= c_{23} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{85} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{85} = \text{Pl}(1, 1, 1, 1, 0, 0)_{16} \\
\ell_{18} &= c_{24} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{122} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{122} = \text{Pl}(0, 1, 1, 1, 1, 0)_{202} \\
\ell_{19} &= c_{25} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{110} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{110} = \text{Pl}(1, 0, 1, 1, 1, 1)_{199} \\
\ell_{20} &= c_{26} = \begin{bmatrix} 1 & 0 & \omega & 0 \\ 0 & 1 & 1 & \omega^2 \end{bmatrix}_{55} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 3 \end{bmatrix}_{55} = \text{Pl}(2, 3, 2, 0, 2, 1)_{244} \\
\ell_{21} &= c_{34} = \begin{bmatrix} 1 & \omega & 0 & 1 \\ 0 & 0 & 1 & \omega^2 \end{bmatrix}_{145} = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 3 \end{bmatrix}_{145} = \text{Pl}(0, 3, 2, 3, 2, 1)_{271} \\
\ell_{22} &= c_{35} = \begin{bmatrix} 1 & 0 & \omega & 1 \\ 0 & 1 & 1 & \omega^2 \end{bmatrix}_{139} = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 3 \end{bmatrix}_{139} = \text{Pl}(2, 0, 2, 3, 2, 1)_{267} \\
\ell_{23} &= c_{36} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{26} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{26} = \text{Pl}(1, 1, 1, 0, 1, 1)_{180} \\
\ell_{24} &= c_{45} = \begin{bmatrix} 1 & \omega^2 & 0 & 0 \\ 0 & 0 & 1 & \omega \end{bmatrix}_{81} = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}_{81} = \text{Pl}(0, 0, 3, 2, 3, 1)_{332} \\
\ell_{25} &= c_{46} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{100} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{100} = \text{Pl}(0, 1, 1, 0, 0, 0)_{6} \\
\ell_{26} &= c_{56} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{1} = \text{Pl}(1, 0, 1, 0, 0, 0)_{3} \\
\end{align*}

Rank of lines: ( 72, 54, 25, 356, 0, 155, 340, 38, 61, 156, 165, 16, 138, 109, 345, 84, 71, 85, 122, 110, 55, 145, 139, 26, 81, 100, 1 )

Rank of points on Klein quadric: ( 308, 238, 177, 1, 0, 314, 9, 198, 265, 335, 337, 2, 256, 189, 13, 10, 299, 16, 202, 199, 244, 271, 267, 180, 332, 6, 3 )

**All Points on surface**

The surface has 45 points
Eckardt Points

The surface has 45 Eckardt points:

0 : $E_{56} = a_5 \cap b_6 \cap c_{56} = P_0 = P_0 = P(1,0,0,0) = P(0,1,0,0)$, $T = 0$
1 : $E_{51} = a_5 \cap b_1 \cap c_{15} = P_1 = P_1 = P(0,1,0,0) = P(0,1,0,0)$, $T = 4$
2 : $E_{46} = a_4 \cap b_6 \cap c_{46} = P_2 = P_2 = P(0,0,1,0) = P(0,0,1,0)$, $T = 20$
3 : $E_{41} = a_4 \cap b_1 \cap c_{14} = P_3 = P_3 = P(0,0,0,1) = P(0,0,0,1)$, $T = 84$
4 : $E_{32} = a_3 \cap b_2 \cap c_{23} = P_4 = P_4 = P(1,1,1,1) = P(1,1,1,1)$, $T = 27$
5 : $E_{52} = a_5 \cap b_2 \cap c_{25} = P_5 = P_5 = P(1,1,0,0) = P(1,1,0,0)$, $T = 1$
6 : $E_{54} = a_5 \cap b_4 \cap c_{45} = P_6 = P_6 = P(\omega,1,0,0) = P(2,1,0,0)$, $T = 2$
7 : $E_{53} = a_5 \cap b_3 \cap c_{35} = P_7 = P_7 = P(\omega^2,1,0,0) = P(3,1,0,0)$, $T = 3$
8 : $E_{36} = a_3 \cap b_6 \cap c_{36} = P_8 = P_8 = P(1,0,1,0) = P(0,1,0,1)$, $T = 5$
9 : $E_{16} = a_1 \cap b_6 \cap c_{16} = P_9 = P_9 = P(\omega,0,1,0) = P(2,0,1,0)$, $T = 10$
10: $E_{26} = a_2 \cap b_6 \cap c_{26} = P_{10} = P_{10} = P(\omega,0,1,0) = P(3,0,1,0)$, $T = 15$
11: $E_{14,23,56} = c_{14} \cap c_{23} \cap c_{56} = P_{11} = P_{11} = P(0,1,1,0) = P(0,1,1,0)$, $T = 9$
12: $E_{13,24,56} = c_{13} \cap c_{24} \cap c_{56} = P_{12} = P_{12} = P(1,1,1,0) = P(1,1,1,0)$, $T = 6$
13: $E_{65} = a_6 \cap b_5 \cap c_{56} = P_{13} = P_{13} = P(\omega,1,1,0) = P(2,1,1,0)$, $T = 12$
14: $E_{12,34,56} = c_{12} \cap c_{34} \cap c_{56} = P_{14} = P_{14} = P(\omega,1,1,0) = P(3,1,1,0)$, $T = 18$
15: $E_{15,23,46} = c_{15} \cap c_{23} \cap c_{46} = P_{15} = P_{23} = P(1,0,0,1) = P(1,0,0,1)$, $T = 21$
16: $E_{31} = a_3 \cap b_1 \cap c_{13} = P_{16} = P_{26} = P(0,1,0,1) = P(0,1,0,1)$, $T = 25$
17: $E_{15,24,36} = c_{15} \cap c_{24} \cap c_{36} = P_{17} = P_{27} = P(1,1,0,1) = P(1,1,0,1)$, $T = 22$
18: $E_{21} = a_2 \cap b_1 \cap c_{12} = P_{18} = P_{30} = P(0,0,0,1) = P(0,0,0,1)$, $T = 46$
19: $E_{15,26,34} = c_{15} \cap c_{26} \cap c_{34} = P_{19} = P_{31} = P(1,\omega,0,1) = P(1,2,0,1)$, $T = 24$
20: $E_{61} = a_6 \cap b_1 \cap c_{14} = P_{20} = P_{34} = P(0,\omega,0,1) = P(0,3,0,1)$, $T = 67$
21: $E_{15} = a_1 \cap b_5 \cap c_{15} = P_{21} = P_{35} = P(1,\omega,0,1) = P(1,3,0,1)$, $T = 23$
22: $E_{42} = a_4 \cap b_2 \cap c_{24} = P_{22} = P_{38} = P(0,0,1,1) = P(0,0,1,1)$, $T = 41$
23: $E_{13,25,46} = c_{13} \cap c_{25} \cap c_{46} = P_{23} = P_{39} = P(1,0,1,1) = P(1,0,1,1)$, $T = 26$
24: $E_{14,25,36} = c_{14} \cap c_{25} \cap c_{36} = P_{24} = P_{42} = P(0,1,1,1) = P(0,1,1,1)$, $T = 30$
25: $E_{62} = a_6 \cap b_2 \cap c_{26} = P_{25} = P_{47} = P(\omega,\omega,1,1) = P(2,2,1,1)$, $T = 53$
26: $E_{25} = a_2 \cap b_5 \cap c_{25} = P_{26} = P_{48} = P(\omega,\omega,1,1) = P(3,2,1,1)$, $T = 80$
27: $E_{16,25,34} = c_{16} \cap c_{25} \cap c_{34} = P_{27} = P_{51} = P(\omega,\omega,1,1) = P(2,3,1,1)$, $T = 55$
28: $E_{12} = a_1 \cap b_2 \cap c_{12} = P_{28} = P_{32} = P(\omega,\omega,2,1) = P(3,3,1,1)$, $T = 79$
29: $E_{43} = a_4 \cap b_3 \cap c_{34} = P_{29} = P_{53} = P(0,0,1,1) = P(0,0,2,1)$, $T = 62$
30: $E_{12,35,46} = c_{12} \cap c_{35} \cap c_{46} = P_{30} = P_{54} = P(1,0,0,1) = P(1,0,2,1)$, $T = 36$
31: $E_{35} = a_3 \cap b_5 \cap c_{35} = P_{31} = P_{59} = P(\omega,\omega,1,1) = P(2,1,2,1)$, $T = 49$
32: $E_{63} = a_6 \cap b_3 \cap c_{36} = P_{32} = P_{60} = P(\omega,2,1,0,1) = P(3,1,2,1)$, $T = 76$
33: $E_{14,26,35} = c_{14} \cap c_{26} \cap c_{35} = P_{33} = P_{61} = P(0,0,1,1) = P(0,2,2,1)$, $T = 51$
34: $E_{23} = a_2 \cap b_3 \cap c_{23} = P_{34} = P_{62} = P(1,\omega,0,1) = P(1,2,2,1)$, $T = 39$
35: $E_{13} = a_1 \cap b_3 \cap c_{13} = P_{35} = P_{67} = P(\omega,\omega,0,1) = P(2,3,2,1)$, $T = 50$
36: $E_{16,24,35} = c_{16} \cap c_{24} \cap c_{35} = P_{36} = P_{68} = P(\omega,\omega,2,0,1) = P(3,3,2,1)$, $T = 74$
37: $E_{45} = a_4 \cap b_5 \cap c_{45} = P_{37} = P_{69} = P(0,0,2,1) = P(0,0,3,1)$, $T = 83$
38: $E_{64} = a_6 \cap b_4 \cap c_{46} = P_{38} = P_{70} = P(1,0,2,0,1) = P(1,0,3,1)$, $T = 31$
39: $E_{12,36,45} = c_{12} \cap c_{36} \cap c_{45} = P_{39} = P_{75} = P(\omega,1,\omega,1) = P(2,1,3,1)$, $T = 59$
40: $E_{34} = a_3 \cap b_4 \cap c_{34} = P_{40} = P_{76} = P(\omega,2,1,2,0) = P(3,1,3,1)$, $T = 71$
$E_{24} = a_2 \cap b_4 \cap c_{24} = P_{41} = P_{79} = P(\omega, \omega, \omega^2, 1) = P(2, 2, 3, 1), \ T = 58$

$E_{13,26,45} = c_{13} \cap c_{26} \cap c_{45} = P_{42} = P_{80} = P(\omega^2, \omega, \omega^2, 1) = P(3, 2, 3, 1), \ T = 70$

$E_{14} = a_1 \cap b_4 \cap c_{14} = P_{43} = P_{81} = P(0, \omega^2, \omega^2, 1) = P(0, 3, 3, 1), \ T = 72$

$E_{16,23,45} = c_{16} \cap c_{23} \cap c_{45} = P_{44} = P_{82} = P(1, \omega^2, \omega^2, 1) = P(1, 3, 3, 1). \ T = 33$

Set of tangent planes: \( \{ 0, 4, 20, 84, 27, 1, 2, 3, 5, 10, 15, 9, 6, 12, 18, 21, 25, 22, 46, 24, 67, 23, 41, 26, 30, 53, 80, 55, 79, 62, 36, 49, 76, 51, 39, 50, 74, 83, 31, 59, 71, 58, 70, 72, 33 \} \)

Line type of Eckardt points: \( 5^{27}, 3^{240}, 1^{90} \)

Plane type of Eckardt points: \( 13^{45}, 9^{40} \)

Hesse planes

Number of Hesse planes: 40

Set of Hesse planes: \( \{ 7, 8, 11, 13, 14, 16, 17, 19, 28, 29, 32, 34, 35, 37, 38, 40, 42, 43, 44, 45, 47, 48, 52, 54, 56, 57, 60, 61, 63, 64, 65, 66, 68, 69, 73, 75, 77, 78, 81, 82 \} \)

subspace 0 / 40 is 7:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & \omega
\end{bmatrix}
\]

subspace 39 / 40 is 82:

\[
\begin{bmatrix}
1 & 0 & \omega^2 & 0 \\
0 & 1 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

0 : 7 : \( E_{56}, E_{31}, E_{15,24,36}, E_{16,25,34}, E_{12}, E_{14,26,35}, E_{23}, E_{45}, E_{64} \)

: \quad 39 : 82 : \( E_{41}, E_{52}, E_{16}, E_{12,34,56}, E_{15,24,36}, E_{35}, E_{23}, E_{64}, E_{13,26,45} \)

Axes

Number of axes: 240

Axes:

0 : 0 = 0,0 = E_{23}, E_{31}, E_{12}

: \quad 239 : 239 = 119,1 = E_{12,36,45}, E_{14,26,35}, E_{13,25,46}

229
Tritangent planes

The 45 tritangent planes are:

\[
\pi_{12} = \pi_0 = 79 = \begin{bmatrix}
1 & 0 & 0 & \omega^2 \\
0 & 1 & 0 & \omega^2 \\
0 & 0 & 1 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 3 \\
0 & 1 & 0 & 3 \\
0 & 0 & 1 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 3 \\
0 & 1 & 0 & 3 \\
0 & 0 & 1 & 1
\end{bmatrix} = V(\omega^2 X_0 + \omega^2 X_1 + X_2 + X_3) = V(3X_0 + 3X_1 + X_2 + X_3)
\]

dual pt rank = 52 = (3, 3, 1, 1).

\[
\pi_{16,25,34} = \pi_{44} = 55 = \begin{bmatrix}
1 & 0 & 0 & \omega \\
0 & 1 & 0 & \omega \\
0 & 0 & 1 & \omega^2
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 2 \\
0 & 1 & 0 & 2 \\
0 & 0 & 1 & 3
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 2 \\
0 & 1 & 0 & 2 \\
0 & 0 & 1 & 3
\end{bmatrix} = V(\omega X_0 + \omega X_1 + \omega^2 X_2 + X_3) = V(2X_0 + 2X_1 + 3X_2 + X_3)
\]

dual pt rank = 79 = (2, 2, 3, 1).

Karaoglu [36] describes a different algorithm, based on non-conical six-arcs and trihedral pairs. The command

```
surface_classify_q4_arc_lifting_two_lines:
  $(ORBITER_PATH)orbiter.out -v 10 \n  -define F -finite_field -q 4 -end \n  -define P -projective_space 3 F -end \n  -with P -do \n  -projective_space_activity \n  -control six arcs -problem_label sixarcs q4 -end \n  -classify_surfaces_through_arcs_and_two_lines \n  -end
```

classifies all cubic surfaces with 27 lines over the field \( \mathbb{F}_4 \) using the algorithm of Karaoglu. The result agrees with the previous algorithm. The only surface with 27 lines in \( \text{PG}(3, 4) \) is the Hirschfeld surface.
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-surface_identify_Eckardt</td>
<td></td>
<td>Identifies the isomorphism type of the Eckardt surface with parameter a. All values of a are considered.</td>
</tr>
<tr>
<td>-surface_identify_F13</td>
<td></td>
<td>Identifies the isomorphism type of the $F_{13}$ surface with parameter a. All values of a are considered.</td>
</tr>
<tr>
<td>-surface_identify_Bes</td>
<td></td>
<td>Identifies the isomorphism type of the Bes surface with parameters a and c. All values of a, c are considered.</td>
</tr>
<tr>
<td>-surface_identify_general_abcd</td>
<td>surface-descr-1, surface-descr-2</td>
<td>Identifies the isomorphism type of the general surface with parameters a, b, c, d. All values of a, b, c, d are considered.</td>
</tr>
<tr>
<td>-surface_isomorphism_testing</td>
<td>surface-descr-1, surface-descr-2</td>
<td>Computes an isomorphism between two given surfaces or concludes that none exists.</td>
</tr>
<tr>
<td>-surface_recognize</td>
<td>surface-descr</td>
<td>Identifies the isomorphism type of the given surface.</td>
</tr>
<tr>
<td>-create_surface</td>
<td>surface-descr</td>
<td>Creates a surface from a description. See Section 7.1.</td>
</tr>
</tbody>
</table>

Table 7.4: Projective space activities related to the recognition of cubic surfaces

### 7.4 Isomorphism Testing and Recognition

Besides classification, Orbiter provides recognition, isomorphism testing and study of cubic surfaces. Table 7.4 lists the relevant Orbiter commands. These commands are projective space activities.

The `-surface_recognize` option can be used to identify a given surface in the list produced by the classification. The command computes an isomorphism between the given surface and the surface in the catalogue. For instance,

```
surface_recognize q7_abcd_2_3_3_4:
  ▶ $(ORBITER_PATH)orbiter.out--v.3\n  ▶ ▶ -define:F:finite_field:-q:7:-end\n  ▶ ▶ -define:P:projective_space-3:F:-end\n  ▶ ▶ -with:P:-do\n  ▶ ▶ -projective_space_activity\n  ▶ ▶ ▶ -classify_surfaces_with_double_sixes_Surf:-W:-end\n  ▶ ▶ -end\n  ▶ -with:Surf:-do\n  ▶ ▶ -classification_of_cubic_surfaces_with_double_sixes_activity\n  ▶ ▶ -recognize\n```
identifies the surface (cf. Table 4.3)

\[ X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 + X_1X_2X_3 = 0 \]  \hspace{1cm} (7.1)

in the classification of surfaces over the field \( F_7 \). This means that an isomorphism from the given surface to the surface in the list is computed. Also, the generators of the automorphism group of the given surface are computed, using the known generators for the automorphism group of the surface in the classification. For instance, executing the command above produces the isomorphism

\[
\begin{bmatrix}
1 & 4 & 4 & 0 \\
6 & 0 & 0 & 0 \\
6 & 2 & 0 & 1 \\
7 & 0 & 4 & 0 \\
\end{bmatrix}
\]  \hspace{1cm} (7.2)

Orbiter can compute isomorphism between two given surfaces. Both surfaces must have 27 lines. For instance, the command

\begin{verbatim}
surface_isomorph_16:
\end{verbatim}
computes an isomorphism between two cubic surfaces with 27 lines

\[ X_0^2X_2 + X_1^2X_2 + X_1X_2^2 + X_0X_2^2 + X_1X_2^2 + X_2^2X_3 + \delta^{13}X_1X_3^2 + \delta^{13}X_2X_3^2 + = 0 \]

and

\[ \delta^{11}X_0^2X_3 + \delta^{12}X_1^2X_2 + \delta^{12}X_1X_2^2 + \delta^{11}X_0X_3^2 + X_1X_2X_3 = 0 \]

over the field \( \mathbb{F}_{16} \).

\[
\begin{bmatrix}
12 & 13 & 0 & 0 \\
8  & 13 & 0 & 0 \\
0  & 0  & 13 & 0 \\
12 & 13 & 11 & 1 \\
\end{bmatrix}
\]

Orbiter can recognize the isomorphism type of a cubic surface with 27 lines. This means that Orbiter can determine the Orbiter Catalogue Number of the surface in the catalogue which is isomorphic to the given surface. For instance, the following command determines the isomorphism type of the surface

\[ X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 + X_1X_2X_3 = 0. \]

```
surface_recognize_8:
  > $(ORBITER_PATH)orbiter.out --v 3 -
  > -define F: finite_field -q 8 -
  > -define P: projective_space -3 F -
  > -with P: do -
  > -projective_space_activity -
  > -classify_surfaces_with_double_sixes Surf27 -W -
  > -end -
  > -with Surf27 -do -
  > -classification_of_cubic_surfaces_with_double_sixes_activity -
  > -recognize -
  > -by_coefficients "1,6,1,8,1,11,1,13,1,19" -
  > -end -
  > -end -
  > -print_symbols
```

The command find that the surface is isomorphic to the surface with OCN=0. An isomorphism will be computed as well.
7.5 Dickson Surfaces

For very small values of $q$, the cubic surfaces over $\mathbb{F}_q$ can be classified using the basic Schreier algorithm from Section 6.1. Let us look at an example. Suppose we want to classify all cubic surfaces in $\text{PG}(3, 2)$. The non-singular ones have been classified by Dickson [23]. Orbiter can be used to recreate this classification and to investigate these surfaces further.

The classification of all cubic surfaces in $\text{PG}(3, 2)$ can be done using this Orbiter command:

```bash
poly_orbits d3 n3 q2.csv:
```

```
$\text{(ORBITER PATH)}\text{orbiter.out} -v 4
 dél -draw_options -yout 500000 -radius 15 -nodes_empty
 dél dél -line_width 0.5 -y_stretch 0.25 -embedded -end
 dél dél -define G -linear_group -PGL 4 2 -end
 dél dél -with G -do
 dél dél -group_theoretic_activity
 dél dél dél -orbits_on_polynomials 3
 dél dél dél -orbits_on_polynomials_draw_tree 6
 dél dél -end.
```

To investigate the properties of these surfaces, the following commands can be used:

```bash
poly_orbits_d3_n3_q2_F2.csv: poly_orbits_d3_n3_q2.csv
```

```
$\text{(ORBITER PATH)}\text{orbiter.out} -v 4
 dél dél -define F -finite_field -q 2 -end
 dél dél -define P -projective_space 3 F -end
 dél dél -with P -do
 dél dél -projective_space_activity
 dél dél -table_of_cubic_surfaces_compute_properties
 dél dél dél poly_orbits_d3_n3_q2.csv 2.0
 dél dél -end.
```

Dickson_q2_analyze: poly_orbits_d3_n3_q2_F2.csv

```
$\text{(ORBITER PATH)}\text{orbiter.out} -v 4
 dél dél -define F -finite_field -q 2 -end
 dél dél -define P -projective_space 3 F -end
 dél dél -with P -do
 dél dél -projective_space_activity
 dél dél -cubic_surface_properties_analyze
 dél dél dél poly_orbits_d3_n3_q2_F2.csv 2
 dél dél -end
 dél pdflatex poly_orbits_d3_n3_q2_F2_report.tex
 dél open poly_orbits_d3_n3_q2_F2_report.pdf
```

To investigate the properties of these surfaces over the extension field $\mathbb{F}_4$, the following commands can be used:
poly_orbits_d3_n3_q2_F4.csv:
poly_orbits_d3_n3_q2.csv

$($(ORBITER_PATH)orbiter.out)$-v4:\
$\define F$-finite_field$-q4$-end:\
$\define P$-projective_space$3F$-end:\
$\with P$-do:\
$\projective_space_activity$:\
$\table_of_cubic_surfaces$-compute_properties:\
$ poly_orbits_d3_n3_q2.csv$-2.0:\
$\text{end}$. 

Dickson_q4_analyze:
poly_orbits_d3_n3_q2_F4.csv

$($(ORBITER_PATH)orbiter.out)$-v4:\
$\define F$-finite_field$-q4$-end:\
$\define P$-projective_space$3F$-end:\
$\with P$-do:\
$\projective_space_activity$:\
$\cubic_surface_properties$-analyze:\
$ poly_orbits_d3_n3_q2_F4.csv$-2:\
$\text{end}$. 

pdflatex:
poly_orbits_d3_n3_q2_F4_report.tex
open:
poly_orbits_d3_n3_q2_F4_report.pdf
The data in Orbiter can be exported to be used for automated processing. It is possible to create a csv file with the cubic surfaces with 27 lines for a given $q$. The following example shows how to export the data about cubic surfaces with $q = 17$:

```
MAKE_TABLE_OF_CUBIC_SURFACES=-define P:-projective_space 3:F:-end.\n  -with P:-do.\n  -projective_space_activity.\n  -table_of_cubic_surfaces.\n  -end\n
cubic_surfaces_tables_17: .\n  $(ORBITER_PATH)orbiter.out-v.3.\n  -define F:-finite_field-q.17.-end.\n  $(MAKE_TABLE_OF_CUBIC_SURFACES)
```

A file `table_of_cubic_surfaces_q17_info.csv` is created. The command

```
cubic_surfaces_table_latex_17: .\n  $(ORBITER_PATH)orbiter.out-v.3.-csv_file_latex.1.\n  table_of_cubic_surfaces_q17_info.csv
```

produces a latex table from the csv file.
Chapter 8

Ring Theory

8.1 Polynomials Over Finite Fields

For $p$ prime, the finite field $\mathbb{F}_p$ of order $p$ can be constructed as factorring of the integers modulo $p$. In this section, we will consider polynomials over $\mathbb{F}_p$. The ring of polynomials in one variable with coefficients in $\mathbb{F}_p$ is denoted as $\mathbb{F}_p[X]$.

The \texttt{-finite_field_activity \ ... \ -end} command sequence can be used to start a command requiring a finite field. The \texttt{-q q} option can be used to specify the order of the finite field. The \texttt{-override_polynomial a} option can be used to specify the polynomial $m(X)$ as integer $a$ in the base $p$ representation. This option can be omitted, in which case Orbiter will use a precomputed and built-in polynomial. Table 8.1 lists Orbiter activities for polynomials over finite fields. For instance, the command

\begin{verbatim}
poly_division:
  $(ORBITER_PATH)orboiter.out-v2-
  -define:Finite_field-q2-end-
  -with:Finite_field-
  -finite_field_activity-
  -polynomial_division:"1,0,0,0,0,0,0,0,1","1,0,1,1"-end
\end{verbatim}

computes the polynomial long division of $A(X)$ by $B(X)$ over $\mathbb{F}_2$ where

\[ A(X) = X^{10} + 1, \quad B(X) = X^3 + X^2 + 1. \]

The result is $Q(X)$ and $R(X)$ with

\[ A(X) = Q(X) \cdot B(X) + R(X) \]

with

\[ Q(X) = X^7 + X^6 + X^5 + X^3 + 1, \quad R(X) = X^2. \]
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-polynomial_division</td>
<td>$A(X) \ B(X)$</td>
<td>Polynomial division of $A(X)$ by $B(X)$ over $\mathbb{F}_q$. $A(X)$ and $B(X)$ are given as coefficient list, starting from the lowest coefficient.</td>
</tr>
<tr>
<td>-extended_gcd_for_polynomials</td>
<td>$A(X) \ B(X)$</td>
<td>Extended gcd for polynomials $A(X)$ and $B(X)$ over $\mathbb{F}_q$. $A(X)$ and $B(X)$ are given as coefficient list, starting from the lowest coefficient.</td>
</tr>
<tr>
<td>-polynomial_mult_mod</td>
<td>$A(X) \ B(X)$</td>
<td>Multiply the polynomials $A(X)$ and $B(X)$ modulo $M(X)$ in $\mathbb{F}_q[X]$.</td>
</tr>
</tbody>
</table>
| -Berlekamp_matrix            | $A(X)$          | Computes the rank of the Berlekamp matrix associated to the polynomial $A(X)$ over $\mathbb{F}_q$. The polynomial $A(X)$ is irreducible over $\mathbb{F}_q$ if the Berlekamp matrix has rank $d - 1$ where $d$ is the degree of $A(X)$. The Berlekamp matrix is $F - I$ where $F$ is the Frobenius matrix and $I$ is the identity matrix. The Frobenius matrix is the matrix of the Frobenius endomorphism with respect to the standard basis of the polynomial ring: $1, X, X^2, \ldots, X^{d-1}$.
| -polynomial_find_roots       | $A(X)$          | Find the roots of $A(X)$ over $\mathbb{F}_q$.                                                                                              |
| -make_table_of_irreducible_polynomials | $d$           | Produces a list of all irreducible polynomials of degree $d$ over $\mathbb{F}_q$.                                                               |
| -find_CRC_polynomials        | $t \ n \ k$     | Computes all CRC polynomials of degree $k$ over $\mathbb{F}_q$ who detect all error patterns of Hamming weight $t$ or less in messages of length $n$. See Section 10.4. |

Table 8.1: Finite Field Activities Related to Polynomials
The coefficient lists in the arguments are from the lowest term up.

It is perhaps more convenient to use the vector builder from Section 2.7 to create the polynomials. The following example illustrates this. First, the coefficient vectors of the two polynomials are created using a `define` command. The vectors are symbolic variables named $A$ and $B$. After that, the division command is called as a finite field activity for $F$. The division command creates the polynomials from the coefficient vectors automatically. Note the difference in how the vectors are created.

```plaintext
poly_division2:
> $(ORBITER_PATH)orbiter.out -v 2 \n> -define F -finite_field -q 2 -end \n> -define A -vector -field F -sparse "1,0,1,10" -end \n> -define B -vector -field F -dense "1,0,1,1" -end \n> -finite_field_activity \n> -polynomial_division A B -end
```

The command `-extended_gcd_for_polynomials` takes two polynomials $A(X)$ and $B(X)$ and computes polynomials $U(X)$ and $V(X)$ and $G(X)$ such that $G(X)$ is the greatest common divisor of $A(X)$ and $B(X)$ and

$$G(X) = U(X) \cdot A(X) + V(X) \cdot B(X).$$

For instance,

```plaintext
poly_gcd:
> $(ORBITER_PATH)orbiter.out -v 2 \n> -define F -finite_field -q 2 -end \n> -finite_field_activity \n> -extended_gcd_for_polynomials "1,0,0,0,0,0,0,0,0,0,1" "1,0,1,1" -end
```

computes

$$U(X) = X + 1, \quad V(X) = X^8 + X^5 + X^4 + X^3 + X, \quad G(X) = 1.$$  

The next command computes

$$(3X^2 + 2X + 1) \cdot (5X^2 + 4X + 3) \mod (X^3 + 7) \mod 7.$$  

```plaintext
poly_mult_mod1:
> $(ORBITER_PATH)orbiter.out -v 2 \n> -define F -finite_field -q 7 -end \n> -finite_field_activity \n> -polynomial_mult_mod "1,2,3" "3,4,5" "6,0,0,1" -end
```
which has a result of
\[ X^2 + 4X + 4. \]

Observe how the coefficients are given from the lowest to the highest term. For the opposite order, the following command computes
\[
(2X^2 + X + 3) \cdot (4X^2 + 3X + 5) \pmod{X^3 + 7} \pmod{7}.
\]

The result is
\[ 4X^2 + X + 4. \]

The finite field \( \mathbb{F}_4 \) can be defined by using polynomial arithmetic over \( \mathbb{F}_2 \) modulo \( X^2 + X + 1 \). Here is a command that computes the three non-trivial products of polynomials:

```
poly_mult_mod_F4:
  > $(ORBITER_PATH)orbiter.out-v.2\n  > define F--finite_field-q.2-\n  > with F--do\n  > -finite_field_activity\n  > -polynomial_mult_mod"1,1","1,1","1,1,1"-end
  > $(ORBITER_PATH)orbiter.out-v.2\n  > define F--finite_field-q.2-\n  > with F--do\n  > -finite_field_activity\n  > -polynomial_mult_mod"0,1","1,1","1,1,1"-end
  > $(ORBITER_PATH)orbiter.out-v.2\n  > define F--finite_field-q.2-\n  > with F--do\n  > -finite_field_activity\n  > -polynomial_mult_mod"0,1","0,1","1,1,1"-end
```

It is possible to use numerical values for polynomials, using the representation in radix \( q \). The following command computes the product of the polynomials 5 and 7 over \( \mathbb{F}_2 \):

```
mult_polynomials_2.5.7:
  > $(ORBITER_PATH)orbiter.out-v.2\n  > define F--finite_field-q.2-\n```

240
The next command performs polynomial long division based on numerical polynomials:

```
polynomial.division.ranked.2.27.13:
  $(ORBITER_PATH)orbiter.out-v.2\n  -define F-finite_field-q.2-end\n  -with F-do\n  -finite_field_activity\n  polynomial.division.ranked.27.13\n  -end
  pdflatex-polynomial_division_27_13.tex
  open-polynomial_division_27_13.pdf
```

Here is a somewhat larger example for numerical arguments. We wish to multiply 999 by 997 modulo 1033. The first command performs multiplication:

```
mult.polynomials.1024.999.997:
  $(ORBITER_PATH)orbiter.out-v.2\n  -define F-finite_field-q.2-end\n  -with F-do\n  -finite_field_activity\n  mult.polynomials.999.997\n  -end
  pdflatex-polynomial_mult.999.997.tex
  open-polynomial_mult.999.997.pdf
```

The next command performs division with remainder:

```
polynomial.division.ranked.2.349147.1033:
  $(ORBITER_PATH)orbiter.out-v.2\n  -define F-finite_field-q.2-end\n  -with F-do\n  -finite_field_activity\n  polynomial.division.ranked.349147.1033\n  -end
  pdflatex-polynomial_division_349147_1033.tex
  open-polynomial_division_349147_1033.pdf
```
The next command performs an independent check, using the finite field with 1024 elements. This check relies on the fact that the irreducible polynomial to create the field \( F_{1024} \) is exactly the polynomial by which we did mod out in the example before:

```
mult_polynomials_1024_999_997_check:
▷ $(ORBITER_PATH)oriter.out:-v.3\n▷ ▷ -define:F:-finite_field:-q.1024:-end\n▷ ▷ -with:F:-do\n▷ ▷ -finite_field_activity:-parse_and_evaluate\n▷ ▷ "test"."".""a*b.""a=999,b=997".-end
```

In this last command, the formula \( a*b \) is used and evaluated over \( F_{1024} \), using \( a = 999 \) and \( b = 997 \).

Orbiter allows polynomial arithmetic modulo a factor polynomial. The coefficient vector of the polynomial can be created using the `vector` object type. Here is an example which performs arithmetic modulo the CRC32 polynomial. The goal is to compute the multiplicative inverse of \( X \). In order to do so, we use the fact that the CRC32 polynomial is irreducible, and hence the factor ring is a finite field of order \( 2^{32} \). The inverse of a polynomial can be computed by raising to the power of \( 2^{32} - 2 \):

```
CRC32_SPARSE="1,32,1,26,1,23,1,22,1,16,1,12,1,11,\n1,10,1,8,1,7,1,5,1,4,1,2,1,1,1,0"
```

```
TWO_TO_THE_32_MINUS_2=4294967294
```

```
power_mod_inverse:
▷ $(ORBITER_PATH)oriter.out:-v.2\n▷ ▷ -define:F:-finite_field:-q.2:-end\n▷ ▷ -define:M:-vector-field:F:-sparse.33$(CRC32_SPARSE).-end\n▷ ▷ -define:A:-vector-field:F:-sparse.2:"1,1".-end\n▷ ▷ -with:F:-do\n▷ ▷ -finite_field_activity\n▷ ▷ -polynomial_power_mod:A:$\langle$TWO_TO_THE_32_MINUS_2$\rangle$.M\n▷ ▷ -end
```

This command produces the polynomial

\[
B(X) = X^{31} + X^{25} + X^{22} + X^{21} + X^{15} + X^{11} + X^{10} + X^9 + X^7 + X^6 + X^4 + X^3 + X + 1
\]

In order to test that this polynomial really is the multiplicative inverse of \( X \) modulo CRC32, we perform the following command:

```
INVERSE_SPARSE="1,31,1,25,1,22,1,21,1,15,\n1,11,1,10,1,9,1,7,1,6,1,4,1,3,1,1,1,0"
```
The product is indeed 1.

The Berlekamp matrix can be used to test if a polynomial is irreducible over a given finite field. The polynomial is irreducible if and only if the rank of the Berlekamp matrix is $d - 1$, where $d$ is the degree of the polynomial. For instance, the command

```
Berlekamp_matrix_2_3:
```

computes the Berlekamp matrix associated with the polynomial $X^3 + X + 1$ over $\mathbb{F}_2$. The matrix is

$$
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 1 \\
0 & 1 & 0
\end{bmatrix}
$$

Since the matrix has rank 2, the polynomial is irreducible.

Orbiter can compute irreducible polynomials. For a given degree over a given field $\mathbb{F}_q$, we distinguish two tasks: The first task is finding one irreducible polynomial of the given degree and with the given field of coefficients. The second task is finding all irreducible polynomials given that one has already been found.

For instance, the command

```
search_primitive_poly_2:
```

would find irreducible polynomials of degree 2 over $\mathbb{F}_2$.
searches for primitive polynomials over $\mathbb{F}_2$ of degree 2 to 10. The unix command `grep` is used to filter the output for lines containing the given pattern “//”. This yields the list

```
"7", // X^{-2} + X + 1
"13", // X^{-3} + X^{-2} + 1
"25", // X^{-4} + X^{-3} + 1
"37", // X^{-5} + X^{-2} + 1
"97", // X^{-6} + X^{-5} + 1
"193", // X^{-7} + X^{-6} + 1
"285", // X^{-8} + X^{-4} + X^{-3} + X^{-2} + 1
"529", // X^{-9} + X^{-4} + 1
"1033", // X^{-10} + X^{-3} + 1
```

Primitive polynomials over the base field $\mathbb{F}_s$ are converted into integers, using the base-$s$ representation of integers. For instance, the polynomial $X^2 + X + 1$ is read as binary string 111, which in turn translates to the integer 7 (we use $s = 2$).

Regarding the problem of creating all irreducible polynomials, we can use the following command:

```
irred_3.4:
  ▶ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot6\cdot\$
  ▶ ▶ -define\cdot F\cdot -finite\cdot field\cdot -q\cdot4\cdot -end\cdot$
  ▶ ▶ -with\cdot F\cdot -do\cdot$
  ▶ ▶ -finite\cdot field\cdot activity\cdot$
  ▶ ▶ -make\cdot table\cdot of\cdot irreducible\cdot polynomials\cdot 3\cdot -end
  ▶ pdflatex\cdot Irred_q4_d3.tex
  ▶ open\cdot Irred_q4_d3.pdf
```

It produces a table of all irreducible polynomials of degree 3 over $\mathbb{F}_4$. The output is:

```
There are 20 irreducible polynomials of degree 3 over the field $\mathbb{F}_4$:
0 : 1123 : 91
1 : 1031 : 77
2 : 1213 : 103
3 : 1323 : 123
4 : 1322 : 122
5 : 1222 : 106
6 : 1021 : 73
7 : 1101 : 81
8 : 1333 : 127
9 : 1232 : 110
10 : 1113 : 87
```
11 : 1233 : 111
12 : 1301 : 113
13 : 1003 : 67
14 : 1112 : 86
15 : 1002 : 66
16 : 1312 : 118
17 : 1011 : 69
18 : 1132 : 94
19 : 1201 : 97
8.2 Ideals

Let us classify arcs in a projective plane and see which conics we get. The next command classifies the \((5, 2)\)-arcs in \(\text{PG}(2, 11)\):

\[
\text{arcs}_5_2_\text{q11}:
\]

\[
\begin{align*}
\text{\texttt{\$ (ORBITER\_PATH)orbiter.out:-v.4\}} \backslash \\
\text{\texttt{\& \texttt{-define\texttt{F:-finite\_field:-q.11:-end}}\}} \backslash \\
\text{\texttt{\& \texttt{-define\texttt{P:-projective\_space\_2:\texttt{F:-end}}\}} \backslash \\
\text{\texttt{\& \texttt{-with\texttt{P:-do}}\}} \backslash \\
\text{\texttt{\& \texttt{-projective\_space\_activity}}\}} \backslash \\
\text{\texttt{\& \texttt{-classify\_arcs}}\}} \backslash \\
\text{\texttt{\& \texttt{-\texttt{\& -problem\_label\texttt{\_arcs\texttt{}}}} \texttt{\_5\_2\_\texttt{q11}}\}} \backslash \\
\text{\texttt{\& \texttt{\& \texttt{-\texttt{\& -w:-depth\texttt{-5}}\}} \backslash \\
\text{\texttt{\& \texttt{\& \texttt{-report\texttt{-end}}} \backslash \\
\text{\texttt{\& \texttt{\& \texttt{-end}} \backslash \\
\text{\texttt{\& \texttt{-target\_size\texttt{-5}}} \backslash \\
\text{\texttt{\& \texttt{\& \texttt{-d\texttt{-2}}} \backslash \\
\text{\texttt{\& \texttt{\& \texttt{-end}}} \backslash \\
\text{\texttt{\& \texttt{-end}}} \backslash \\
\text{\texttt{\& pdflatex:arcs\texttt{}} \texttt{\_5\_2\_\texttt{q11\_poset.tex}}} \backslash \\
\text{\texttt{\& open-arcs\texttt{}} \texttt{\_5\_2\_\texttt{q11\_poset.pdf}}} \backslash \\
\end{align*}
\]

It finds exactly two isomorphism types of arcs. The representative sets are

\[
\{0, 1, 2, 3, 37\}, \quad \{0, 1, 2, 3, 49\}.
\]

They are stored in the file \texttt{arcs\_5\_2\_q11\_lvl\_5}. Let us now create the ideal in the quadratic component of the polynomial ring in three variables over \(\mathbb{F}_{11}\):

\[
\text{arcs}_5_2\_\text{q11\_ideal}:
\]

\[
\begin{align*}
\text{\texttt{\$ (ORBITER\_PATH)orbiter.out:-v.2\}} \backslash \\
\text{\texttt{\& \texttt{-define\texttt{F:-finite\_field:-q.11:-end}}\}} \backslash \\
\text{\texttt{\& \texttt{-define\texttt{R:-polynomial\_ring}}\}} \backslash \\
\text{\texttt{\& \texttt{-field\texttt{\_F}}\}} \backslash \\
\text{\texttt{\& \texttt{-number\_of\_variables\_3}}} \backslash \\
\text{\texttt{\& \texttt{-homogeneous\_of\_degree\_2}}} \backslash \\
\text{\texttt{\& \texttt{-monomial\_ordering\_lex}}} \backslash \\
\text{\texttt{\& \texttt{-variables\texttt{\\"x\texttt{0, x\texttt{1, x\texttt{2}}\\"}}}}} \backslash \\
\text{\texttt{\& \texttt{-end}}} \backslash \\
\text{\texttt{\& \texttt{-define\texttt{C:-combinatorial\_objects}}\}} \backslash \\
\text{\texttt{\& \texttt{-\texttt{\& -file\texttt{\_of\_points\texttt{\_arcs\texttt{}}}} \texttt{\_5\_2\_\texttt{q11\_l\texttt{\_vl\texttt{\_5}}}}} \backslash \\
\text{\texttt{\& \texttt{\& -end}}} \backslash \\
\text{\texttt{\& \texttt{-with\texttt{C:-do}}} \backslash \\
\text{\texttt{\& \texttt{-combinatorial\_object\_activity}}\}} \backslash \\
\end{align*}
\]
The ideals are generated by

\[ 7x_0x_1 + 5x_0x_2 + 10x_1x_2 \]

and

\[ 4x_0x_1 + 8x_0x_2 + 10x_1x_2, \]

respectively.

Let us consider a smooth cubic surface with 9 lines and 4 Eckardt points. We wish to determine the equation of the object. To do so, we first encode the object as a set of points. Then, we create a ring and compute the ideal:

```
PTS_OF_SURFACE_ORBIT211_Q3_L9_E4="
0,1,2,5,7,8,10,14,9,12,\n15,3,16,37,31,34,20,19,17,32,36,33"
```

We find a two-dimensional ideal. Generators are:

\[ x_0x_0x_1 + 2x_0x_1x_1 + 2x_0x_1x_3 \quad \text{and} \quad 2x_2x_2x_3 + 2x_2x_3x_3. \]
In the next example, we wish to explore the relationship between conics and \((5, 2)\)-arcs. We consider the plane \(\text{PG}(2, 11)\). Instead of classification, we will try random generation this time. Since there are 133 points, we create a number of 5-subsets of a set of size 133. In this case, we create 20 sets at random:

```plaintext
random_k_subsets:
  $\$(\text{ORBITER\_PATH})\text{orbiter.out}\$ -v.4\$
  -create_random_k_subsets:133.5.20
```

The sets are stored in the file `random_k_subsets_n133_k5_nb20.csv`. Now, let’s compute the line type of these subsets, to see which ones are arcs:

```plaintext
line_type_in_PG_2_11:
  $\$(\text{ORBITER\_PATH})\text{orbiter.out}\$ -v.3\$
  -orbirter_path:$(\text{ORBITER\_PATH})\$
  -define_F:finite_field:-q:11:-end:\n  -define_P:projective_space:2:F:-end:\n  -define_C:combinatorial_objects:\n  -file_of_points:random_k_subsets_n133_k5_nb20.csv\n  -end\n```

248
It turns out that the second set is an arc. It is the set \{3, 33, 40, 83, 102\}. We create the conic through these 5 points:

```
random_arc_5_2_q11_ideal:
  $(ORBITER_PATH)orbiter.out:-v.2-
  -define:F:-finite_field:-q.11:-end-
  -define:R:-polynomial_ring-
  -field:F-
  -number_of_variables:3-
  -homogeneous_of_degree:2-
  -monomial_ordering:lex-
  -variables:"x0,x1,x2","x_0,x_1,x_2"-
  -end-
  -define:C:-combinatorial_objects-
  -set_of_points:"3,33,40,83,102"-
  -end-
  -with:C:-do-
  -combinatorial_object_activity-
  -ideal:R-
  -end
```

The ideal is generated by

$$10*x0*x0 + 3*x0*x1 + 8*x0*x2 + 2*x1*x1 + 10*x2*x2.$$  

The conic contains the following 12 points:

\{3, 15, 19, 33, 40, 42, 46, 50, 83, 88, 102, 108\}.
Chapter 9

Applications

9.1 Number Theory

In Table 9.1, some number theoretic commands are shown. For instance,

\[ \text{inverse}\mod a: \]
\[ \text{\textbackslash{}dollar\textbackslash{}(ORBITER\_PATH)orbiter.out-v.2-inverse_mod.18059241.58014043} \]

computes the inverse of 18059241 modulo 58014043.

The Legendre symbol tells us if a number \( a \) is a square modulo an odd prime \( p \). By definition,

\[
\left( \frac{a}{p} \right) = \begin{cases} 
1 & \text{if there exists } r \text{ s.t. } r^2 \equiv a \mod p \\
-1 & \text{if there does not exist } r \text{ s.t. } r^2 \equiv a \mod p \\
0 & \text{if } p \text{ divides } a.
\end{cases}
\]

The Jacobi symbol generalizes the Legendre symbol to allow non-prime bottom arguments. By definition,

\[
\left( \frac{a}{b} \right) = \prod_{i=1}^{k} \left( \frac{a}{r_i} \right)^{e_i},
\]

where

\[ b = \prod_{i=1}^{k} r_i^{e_i} \]

is the prime factorization of \( b \) with pairwise distinct primes \( r_i \). The Jacobi symbol agrees with the Legendre symbol whenever the bottom argument \( b \) is an odd prime. Because there is no ambiguity, the same notation is used for the Jacobi symbol as for the Legendre symbol. Orbiter can compute Jacobi symbols. For instance, the command

\[ \text{jacobi\_a:} \]
\[ \text{\textbackslash{}dollar\textbackslash{}(ORBITER\_PATH)orbiter.out-v.5-jacobi.2221.7817} \]
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-jacobi</td>
<td>a p</td>
<td>Computes the Jacobi symbol ( \left( \frac{a}{p} \right) )</td>
</tr>
<tr>
<td>-sift_smooth</td>
<td>a n primes</td>
<td>Computes all smooth numbers in the interval ([a, a + n - 1]). Smooth means that they factor completely over the list of primes given.</td>
</tr>
<tr>
<td>-random</td>
<td>n fname</td>
<td>Creates ( n ) random numbers and writes them to the csv file \texttt{fname}</td>
</tr>
<tr>
<td>-random_last</td>
<td>n</td>
<td>Creates ( n ) random numbers prints the last one</td>
</tr>
<tr>
<td>-affine_sequence</td>
<td>a b p</td>
<td>Splits the interval ([0, p - 1]) into affine sequences of the form ( x_{n+1} = ax_n + b \mod p )</td>
</tr>
</tbody>
</table>

Table 9.1: Number Theoretic Commands

computes the Jacobi symbol

\[
\left( \frac{2221}{7817} \right)
\]

In the Jacobi symbol, the denominator \( p \) has to be a positive odd integer. This command creates the file \texttt{jacobi_2221_7817.tex} which contains a detailed step-by-step description of the computation. The steps correspond to the basic rules for computing the Jacobi symbol and can be found in many textbooks. After reformatting, the description looks like this:

\[
\left( \frac{2221}{7817} \right) \\
= \left( \frac{7817}{2221} \right) \cdot (-1)^{\frac{2221 - 1}{2}} \frac{7817 - 1}{2} \\
= \left( \frac{7817}{2221} \right) \\
= \left( \frac{1154}{2221} \right) \\
= \left( \frac{2}{2221} \right) \cdot \left( \frac{577}{2221} \right) \\
= (-1)^{\frac{2221^2 - 1}{2}} \cdot \left( \frac{577}{2221} \right) \\
= (-1) \cdot \left( \frac{577}{2221} \right) \\
= (-1) \cdot \left( \frac{2221}{577} \right) \cdot (-1)^{\frac{577 - 1}{2}} \frac{2221 - 1}{2} \\
= (-1) \cdot \left( \frac{2221}{577} \right) \\
= (-1) \cdot \left( \frac{490}{577} \right)
\]
The answer 1 tells us that 2221 is a square modulo 7817. Because 7817 is prime, the Jacobi symbol and the Legendre symbol agree on this input pair. We can use the \texttt{square\_root\_mod} command from Section 3.1 to compute a square root of 2221 modulo 7817 and verify this fact. The command

\begin{verbatim}
sqrt_mod_7817:
  $(ORBITER\_PATH)orbiter.out\:-v\:-square\_root\_mod\:2221\:7817
\end{verbatim}

yields that 7634 is a square root. Indeed,

$$7634^2 \equiv 2221 \mod 7817.$$
### 9.2 Representation Theory

Orbiter has some commands for representations of finite groups. Table 9.2 lists the commands available to classify arcs. The command

```plaintext
representation_on_polynomials_of_degree_3:
> $(ORBITER_PATH)orbiter.out:-v.4:\
>  -define:G:-linear_group:-PGL:4:3:-end:\
>  -with:G:-do:\
>  -group_theoretic_activity:\
>  -representation_on_polynomials:3:\
>  -end.
> $(ORBITER_PATH)orbiter.out:-v.2:\
>  -loop:L:0:9:1:-draw_matrix:\
>  -input_csv_file:PGL:4:3_rep:3_%L.csv:\
>  -box_width:40:-bit_depth:24:-partition:3:20:20:-end:\
>  -end_loop
```

creates \( G = \text{PGL}(4,3) \) and computes the representation on polynomials of degree 3 in 4 variables. The representation has degree 20. The second command produces bitmap drawings for the representing matrices associated with a generating set of the group. Figure 9.1 shows the representing matrices for a generating set of size 9.
Figure 9.1: Representation of PGL(4, 3) on cubic polynomials
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-solovay_strassen</td>
<td>a n</td>
<td>Performs n Solovay / Strassen tests on the number a</td>
</tr>
<tr>
<td>-miller_rabin</td>
<td>a n</td>
<td>Performs n Miller / Rabin tests on the number a</td>
</tr>
<tr>
<td>-fermat</td>
<td>a n</td>
<td>Performs n Fermat tests on the number a</td>
</tr>
<tr>
<td>-find_pseudoprime</td>
<td>a n1 n2 n3</td>
<td>Computes a pseudoprime which survives n1 Fermat tests, n2 Miller Rabin tests, n3 Solovay Strassen tests</td>
</tr>
<tr>
<td>-find_strong_pseudoprime</td>
<td>a n1 n2</td>
<td>Computes a pseudoprime which survives n1 Fermat tests and n2 Miller Rabin tests</td>
</tr>
<tr>
<td>-RSA_encrypt_text</td>
<td>d n b text</td>
<td>Using blocks of b letters at a time, encrypt “text” using RSA with exponent d modulo n</td>
</tr>
<tr>
<td>-RSA</td>
<td>d n list-of-integers</td>
<td>encrypt the given sequence of integers using RSA with exponent d modulo n</td>
</tr>
</tbody>
</table>

Table 9.3: Cryptographic Commands

### 9.3 Cryptography

In Table 9.3, some cryptographic commands are shown. In Table 9.3, some cryptographic commands depending on a finite field are shown. We assume that the field $\mathbb{F}_q$ has been defined. For instance,

**EC_add:**

```
▷ $(ORBITER_PATH)orbiter.out-v.2-
▷ ▷ -define:F:-finite_field:-q11:-end-
▷ ▷ -with:F:-do-
▷ ▷ -finite_field_activity-
▷ ▷ -EC_add:1.3:"1,4":"1,4".-end
```

adds the point $(1, 4)$ on the curve $y^2 = x^3 + x + 3 \mod 11$ to itself. The command

**EC_cyclic_subgroup:**

```
▷ $(ORBITER_PATH)orbiter.out-v.2-
▷ ▷ -define:F:-finite_field:-q11:-end-
▷ ▷ -with:F:-do-
▷ ▷ -finite_field_activity-
▷ ▷ -EC_cyclic_subgroup:1.3:"1,4".-end
```

256
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-EC_add</td>
<td>$a \ b \ i_1 \ i_2$</td>
<td>On the elliptic curve $y^2 \equiv x^3 + ax + b$ in $\mathbb{F}_q$, add the points with indices $i_1$ and $i_2$, each given as a pair $x, y$.</td>
</tr>
<tr>
<td>-EC_points</td>
<td>$a \ b$</td>
<td>Computes all points of the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-EC_multiple_of</td>
<td>$a \ b \ pt \ n$</td>
<td>Computes the $n$ fold multiple of the given point $pt$ on the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-EC_cyclic_subgroup</td>
<td>$a \ b \ pt$</td>
<td>Computes the cyclic subgroup generated by the given point $pt$ on the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-EC_Koblitz_encoding</td>
<td>$a \ b \ s \ pt \ plain$</td>
<td>Computes the Koblitz encoding of “plain” (all caps) on the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$ using the base point $pt$ and the secret exponent $s$.</td>
</tr>
<tr>
<td>-EC_bsgs</td>
<td>$a \ b \ pt \ n \ cipher$</td>
<td>Prepare the baby-step giant-step tables for the ciphertext “cipher” on the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$ using the base point $pt$ of order $n$.</td>
</tr>
<tr>
<td>-EC_bsgs_decode</td>
<td>$a \ b \ pt \ n \ cipher \ round-keys$</td>
<td>Decodes the ciphertext “cipher” on the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$ using the base point $pt$ of order $n$ and the round keys “keys”.</td>
</tr>
<tr>
<td>-EC_discrete_log</td>
<td>$a \ b \ pt \ base-pt$</td>
<td>Computes the elliptic curve discrete log analogue of $pt$ with respect to base-pt on the elliptic curve $y^2 \equiv x^3 + ax + b$ over $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-NTRU_encrypt</td>
<td>$N \ p \ H \ R \ M$</td>
<td>NTRU encryption for the message $M(X)$ using the public key $H(X)$ and one-time-key $R(X)$.</td>
</tr>
<tr>
<td>-polynomial_center_lift</td>
<td>$A(X)$</td>
<td>Compute the center lift mod $q$ for the coefficients of $A$.</td>
</tr>
<tr>
<td>-polynomial_reduce_mod_p</td>
<td>$p \ A(X)$</td>
<td>Reduce the coefficients of the polynomial $A$ modulo $p$.</td>
</tr>
</tbody>
</table>

Table 9.4: Finite Field Activities related to Cryptography
Figure 9.2: The elliptic curve $y^2 = x^3 + 5x + 7 \mod 199$

computes the cyclic subgroup generated by the point $(1, 4)$ on the curve $y^2 = x^3 + x + 3 \mod 11$. The command

```
EC_points_199: 258
$\text{(ORBITER\_PATH)orbiter.out\,-v\,-2\,-}\$
$\,\,\,\,-\text{define}\,\,-\text{finite\_field}\,-\text{q\,199}\,-\text{end}\,\$
$\,\,\,\,-\text{with}\,\,-\text{do}\,\$
$\,\,\,\,-\text{finite\_field\_activity}\,\$
$\,\,\,\,-\text{EC\_points}\,\,\text{"EC\_5\_7\_q199"}\,\,5\,\,7\,-\text{end}\,\$
$\,\,\,\,$\text{(ORBITER\_PATH)orbiter.out\,-v\,-2\,-}\$
$\,\,\,\,-\text{draw\_matrix}\,-\text{input\_csv\_file}\,-\text{EC\_5\_7\_q199\_points\_xy\_csv}\,\$
$\,\,\,\,-\text{box\_width}\,10\,-\text{bit\_depth}\,24\,\$
$\,\,\,\,-\text{partition}\,2\,\,199\,-199\,-\text{end}\,$
```

computes all points on the curve $y^2 = x^3 + 5x + 7 \mod 199$ and produces a bitmap drawing of the points in the affine plane shown in Figure 9.2. Both the $x$-axis and the $y$-axis are indexed by the field elements from 0 to 198.

The command
The command

```
EC_Koblitz_encoding:
  $(ORBITER_PATH)orbiter.out:-v.6.-seed.17.\n  -define:F:-finite_field:-q.199:-end\n  -with:F:-do\n  -finite_field_activity\n  -EC_Koblitz_encoding:5.7.67."147,164"."DEADBEEF".\n  -end
```

```
encode the message “DEADBEEF” on the curve $y^2 = x^3 + 5x + 7 \mod 199$ using the base point (147, 164) and the secret key 67. The $i$th input character is encoded as two points $(R_i, T_i)$ on the curve using the Elgamal scheme. A random round key is generated for each plaintext symbol. As seen in this example, the `-seed` command can be used to seed the random number generator with an arbitrary integer (here 17).
```

The command

```
EC_bsgs:
  $(ORBITER_PATH)orbiter.out:-v.2.\n  -define:F:-finite_field:-q.199:-end\n  -with:F:-do\n  -finite_field_activity\n  -EC_bsgs:5.7."147,164".212.\n  "172,158,45,195,50,22,10,103,55,33,50,22,145,105,31,74,73,155,67,60,25,6".\n  -end
```

```
performs a baby-step-giant-step brute force attack on the ciphertext sequence

$$R_i = (172, 158), (45, 195), (50, 22), (10, 103), (55, 33),$$
$$ (50, 22), (145, 105), (31, 74), (73, 155), (67, 60), (25, 6),$$

using the base point (147, 164) on the curve $y^2 = x^3 + 5x + 7 \mod 199$, assuming a group order of 212. The command

```
EC_bsgs_decode:
  $(ORBITER_PATH)orbiter.out:-v.2.\n  -define:F:-finite_field:-q.199:-end\n  -with:F:-do\n  -finite_field_activity\n  -EC_bsgs_decode:5.7."129,176".212.\n  "127,188,51,141,85,29,106,90,41,105,179,71,171,2,16,197,183,72,27,129,37,10".\n  "50,179,169,13,153,169,115,116,188,110,176".\n  -end
```

259
decodes the ciphertext sequence

\[ T_i = (127, 188), (51, 141), (85, 29), (106, 90), (41, 105), (179, 71), \\
(171, 2), (16, 197), (183, 72), (27, 129), (37, 10), \]

assuming round keys

\[ k_i = 50, 179, 169, 13, 153, 169, 115, 116, 188, 110, 176, \]

using the base point \((147, 164)\) on the curve \(y^2 = x^3 + 5x + 7 \mod 199\), and assuming a group order of 212.

The next sequence of examples discusses the NTRU cryptosystem (cf. Example 7.53 in [34]). In the example, we choose the parameters of the cryptosystem to be \((N, p, q, d) = (7, 41, 3, 2)\). Orbiter uses the following convention for polynomials over a finite field \(F_q\): The coefficients of \(A(X) = a_0 + a_1X + \cdots + a_dX^d\) are listed as a sequence, starting with the constant term and ending with the leading coefficient. The cryptosystem requires coefficients \(a_i\) in the range \(-\frac{p}{2} \leq a_i \leq \frac{p}{2}\). So, in an extension to the convention for field elements in \(F_q\), Orbiter allows negative coefficients as well. The assumption is that \(q\) is prime and negative coefficients are considered modulo \(q\). In the example, Alice picks the private polynomials \(f(x) = x^6 - x^4 + x^3 + x^2 - 1\) (with \(d + 1\) coefficients equal to plus one and \(d\) coefficients equal to minus one) and \(g(x) = x^6 + x^4 - x^2 - x\) with \(d\) coefficients plus one and \(d\) coefficients minus one. We also need the polynomial \(x^N - 1\). The makefile commands

\[
\begin{align*}
\text{NTRU} \_N &= 7 \\
\text{NTRU} \_P &= 3 \\
\text{NTRU} \_Q &= 41 \\
\text{NTRU} \_D &= 2 \\
\text{NTRUE} \_XN1 &= "-1,0,0,0,0,1," \\
# \text{D}:+1+\text{ones} \text{and} \text{D}:-\text{minus} \text{ones} \\
\text{ALICE} \_PRIVATE \_F &= "-1,0,1,1,-1,0,1" \\
# \text{D}:+1+\text{ones} \text{and} \text{D}:-\text{minus} \text{ones} \\
\text{ALICE} \_PRIVATE \_G &= "0,-1,-1,0,1,0,1" \\
\end{align*}
\]

are used to set up the appropriate variables according to these choices.

Regarding the NTRU set-up, Alice needs to compute her private keys \(F_p(x)\) and \(F_q(x)\). These two polynomials are defined as follows:

1. \(F_p(x)\) is the inverse of \(f(x)\) in \(F_p[x]/(x^n - 1)\),

2. \(F_q(x)\) the inverse of \(f(x)\) in \(F_q[x]/(x^n - 1)\).

To this end, we can use the `extended_gcd_for_polynomials` command from Table 9.1. The following two makefile commands do the job:
NTRU_Alice1:
▷ $(ORBITER\_PATH)\text{orbiter.out}\,-v\,2\,$
▷ ▷ -define\:\text{F}\,-\text{finite\_field}\,-q\,$(NTRU\_Q)\,-\text{end}\,$
▷ ▷ -with\:\text{F}\,-\text{do}\,$
▷ ▷ -\text{finite\_field\_activity}\,$
▷ ▷ -\text{extended\_gcd\_for\_polynomials}\,$(NTRUE\_XN1)\,$(ALICE\_PRIVATE\_F)\,-\text{end}\,$

\#F_{q}(x)\,=\,8X^{6}\,\cdot\,26X^{5}\,\cdot\,31X^{4}\,\cdot\,21X^{3}\,\cdot\,40X^{2}\,\cdot\,2X\,\cdot\,37
ALICE\_PRIVATE\_FQ=37,2,40,21,31,26,8

NTRU_Alice2:
▷ $(ORBITER\_PATH)\text{orbiter.out}\,-v\,2\,$
▷ ▷ -define\:\text{F}\,-\text{finite\_field}\,-q\,$(NTRU\_P)\,-\text{end}\,$
▷ ▷ -with\:\text{F}\,-\text{do}\,$
▷ ▷ -\text{finite\_field\_activity}\,$
▷ ▷ -\text{extended\_gcd\_for\_polynomials}\,$(NTRUE\_XN1)\,$(ALICE\_PRIVATE\_F)\,-\text{end}\,$

\#F_{p}(x)\,=\,X^{6}\,\cdot\,2X^{5}\,\cdot\,X^{3}\,\cdot\,X^{2}\,\cdot\,X\,\cdot\,1
ALICE\_PRIVATE\_FP=1,1,1,1,0,2,1

The resulting polynomials (indicated as comments by means of the \# symbol) are again encoded as makefile variables. There is a chance that the polynomial \(f(x)\) does not have an inverse in either \(\mathbb{F}_{p}[x]\) or in \(\mathbb{F}_{q}[x]\). In that case, Alice simply chooses a different polynomial \(f(x)\) and tries again. Alice can now compute her public key:

NTRU_Alice\_public\_key:
▷ $(ORBITER\_PATH)\text{orbiter.out}\,-v\,2\,$
▷ ▷ -define\:\text{F}\,-\text{finite\_field}\,-q\,$(NTRU\_Q)\,-\text{end}\,$
▷ ▷ -with\:\text{F}\,-\text{do}\,$
▷ ▷ -\text{finite\_field\_activity}\,$
▷ ▷ -\text{polynomial\_mult\_mod}\,$(ALICE\_PRIVATE\_F)\,$
▷ ▷ ▷ -$(ALICE\_PRIVATE\_G)\,$$(NTRUE\_XN1)\,$
▷ ▷ ▷ -\text{end}\,$

\#C(X)=20X^{6}\,\cdot\,40X^{5}\,\cdot\,2X^{4}\,\cdot\,38X^{3}\,\cdot\,8X^{2}\,\cdot\,26X\,\cdot\,30
ALICE\_PUBLIC\_KEY=30,26,8,38,2,40,20

The public key is assigned to the makefile variable \textit{ALICE\_PUBLIC\_KEY}. Now, Bob chooses his message to Alice and his one-time-key. The message must be the center lift of a polynomial in \(\mathbb{F}_{p}[x]\). The round-key must have exactly \(d\) coefficients one and \(d\) coefficients -1 (rest zeroes).

\textit{BOB\_MESSAGE}=1,-1,1,1,0,-1
The encryption proceeds using the `NTRU_encrypt` command, and the result is stored in the makefile variable `BOB_ENCRYPT`:

```
NTRU_encrypt:
  $(ORBITER_PATH)orbiter.out -v.2`
  define:F:-finite_field-q$(NTRU_Q)-end
  with:F-do
  finite_field_activity
  NTRU_encrypt $(NTRU_N) $(NTRU_P) $(ALICE_PUBLIC_KEY) $(BOB_ONE_TIME_KEY) $(BOB_MESSAGE) -end

#E(X)=.31X^6+.19X^5+.4X^4+.2X^3+.40X^2+.3X+.25
BOB_ENCRYPT="25,3,40,2,4,19,31"
```

Decryption is done in five steps.

```
NTRU_decrypt1:
  $(ORBITER_PATH)orbiter.out -v.2`
  define:F:-finite_field-q$(NTRU_Q)-end
  with:F-do
  finite_field_activity
  polynomial_mult_mod $(ALICE_PRIVATE_F) $(BOB_ENCRYPT) $(NTRUE_XN1) -end

#C(X)=X^6+.10X^5+.33X^4+.40X^3+.40X^2+.X+.40
ALICE_C1="40,1,40,40,33,10,1"
```

```
NTRU_decrypt2:
  $(ORBITER_PATH)orbiter.out -v.2`
  define:F:-finite_field-q$(NTRU_Q)-end
  with:F-do
  finite_field_activity
  polynomial_center_lift $(ALICE_C1) -end

#A(X)=X^6+.10X^5-.8X^4-.X^3-X^2+.X-.1
ALICE_C2="-1,1,-1,-8,10,1"
```

```
NTRU_decrypt3:
  $(ORBITER_PATH)orbiter.out -v.2`
  define:F:-finite_field-q$(NTRU_P)-end
  with:F-do
```

BOB_ONE_TIME_KEY="-1,1,0,0,0,-1,1"
Decryption produces Bob's message to Alice.
Chapter 10
Coding Theory

10.1 Introduction
In Table 10.1, global coding theoretic commands of Orbiter are shown. The commands

```
Hamming_space_4_2_distance_matrix:
\> $(ORBITER_PATH)orbiter.out--Hamming_space_distance_matrix 4 2
\> $(ORBITER_PATH)orbiter.out--v 2--draw_matrix
\> \> -input_csv_file Hamming_n4_q2.csv
\> \> -box_width 20--bit_depth 24--partition 4 16 16--end
\> open Hamming_n4_q2_draw.bmp
```

create the csv-file `Hamming_n4_q2.csv` and produce the bitmap file

```
Hamming_n4_q2_draw.bmp
```

shown in Figure 10.1. Table 10.2 lists coding theoretic activities in Orbiter.

The following command creates the $[5,2]_2$ code whose codewords are $\{0,7,25,30\}$:

```
CODE_5_2_3_CODEWORDS="0,7,25,30"
```

code_5_2_3_diagram:
```
\> $(ORBITER_PATH)orbiter.out--v 2--code_diagram "code_5_2_3"
\> \> $(CODE_5_2_3_CODEWORDS)5--metric_balls 1
\> $(ORBITER_PATH)orbiter.out--v 2--draw_matrix
\> \> -input_csv_file code_5_2_3_diagram_01_5_4.csv
\> \> -box_width 25--bit_depth 24--partition 4 8 4--end
```

The Hamming graph $H(5,2)$ can be created with the following command:

```
Hamming_5_2_graph:
\> $(ORBITER_PATH)orbiter.out--v 2
```

265
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Hamming_graph</td>
<td>n q</td>
<td>Creates the distance matrix of the Hamming graph $H(n,q)$. The vertices are the elements of $\mathbb{F}_q^n$, and the $i,j$-entry is the distance between the vectors whose affine ranks are $i$ and $j$, respectively. The matrix is written as csv-file.</td>
</tr>
<tr>
<td>-general_code_binary</td>
<td>n R</td>
<td>Creates the binary code of length $n$ containing the elements corresponding to the integers in the list $R$ under the binary representation.</td>
</tr>
<tr>
<td>-linear_code_through_basis</td>
<td>n R</td>
<td>Creates the binary linear code of length $n$ generated by the elements corresponding to the integers in the list $R$ under the binary representation.</td>
</tr>
<tr>
<td>-long_code</td>
<td>n k r₁ ... rₖ</td>
<td>Creates the binary code of length $n$ and dimension $k$ whose generators are given as $r_1, \ldots, r_k$.</td>
</tr>
<tr>
<td>-make_macwilliams_system</td>
<td>q n k</td>
<td>Creates the MacWilliams system for a linear $[n,k]_q$-code.</td>
</tr>
<tr>
<td>-table_of_bounds</td>
<td>N q</td>
<td>Compute Singleton, Hamming, Plotkin, Griesmer upper bounds on $d$ for a $[n,k]_q$ code for all $n \leq N$ and all $k \leq n$. The results are written to a csv file.</td>
</tr>
</tbody>
</table>

Table 10.1: Global Coding Theoretic Commands
Figure 10.1: The color-coded distance matrix of the Hamming graph $H(4,2)$

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-weight Enumerator</td>
<td>$m \ n \ L$</td>
<td>Compute the complete weight enumerator of the linear code generated by the $m \times n$ matrix $L$</td>
</tr>
<tr>
<td>-field Reduction</td>
<td>$q_0 \ m \ n \ L$</td>
<td>Perform field reduction. The input is a $m \times n$ generator matrix $L$ over the field $\mathbb{F}_q$. The output is the $sm \times sn$ generator matrix of the code obtained by field reduction. The code is defined over the field of order $q_0$, which must be a subfield of $\mathbb{F}_q$, with $q_0^s = q$. A latex report is written.</td>
</tr>
</tbody>
</table>

Table 10.2: Coding Theoretic Activities
Figure 10.2: Drawing of the Hamming graph $H(5, 2)$

Using the unix dot program, this command sequence creates the drawing of $H(5, 2)$ shown in Figure 10.2.

```bash
-define G:graph:Hamming 5 2: end
-define G:with G:do:
-define graph_theoretic_activity:export_csv:end
-define G:with G:do:
-define graph_theoretic_activity:export_graphviz:end
-define G:with G:do:
-define graph_theoretic_activity:save:end
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix
-define input_csv_file:Hamming 5 2.csv
-define -box_width 8 -bit_depth 24 -partition 4 32 32 -end
dot -Tpng Hamming 5 2.gv -> Hamming 5 2.png
```
10.2 Hamming Codes

The Hamming code is the dual of the simplex code. The simplex code has a generator matrix whose columns are the coordinate vectors of the points of \( \text{PG}(2, 2) \). To compute the dual, we need to compute the nullspace of this matrix. The following command does that:

\[
\text{SIMPLEX\_CODE\_GENERATOR} = "\
1,0,1,0,1,0,1,\
0,1,1,0,0,1,1,\
0,0,0,1,1,1,1"
\]

Hamming generator:
\[
\begin{array}{c}
\text{\LaTeX}\\
\$\text{ORBITER\_PATH}\text{orbiter.out}\cdot\text{v.2}\cdot\\
\text{-define}\cdot\text{F}\cdot\text{-finite\_field}\cdot\text{-q}\cdot2\cdot\text{-end}\cdot\\
\text{-define}\cdot\text{v}\cdot\text{-vector}\cdot\text{-field}\cdot\text{F}\cdot\text{-format}\cdot3\cdot\\
\text{-dense}\cdot\text{$\{$SIMPLEX\_CODE\_GENERATOR$\}$$}\cdot\\
\text{-end}\cdot\\
\text{-with}\cdot\text{F}\cdot\text{-do}\cdot\\
\text{-finite\_field\_activity}\cdot\\
\text{-nullspace}\cdot\text{v}\cdot\\
\text{-end}
\end{array}
\]
\[
pdflatex\cdot\text{nullspace_3_7.tex}
\]
\[
\text{open-nullspace_3_7.pdf}
\]

This produces the following output:

\[
\text{Input matrix:}
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]
\[
\text{RREF:}
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]
\[
\text{Basis for Perp:}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]
Suppose we want to look at the codewords of the Hamming code in the Hamming space. The following command will produce Figure 10.3.

```
Hamming_code_words:
  $(ORBITER_PATH)orbiter.out -v 2
  -define v vector -dense $(HAMMING_CODE_ROWS_IN_BINARY_RANKS) -end
  -linear_code_through_basis 7 v
  pdflatex code_n7_k4_q2.tex
  open_code_n7_k4_q2.pdf
```

Suppose we want to compute the weight enumerator of the Hamming code. We use the following command:

```
HAMMING_CODE_GENERATOR="\n1,0,0,0,1,1,1,1,1,1,1,0,1,0,0,0,0,0,1,0,0,0,0,1,1,1,1,0,0,1,1,1,0,0,0,1,1,1,1"\n
Hamming_weight Enumerator:
  $(ORBITER_PATH)orbiter.out -v 2
  -define F finite_field -q 2 -end
  -define v vector -field F -format 4
  -dense $(HAMMING_CODE_GENERATOR)
  -end
```
We find that the weight enumerator is 

\[(1,0,0,7,7,0,0,1)\].

Suppose we want to establish the MacWilliams relations for the Hamming code. The following command creates the matrix of Kravtchuck numbers:

```bash
Hamming_code_macwilliams:
$(ORBITER_PATH)orbiter.out -v 2 -make_macwilliams_system 7 4 2
pdflatex MacWilliams_n7_k4_q2.tex
open MacWilliams_n7_k4_q2.pdf
```

This produces the following output:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
7 & 5 & 3 & 1 & -1 & -3 & -5 & -7 \\
21 & 9 & 1 & -3 & -3 & 1 & 9 & 21 \\
35 & 5 & -5 & -3 & 3 & 5 & -5 & -35 \\
35 & -5 & -5 & 3 & 3 & -5 & -5 & 35 \\
21 & -9 & 1 & 3 & -3 & -1 & 9 & -21 \\
7 & -5 & 3 & -1 & -1 & 3 & -5 & 7 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1
\end{bmatrix}
\]

The Hamming code is cyclic. To see this, we need to consider the action of the Singer cycle on the set of points of PG(2, 2). The following command creates the Singer cycle:

```bash
Hamming_singer:
$(ORBITER_PATH)orbiter.out -v 3 \
define G linear group PGL 3 2 -singer 1 -end \
-with G -do \
-group_theoretic_activity \
-report \
-orbits_on_points \
-end 
pdflatex PGL_3_2_Singer_3_2_1_report.tex 
opend PGL_3_2_Singer_3_2_1_report.pdf
```

This produces the following output:
Strong generators for a group of order 7:

$$\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 1
\end{bmatrix}.$$ 

Basic Orbit 0

0
1
2
5
3
4
6

Basic orbit 0 has size 7
0, 1, 2, 3, 4, 5, 6

From this, we know how to rearrange the points of PG(2, 2) to exhibit the cyclic structure. We issue the following command to recreate the Hamming code:

```
SIMPLEX_CODE_GENMA_CYCLIC="\n1,0,0,1,1,1,0,\n0,1,0,0,1,1,1,\n0,0,1,1,1,0,1"
```

Hamming cyclic generator:

- $(ORBITER_PATH)orbiter.out -v 2 -
- -define F: -finite_field: -q 2 -
- -define v: -vector: -format 3: -field F: -
- -dense $(SIMPLEX_CODE_GENMA_CYCLIC): -
- -end: -
- -with F: -do -finite_field_activity: -
- -nullspace v: -
- -end
- pdflatex nullspace_3_7.tex
This produces the following output:

| Input matrix: | \[
\begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\end{bmatrix}
\] |
|--------------|--------------------------------------------------|
| RREF:        | \[
\begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\end{bmatrix}
\] |
| Basis for Perp: | \[
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\end{bmatrix}
\] |
10.3 Golay Codes

The Golay code of length 23 is a perfect code of dimension 12 and minimum distance 7. The metric balls of radius three centered around codewords cover the whole Hamming space. We can create the code by listing the columns of a generator matrix in Orbiter ranks of points in PG(11,2). The following makefile variable does that:

```
GOLAY_23_COLUMN_RANKS_PROJECTIVELY="0,1,2,3,4,5,6,7,\n8,9,10,11,132,913,1460,1750,1898,2518,2787,2874,\n3320,3357,3662"
```

Suppose we want to list the code words. The following command can be used:

```
Golay23_code_words:
	$(ORBITER_PATH)orbiter.out -v 2 \\
	define v-vector dense $(GOLAY_23_COLUMN_RANKS_PROJECTIVELY) -end \\
	-linear_code_through_columns_of_parity_check_projectively 12:v \\
	pdflatex code_n23_k12_q2.tex \\
	open-code_n23_k12_q2.pdf
```
10.4 CRC Codes

A CRC code can be used to detect communication errors. It is a cyclic code, and hence generated by a polynomial over a finite field. The message is encoded as a string, which is then thought of as a polynomial, called the information polynomial. Assume that the check polynomial has degree \( d \). The information polynomial is then divided by the check polynomial. The remainder is added to the information polynomial multiplied by \( X^d \). This is the codeword, which is sent.

Here is an example. We consider a short string of English text and encode it with 5 bits per character. This is done using the `-encode_text_5bits` command. The encoded text is stored in a csv file, which we decide to call `text.csv`.

```
encode_text_5bits:
  $(ORBITER_PATH)orbiter.out-encode_text_5bits-
  "Hithere"."text.csv"
  $(ORBITER_PATH)orbiter.out-v.2-
  -define:F:-finite_field:-q:2:-end-
  -with:F:-do-
  -finite_field_activity-
  -polynomial_division_from_file-
  text.csv.13-end
  pdflatex-polynomial_division_file_13.tex
  open-polynomial_division_file_13.pdf
```

We decide to pick the binary polynomial \( 13 = X^3 + X^2 + 1 \). We divide the information polynomial by the check polynomial:

```
encode_text_5bits_check:
  $(ORBITER_PATH)orbiter.out-v.2-
  -define:F:-finite_field:-q:2:-end-
  -with:F:-do-
  -finite_field_activity-
  -polynomial_division_from_file-text_with_1error.csv.13-end
  pdflatex-polynomial_division_file_13.tex
  open-polynomial_division_file_13.pdf
```

This creates the following output:

```
text.csv / 13 =
1010110100110101010111000010111110 / 1101 =
110111100011101111110100000101
==================================
1101 | 10101101100110101010111000010111100
```

275
The remainder after division by the check polynomial is 5, or the polynomial $X^2 + 1$, or the bit-sequence 101.

The following command investigates all 1-bit errors, to see which of them can be detected using the given CRC-polynomial:

```
encode_text_5bits_1error:
  ▶ $(ORBITER_PATH)orbiter.out -encode_text_5bits:
  ▶ ▶ "Hithere"."text.csv"
  ▶ ▶ $(ORBITER_PATH)orbiter.out -v.2:
  ▶ ▶ -define:F:-finite_field:-q:2:-end:"
  ▶ ▶ -with:F:-do:
  ▶ ▶ -finite_field_activity:
  ▶ ▶ -polynomial_division_from_file_all_k_bit_error_patterns:
  ▶ ▶ ▶ text.csv.13.1-end
  ▶ ▶ pdflatex:polynomial_division_file_all_1_error_patterns.13.tex
  ▶ open:polynomial_division_file_all_1_error_patterns.13.pdf
```

The following output is created:

```
  0: 01010110100110101010101010111000010111100
  1: 010101101001101010101010111000010111110: 111: 7: X^{-2} + X + 1
  2: 01010110100110101011101011111000010111100: 001: 1: 1
  3: 01010110100110101011110110010111100: 000: 0: 0
  4: 01010110100110101011110111010111100: 000: 0: 0
  5: 01010110100110101011110111010111100: 010: 2: X
  6: 0101011010011010101111011101111100: 011: 3: X + 1
  7: 0101011010011010101111011101111100: 011: 4: X^{-2}
  8: 0101011010011010101111011101111100: 100: 7: X^{-2} + X + 1
  9: 0101011010011010101111011101111100: 001: 1: 1
 10: 0101011010011010101111011101111100: 000: 0: 0
 11: 0101011010011010101111011101111100: 010: 2: X
```

\[ \frac{1}{13} = 1841528453 \text{ Remainder 5} \]
It shows that 5 single bit errors are undetected.

The following command performs an exhaustive search over all binary CRC polynomials of degree \( k = 10 \) which can detect every error pattern of Hamming weight at most \( t = 3 \) in messages of length \( n = 128 \).

**CRC.3.128.10:**

```
$ (ORBITER_PATH) orbiter.out -v -1
```

The program finds 244 polynomials. The execution time is about 1 minute.
10.5 Reed-Muller Codes

The following command creates the generator matrix of the first order Reed-Muller code in 3 dimensions, RM$_{3,1}$. The codewords are listed as well.

```
REED_MULLER_3_1_BASIS_IN_BINARY="255,170,204,240"

RM_3_1_code_words:
  $ (ORBITER_PATH)orbiter.out-\-v.2\-
  \-define.v.-vector.-dense.$ (REED_MULLER_3_1_BASIS_IN_BINARY).-end\-
  \-linear_code_through_basis.8.v
  pdflatex-code_n8_k4_q2.tex
  open-code_n8_k4_q2.pdf

#Codewords: (0,255,170,85,204,51,102,153,240,15,90,165,60,195,150,105)
```

The output is shown in Figure 10.4.

The following command produces a diagram of the characteristic function of the Reed Muller code in the Hamming space.

```
RM_3_1_Hamming_space_diagram:
  $ (ORBITER_PATH)orbiter.out-\-v.2-\-code_diagram."RM_3.1".\-
  \-define.v.-metric.balls.1
  pdflatex-code_n8_k4_q2.tex
  open-code_n8_k4_q2.pdf
```
Figure 10.5: Boolean function representation of RM\(_{3,1}\) in \(H(8,2)\)

\[ \text{\textbf{produces a representation of the code as boolean function in the Hamming space } } H(8,2), \text{ shown in Figure 10.5. The different codewords are given different colors.} \]
10.6 BCH Codes

Let $\beta$ be an $n$-th root of unity over $\mathbb{F}_q$. The minimum polynomial of $\beta$ over $\mathbb{F}_q$ is denoted as $m_{\beta, \mathbb{F}_q}$. The BCH code of length $n$ and designed distance $d$ is the cyclic code with generator polynomial

$$\text{lcm}(m_{\beta, \mathbb{F}_q}, m_{\beta^2, \mathbb{F}_q}, \ldots, m_{\beta^{d-1}, \mathbb{F}_q}).$$

To create the polynomial $m_{\beta^a, \mathbb{F}_q}$, we consider the $q$-cyclotomic set of $a$ modulo $n$, which is

$$\{a q^i \mod n \mid i \in \mathbb{Z}\}.$$

Suppose we want to make a BCH-code of length 21 over $\mathbb{F}_8$. In Section 3.3, we considered the $q$-cyclotomic sets modulo 21 for $q = 8$. Let us produce a pictorial representation. Omitting the singletons, a transversal is given by the sets containing 1, 2, 4, 5, 7, 10, 13. For this reason, we issue the command

```
draw cyclotomic_mod_21_q8:
  $(\text{ORBITER\_PATH})\text{orbiter.out} -v -2 -\text{-draw\_options} -\text{-radius} 100 -\text{-line\_width} 1.0 -\text{-embedded} -\text{-radius} 100 -\text{-line\_width} 1.0 -\text{-embedded} -\text{-draw\_mod\_n} -\text{-n} 21 -\text{-file} mod_21\_cyclotomic -\text{-cyclo\_tomic\_sets} 8 "1,2,4,5,7,10,13" -\text{-end}
  \text{pdflatex} mod_21\_cyclotomic\_draw.tex
  \text{open} mod_21\_cyclotomic\_draw.pdf
```

The output is shown in Figure 10.6. We will try BCH-codes with minimum distances 3, 5 and 7. Here is distance 3:

```
F_8_BCH_code_d3:
  $(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 -\text{-define\_F\_finite\_field\_q} 8 -\text{-override\_polynomial} 11 -\text{-end}
  -\text{-with\_F\_do\_finite\_field\_activity} -\text{-make\_BCH\_code} 21\_3 -\text{-end}
  \text{pdflatex} BCH\_codes\_q8\_n21\_d3.tex
  \text{open} BCH\_codes\_q8\_n21\_d3.pdf
```

The code is described in a latex output file:

```
\text{BCH-code:}
\quad n = 21, \quad k = 17, \quad d_0 = 3, \quad q = 8, \\
g(x) = m_1 m_2 = X^4 + 4X^3 + 4X^2 + 3X + 4
\quad \text{Chosen cyclotomic sets:}
\quad \{ 1, 8 \}
\quad \{ 2, 16 \}
```
The generator polynomial has degree 4

- dense "4,3,4,4,1"

- sparse "4,0,3,1,4,2,4,3,1,4"
The generator matrix is:

\[
\begin{bmatrix}
4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 4 & 4 & 1 & 0 & 0
\end{bmatrix}
\]

And now for \( d = 5 \):

\begin{verbatim}
F_8_BCH_code_d5:
  ➤ $(ORBITER_PATH)orbiter.out-\v.3\$
  ➤ -define F\-finite_field-q.8\-override_polynomial.11\-end\$
  ➤ -with F\-do\-finite_field_activity\-make_BCH_code.21.5\-end
  ➤ pdflatex.BCH_codes_q8_n21_d5.tex
  ➤ open.BCH_codes_q8_n21_d5.pdf
\end{verbatim}

The output file is:

\begin{verbatim}
BCH-code:
n = 21, k = 14, d_0 = 5, q = 8,
g(x) = m_1m_2m_3m_4 = X^7 + 3X^6 + 3X^5 + 2X^4 + X^3 + 2X^2 + X + 2
Chosen cyclotomic sets:
{ 1, 8 }
{ 2, 16 }
{ 3 }
{ 4, 11 }
The generator polynomial has degree 7
\end{verbatim}
The generator matrix is:

\[
\begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 \\
\end{bmatrix}
\]

Finally, \( d = 7 \):

\texttt{F\_8\_BCH\_code\_d7:}

\begin{itemize}
  \item \texttt{\$(ORBITER\_PATH)\textbackslash orbiter.out-\_v.3\backslash}
  \item \texttt{\quad -define:\textbackslash F:\textbackslash -finite\_field\_q.8\_\_override\_polynomial.11\_\_end\backslash}
  \item \texttt{\quad -with:\textbackslash F:\_\_do:\textbackslash finite\_field\_activity\_\_\_make\_BCH\_code\_21.7\_\_end}
  \item \texttt{pdflatex\_BCH\_codes\_q8\_n21\_d7.tex}
  \item \texttt{open\_BCH\_codes\_q8\_n21\_d7.pdf}
\end{itemize}

The output file is:

\begin{itemize}
  \item \textbf{BCH-code:}
  \item \textit{n = 21, k = 11, d = 7, q = 8,}\n  \item \textit{g(x) = m_1m_2m_3m_4m_5m_6 = X^{10} + X^9 + 2X^8 + 5X^7 + 2X^6 + 4X^4 + 6X^3 + 5X^2 + 6X + 6}\n  \item \textit{Chosen cyclotomic sets:}\n  \item \{ 1, 8 \}\n  \item \{ 2, 16 \}\n  \item \{ 3 \}\n\end{itemize}
The generator polynomial has degree 10

- dense "6,6,5,6,4,0,2,5,2,1,1"

- sparse "6,0,6,1,5,2,6,3,4,4,2,6,5,7,2,8,1,9,1,10"

The generator matrix is:

$$
\begin{bmatrix}
6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 5 & 6 & 4 & 0 & 2 & 5 & 2 & 1 & 1 & 0 \\
\end{bmatrix}
$$

As a larger example, let us consider the 2-cyclotomic sets of 2 and 3 modulo 255. The following command produces a graphical representation on a circle (similar to the unit circle in complex analysis). The 255-th roots of unity are placed in the appropriate position.

```bash
draw_mod_255_cyclotomic_1_and_3:
  $ (ORBITER_PATH)orbiter.out -v 2 -
  -draw.options -nodes_empty -radius 10 -
  -line_width 0.4 -embedded -end -
  -draw_mod.n -n 255 -file mod_255_cyclotomic_1_and_3 -
  -cyclotomic_sets 2 "1,3" -end
  pdflatex mod_255_cyclotomic_1_and_3 draw.tex
  open mod_255_cyclotomic_1_and_3 draw.pdf
```

The drawing is shown in Figure 10.7.

Suppose we want to make a BCH-code over $\mathbb{F}_{256}$. In order to keep the degree of the generator polynomial low, we try a quadratic field extension. This way, each cyclotomic set has size 285.
either 1 or 2. Since

\[ 256^2 - 1 = (256 + 1)(256 - 1) = 257 \cdot 3 \cdot 5 \cdot 17, \]

we can consider a code of length \( n = 771 = 257 \cdot 3 \). The following command computes the 256-cyclotomic cosets modulo 771:

F256_roots_771:

```
$ (ORBITER_PATH)orbiter.out -v 3 \n  \define F: finite_field -q 256 -end \n  \with F: do finite_field_activity -nth_roots 771 -end
```

The next command creates a BCH-code of length 771 over \( \mathbb{F}_{256} \) with minimum distance at least 16:

F256_BCH_code_d16:

```
$ (ORBITER_PATH)orbiter.out -v 3 \n  \define F: finite_field -q 256 -end \n  \with F: do finite_field_activity -make_BCH_code 771 16 -end
```

The generator polynomial is printed in two ways, sparse and dense. The notion of sparse and dense agrees with that of Section 2.7. Dense means that the coefficient vector of the polynomial is listed in full. Sparse means that only the nonzero terms are listed as pairs, the nonzero coefficient and the index of the term. The coefficient vector determines the generator polynomial \( g(X) \in \mathbb{F}_{256}[X] \) of the BCH code of length 771 over \( \mathbb{F}_{256} \). Next, we test if \( g(x) \) divides \( X^{771} - 1 \), as it should:
This confirms that the remainder after dividing $X^{771} - 1$ by $g(X)$ is indeed zero.
10.7 Reed-Solomon Codes

Reed-Solomon codes are BCH-codes where the length $n$ divides $q - 1$. In particular, they are cyclic codes. They are almost never binary.

To create a Reed-Solomon code over $\mathbb{F}_7$, we use the primitive element $\alpha = 3$. The Reed-Solomon code of designed distance 3 over $\mathbb{F}_7$ is the cyclic code generated by

$$(X - \alpha)(X - \alpha^2) = (X - 3)(X - 2) = X^2 + 2X + 6.$$ 

The generator matrix of the code in cyclic form is

$$\begin{bmatrix}
6 & 2 & 1 & 0 & 0 & 0 \\
0 & 6 & 2 & 1 & 0 & 0 \\
0 & 0 & 6 & 2 & 1 & 0 \\
0 & 0 & 0 & 6 & 2 & 1
\end{bmatrix}.$$ 

Let us investigate this code. We start with the weight enumerator. The command

```
CODE_RS_6_4_7="\n621000\n062100\n006210\n000621"
```

computes the weight enumerator, which turns out to be

$$(1, 0, 0, 120, 360, 972, 948).$$ 

In polynomial form, this is

$$1y^6 + 120x^3y^3 + 360x^4y^2 + 972x^5y + 948x^6.$$ 

This confirms that the minimum distance is three.
Let us consider an example of a Reed-Solomon code in characteristic two: The Reed Solomon code of designed distance 3 over \( \mathbb{F}_8 \) is the cyclic code generated by

\[
(X - \alpha)(X - \alpha^2) = X^2 + 6X + 5.
\]

The associated cyclic generator matrix is

\[
\begin{bmatrix}
5 & 6 & 1 & 0 & 0 & 0 & 0 \\
0 & 5 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 5 & 6 & 1 & 0 & 0 \\
0 & 0 & 0 & 5 & 6 & 1 & 0 \\
0 & 0 & 0 & 0 & 5 & 6 & 1
\end{bmatrix}.
\]

We use the makefile variable \texttt{CODE\_RS\_8} to hold this generator matrix. The following command computes the weight enumerator

\begin{verbatim}
CODE_RS_8="\ 561000\ 056100\ 005610\ 000561\ 0000561"

RREF_RS_8_weight_enumerator:
▷ $(ORBITER\_PATH)orbiter.out.-v.2-
▷ -define:F.-finite_field:-q:8.-end-
▷ -define:v.-vector.-format:5.-field:F.
▷ -compact:$(CODE_RS_8).
▷ -end-
▷ -with:F.-do-
▷ -finite_field_activity-
▷ -weight_enumerator.v-
▷ -end
\end{verbatim}

which turns out to be

\[
y^7 + 245x^3y^4 + 1225x^4y^3 + 5586x^5y^2 + 12838x^6y + 12873x^7.
\]

Computing the automorphism group of the code is computationally infeasible. The next command performs field reduction on the code. This produces a \([21, 15]_2\) code.

\begin{verbatim}
RS_8_field_reduction:
▷ $(ORBITER\_PATH)orbiter.out.-v.2-
▷ -define:F.-finite_field:-q:8.-end-
▷ -with:F.-do-
\end{verbatim}
The reduced matrix is shown in Figure 10.8. Let us compute the weight enumerator of the reduced code. The command

```plaintext
RS_8_reduced="
010001100000000000000000
001110010000000000000000
110011001000000000000000
000010001100000000000000
000001110010000000000000
001100111000000000000000
000000010001100000000000
000000001110010000000000
000000000110010000000000
000000011001100000000000"
```
0000000000010001100000\n000000000001110010000\n000000000110011001000\n000000000000010001100\n000000000000001110010\n000000000000110011001”

RREF_RS_8_reduced_weight Enumerator:
\[ \text{(ORBITER PATH)} \text{orbiter.out -v} \]
\[ \text{-define F -finite_field -q 2 -end} \]
\[ \text{-define v -vector -format 15 -field F} \]
\[ \text{-compact $(RS 8$ reduced)} \]
\[ \text{-end} \]
\[ \text{-with F -do} \]
\[ \text{-finite_field_activity} \]
\[ \text{-weight Enumerator v} \]
\[ \text{-end} \]

computes the weight enumerator of the binary code. It is

\[
1y^{21} + 28x^5y^{18} + 84x^4y^{17} + 273x^5y^{16} + 924x^6y^{15} + 1956x^7y^{14} + \\
2982x^8y^{13} + 4340x^9y^{12} + 5796x^{10}y^{11} + 5796x^{11}y^{10} + 4340x^{12}y^9 + \\
2982x^{13}y^8 + 1956x^{14}y^7 + 924x^{15}y^6 + 273x^{16}y^5 + 84x^{17}y^4 + 28x^{18}y^3 + \\
x^{21} \]

In particular, the field reduced Reed-Solomon code is not optimal. It has minimum distance three, but there are codes of minimum distance 4. Here is one. We store the code to a file and then draw the generator matrix as bitmap.

CODE_21_15_4=""\n1110001000000000000000\n1010000100000000000000\n1010000100000000000000\n0110000010000000000000\n110010000010000000000\n1010000001000000000000\n0110100000100000000000\n1001100000010000000000\n0101100000001000000000\n0011100000000100000000\n1111000000000010000000\n1100010000000000100000\n1010010000000000010000\n0110010000000000001000\n10010100000000000000001

291
CODE_21_15_4_store:
  ▷ $(ORBITER_PATH)orbiter.out-\v.2\$
  ▷ ▷ -store_as_csv_file:"code_21_15_4.csv"\$
  ▷ ▷ 15\cdot21\cdot$(CODE_21_15_4).
  ▷ $(ORBITER_PATH)orbiter.out-\v.2-draw_matrix\$
  ▷ ▷ -input_csv_file:code_21_15_4.csv\$
  ▷ ▷ -box_width\cdot40--bit_depth\cdot24\$
  ▷ ▷ -partition\cdot4."15"."21".\$
  ▷ ▷ -end

We compute the weight enumerator

CODE_21_15_4_weight Enumerator:
  ▷ $(ORBITER_PATH)orbiter.out-\v.2\$
  ▷ ▷ -define:F:-\text{}finite_field:-q.2-end\$
  ▷ ▷ -define:v:-vector:-format.15:-\text{}field:F\$
  ▷ ▷ ▷ -compact:$(CODE_21_15_4).\$
  ▷ ▷ -end\$
  ▷ ▷ -with:F:-do\$
  ▷ ▷ -\text{}finite_field_activity\$
  ▷ ▷ -weight Enumerator:v\$
  ▷ ▷ -end

which turns out to be

$$1y^{21}+221x^4y^{17}+1600x^6y^{15}+6498x^8y^{13}+10912x^{10}y^{11}+9250x^{12}y^9+3584x^{14}y^7+669x^{16}y^5+32x^{18}y^3+1x^{20}y.$$  

This shows that this code is a $[21,15,4]_2$. It is optimal.
10.8 Bounds

In coding theory, one main question is to determine the best value of $d_{\text{max}}$ for a fixed $n$, $k$ and $q$ such that a linear $[n, k, d]_q$ code exists. There are many bounds, both upper and lower bounds. An upper bound tells us that no code with $d \geq d_{\text{max}}$ exists. A lower bound tells us that a code with $d \geq d_{\text{max}}$ exists. The command

```
bounds_for_d_given_n15_k6_q2:
> $(ORBITER_PATH)orbiter.out:-v.2:\n> -make_bounds_for_d_given_n_and_k_and_q:15:6:2
```

gives upper and lower bounds on the optimal minimum distance $d_{\text{max}}$ of a $[16, 6]_2$ code. The values of the Gilbert-Varshamov lower bound and the Singleton, Hamming, Plotkin and Griesmer upper bounds are computed. The output is:

```
d_{\text{GV}} = 5
d_{\text{singleton}} = 10
d_{\text{hamming}} = 6
d_{\text{plotkin}} = 7
d_{\text{griesmer}} = 6
```

This shows that $5 \leq d_{\text{max}} \leq 6$. The command

```
coding_theory_bounds_q2:
> $(ORBITER_PATH)orbiter.out:-v.2:-table_of_bounds:20:2
```

produces a table of bounds for binary codes with $n, k \leq 20$. A file

```
table_of_bounds_n20_q2.csv
```

is computed. The command

```
GV_n15_k6_d5:
> $(ORBITER_PATH)orbiter.out:-v.2:\n> -define:F:-finite_field:-q:2:-end:\n> -define:P:-projective_space:8:F:-end:\n> -with:P:-do:\n> -projective_space_activity:-make_gilbert_varshamov_code:15:5:-end
```

creates a $[15, 6, d]_2$ with minimum distance $g \geq 5$ using a greedy algorithm based on the proof of the Gilbert-Varshamov bound. The code that is produced has the following generator matrix:

```
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 1 0 0 0
1 1 1 0 0 1 1 0 0 0 0 1 0 0 0
1 1 0 1 0 1 0 1 0 0 0 0 1 0 0
1 0 1 0 1 0 1 1 0 0 0 0 0 1 0
1 0 1 1 0 1 0 0 1 0 0 0 0 0 1
```
To compute the minimum distance of the code, we do:

```plaintext
CODE_GV_N15_K6="
111111111100000
111110000010000
111001100001000
110101010000100
101010110000010
101101001000001"
```

GV_n15_k6_weight_enumerator:

```plaintext
$\text{ORBITER\_PATH}/\text{orbiter.out}-v.2-
$\text{define}\_F\_\text{-finite\_field}\_\text{\_q}2\_\text{-end}\_\text{-define}\_v\_\text{-vector}\_\text{-format}6\_\text{-field}\_F\_\text{-compact}\_\text{\_CODE\_GV\_N15\_K6}\_\text{-end}\_\text{-with}\_F\_\text{-do}\_\text{-finite\_field\_activity}\_\text{-weight\_enumerator}\_v\_\text{-end}
```

The weight enumerator is

$$1y^{15} + 27x^6y^9 + 24x^8y^7 + 9x^{10}y^5 + 3x^{12}y^3.$$  

From this, we see that the code has minimum distance 6, which is better than predicted.
10.9 Classification of Optimal Linear Codes

The classification problem of optimal codes in coding theory is the problem of determining the equivalence classes of codes for a given set of values of \( n \) and \( k \) and \( q \) with a lower bound on \( d \). Orbiter can be used to classify linear codes with given redundancy and bounded minimum distance. The redundancy of a linear \([n, k]\) code is the parameter \( r = n - k \). Codes with redundancy \( r \) can be identified with subsets of \( PG(r-1, q) \). Under this correspondence, a code with minimum distance at least \( d \) corresponds to a subset such that any \( d-1 \) elements are independent. We use the notation \( \Lambda_{r-1,s}(q) \) to denote the poset of subsets of \( PG(r-1, q) \) for which any \( d-1 \)-subset (if any) is independent. Under the correspondence, the action of \( PGL(r, q) \) on \( \Lambda_{r-1,s}(q) \) corresponds to the orbits of equivalent linear codes. For this reason, we are interested in determining the orbits of \( PGL(r, q) \) on \( \Lambda_{r-1,s}(q) \). An orbit of size \( n \) represents an isometry class of \([n, n-r, d; q]\) codes with \( d \geq s + 1 \). The projective stabilizer of the subset is the automorphism group of the code. The Orbiter command

```bash
codes_8_4_4:
  $(ORBITER_PATH)orboter.out -v 6 \n  -orboter_path=$(ORBITER_PATH) \n  -define G \n  -linear_group -PGL 4 2 \n  -group_theoretic_activity \n  -poset_classification_control \n  -problem_label codes_8_4_4 \n  -draw_poset \n  -draw_options -embedded -radius 250 \n  -line_width 1.0 -spanning_tree -end \n  -report -end \n  -linear_codes 3 8 \n  -end
```

classifies linear codes with redundancy 4 and minimum distance at least 4. Orbiter confirms that there is exactly one such code, and it computes the code together with the projective stabilizer. Orbiter creates the action of the group \( PGL(4, 2) \) on the poset \( \Lambda_{3,3}(2) \). Using poset classification, Orbiter then produces the poset of orbits shown in Figure 10.9. In this diagram, the numbers stand for Orbiter ranks of points in \( PG(3, 2) \). All nodes except for the root node have a number attached to it. The nodes represent subsets. In order to determine the set associated to a node, follow the path from the root node to the node and collect the points according to their labels. The root node represents the empty set. The \([8, 4, 4; 2]\)-code is represented by the set \( \{0, 1, 2, 3, 8, 11, 13, 14\} \). The fact that there is only one node at level
Figure 10.9: Orbits of PGL(4, 2) on the poset $\Lambda_{3,3}(2)$
8 in the poset of orbits tells us that the code is unique up to equivalence. Let us look at the code. The elements of the set \{0, 1, 2, 3, 8, 11, 13, 14\} are points in PG(3, 2). We write the coordinate vectors in the columns of a matrix $H$:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$ 

This matrix is the parity check matrix $H$ of the code $C$. This means that the words of the code are the vectors $c$ such that $c \cdot H^\top = 0$. Observe that the vectors that we put in the columns of $H$ all have odd weight. They are in fact the points of the hyperplane $x + y + z + w = 0$. This shows that the stabilizer of the code which is the stabilizer of the set is equal to $AGL(3, 2)$, a group of order 1344.
Chapter 11

Combinatorics

11.1 Introduction

In Tables 11.1 and 11.2, global Orbiter commands for Combinatorics are summarized. Global means that the commands are not associated with an activity related to an object.

The command

\[
\text{Sym}_{10}.\text{conj} \text{classes:}
\]

\[
\text{\texttt{\$(ORBITER\_PATH)orbiter.out\_v\_2\_conjugacy\_classes\_Sym\_n\_10}}
\]

\[
\text{\texttt{open classes Sym\_10.csv}}
\]

produces a list of the conjugacy classes of Sym(10). The list is written to a csv file. On Macintosh, the open command invokes Numbers, which can be used to create a pie chart of the class size distribution (see Fig. 11.1).

The next command computes the character table of the symmetric group Sym(4):

\[
\text{Char Sym}_{4}:
\]

\[
\text{\texttt{\$(ORBITER\_PATH)orbiter.out\_v\_2\_character\_table\_symmetric\_group\_4}}
\]

The command produces the following output:

The character table of Sym(4) is the matrix

\[
\begin{bmatrix}
1 & -1 & 1 & 1 & -1 \\
3 & 1 & 0 & -1 & -1 \\
2 & 0 & -1 & 2 & 0 \\
3 & -1 & 0 & -1 & 1 \\
1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-random_permutation</code></td>
<td>n fname</td>
<td>Creates a random permutation in Sym(n) and stores it in the given file.</td>
</tr>
<tr>
<td><code>-create_random_k_subsets</code></td>
<td>n k N</td>
<td>Creates N random k-subsets of an n-set.</td>
</tr>
<tr>
<td><code>-read_poset_file</code></td>
<td>fname</td>
<td>Reads a poset from the given file.</td>
</tr>
<tr>
<td><code>-read_poset_file_with_grouping</code></td>
<td>fname x-stretch</td>
<td>Reads a poset from the given file and sets stretch factor for orbit grouping.</td>
</tr>
<tr>
<td><code>-list_parameters_of_SRG</code></td>
<td>v_max</td>
<td>Performs a sift for putative parameter sets of SRGs.</td>
</tr>
<tr>
<td><code>-conjugacy_classes_SYM_n</code></td>
<td>n</td>
<td>Compute a list of conjugacy classes of Sym(n).</td>
</tr>
<tr>
<td><code>-tree_of_all_k_subsets</code></td>
<td>n k</td>
<td>Creates a tree-file for all k-subsets of an n-set.</td>
</tr>
<tr>
<td><code>-Delandtsheer_Doyen</code></td>
<td></td>
<td>See Section 11.7.</td>
</tr>
<tr>
<td><code>-tdo_refinement</code></td>
<td></td>
<td>See Section 11.8.</td>
</tr>
<tr>
<td><code>-tdo_print</code></td>
<td></td>
<td>See Section 11.8.</td>
</tr>
<tr>
<td><code>-convert_stack_to_tdo</code></td>
<td></td>
<td>See Section 11.8.</td>
</tr>
<tr>
<td><code>-maximal_arc_parameters</code></td>
<td></td>
<td>See Section 11.8.</td>
</tr>
<tr>
<td><code>-arc_parameters</code></td>
<td></td>
<td>See Section 11.8.</td>
</tr>
<tr>
<td><code>-pentomino_puzzle</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11.1: Commands related to Combinatorics (Part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-draw_layered_graph</code></td>
<td>options</td>
<td>Draws a graph.</td>
</tr>
<tr>
<td><code>-make_elementary_symmetric_functions</code></td>
<td>$n \ k_{\text{max}}$</td>
<td>Computes the elementary symmetric functions in $n$ variables of degree 1, $\ldots$, $k_{\text{max}}$.</td>
</tr>
<tr>
<td><code>-Dedekind_numbers</code></td>
<td>$n_{\text{min}} \ n_{\text{max}} \ q_{\text{min}} \ q_{\text{max}}$</td>
<td>Computes the Dedekind numbers $D_{n,q}$ for $n_{\text{min}} \leq n \leq n_{\text{max}}$ and $q_{\text{min}} \leq q \leq q_{\text{max}}$.</td>
</tr>
<tr>
<td><code>-rank_k_subset</code></td>
<td>$n \ k \ L$</td>
<td>Computes the ranks of $k$-subsets chosen from an $n$-set. $L$ is a list of $k$-sets taken from an $n$-set.</td>
</tr>
<tr>
<td><code>-geometry_builder</code></td>
<td></td>
<td>See Section 11.4.</td>
</tr>
<tr>
<td><code>-character_table_symmetric_group</code></td>
<td>$n$</td>
<td>Computes the character table of $\text{Sym}(n)$ using the algorithm of Burnside.</td>
</tr>
<tr>
<td><code>-domino_portrait</code></td>
<td>$D \ s \ \text{fname}$</td>
<td>Computes a domino portrait for a graphics file in r/g/b format using double $D$ domino sets.</td>
</tr>
</tbody>
</table>

Table 11.2: Commands related to Combinatorics (Part 2)
The following command creates the elementary symmetric functions in 4 variables.

```
$ (ORBITER_PATH) orbiter.out --make_elementary_symmetric_functions 4 4
```

The output is:

```
k=1 :
x0 + x1 + x2 + x3
k=2 :
x0*x1 + x0*x2 + x0*x3 + x1*x2 + x1*x3 + x2*x3
k=3 :
x0*x1*x2 + x0*x1*x3 + x0*x2*x3 + x1*x2*x3
k=4 :
x0*x1*x2*x3
```

Orbiter can compute domino portraits. To do so, we need an input file in r/g/b format of size \((D + 1)s \times Ds\), where \(D = 7\) for double-six dominos.
The portrait is shown in Figure 11.2. It is possible to compare the domino portrait with a grayscale version of the input image. The following command creates a grayscale image of the input file that was written during the previous command.

domino_portrait_input:
> $(ORBITER_PATH)orbiter.out\-v\-domino_portrait\-7\-anton_28x32\-end

> $(ORBITER_PATH)orbiter.out\-v\-2\-define all_one_r \-vector \-repeat 1\-28\-end
> $define all_one_c \-vector \-repeat 1\-32\-end
> $draw_matrix
> $grayscale
> $invert_colors
> $input_csv_file anton_28x32_m.csv
> $box_width 20 \-bit_depth 8
> $partition 3
> all_one_c \all_one_r
Figure 11.3: Domino portrait input file in grayscale

```
open-anton_28x32_m_draw.bmp
```

The grayscale version of the input file is shown in Figure 11.3.
11.2 Diophantine Systems

Diophantine systems of equations and inequalities arise frequently in Combinatorics. In Table 11.3, Orbiter commands for creating and solving diophantine systems are shown. In Table 11.4, Orbiter activities for diophantine systems are shown.

Consider the matrix

\[
A = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

Suppose we want to find all column vectors \( x \) with entries in 0, 1 such that

\[
Ax = 1
\]

where \( 1 \) is the all-one column vector. Orbiter offers two algorithms to do this. One is McKay’s possolve, the other is Knuth’s dancing links (DLX). In order to get started, we need to create a diophant object. The following example shows how this is done. Note the use of makefile variables:

\[
\text{TEST\_SYSTEM} = "\\
0,1,0,1,0,0,\\
0,0,1,0,1,0,\\
1,0,1,0,0,0,\\
0,1,0,1,0,1,\\
1,0,0,0,0,1,\\
1,0,1,0,0,0,\\
0,1,0,0,1,1"
\]

\[
\text{TEST\_RHS} = "1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"
\]

\begin{verbatim}
test_system:
  $(ORBITER\_PATH) orbiter.out -v 4 -
  -define A -vector -format 7 -dense $(TEST\_SYSTEM) -end
  -define D -diophant
  -label test_system
  -coefficient_matrix A
  -RHS $(TEST\_RHS)
  -x_min_global 0 -x_max_global 1
  -end
\end{verbatim}

There are two commands to solve a diophantine system: -solve_mckay and -solve_DLX. The latter is more restrictive, as it allows only 0,1 variables. Here is the McKay solver:
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-label</td>
<td>label</td>
<td>Use the given name as file name.</td>
</tr>
<tr>
<td>-coefficient_matrix</td>
<td>A</td>
<td>Set the coefficient matrix to the previously created vector with format information.</td>
</tr>
<tr>
<td>-coefficient_matrix_csv</td>
<td>fname</td>
<td>Read the coefficient matrix from the given csv-file.</td>
</tr>
<tr>
<td>-RHS</td>
<td>list-of-integers</td>
<td>3n values: (RHS-low, RHS-high, RHS-type) for each row of the system.</td>
</tr>
<tr>
<td>-RHS_csv</td>
<td>fname</td>
<td>Read the RHS information from the given csv file.</td>
</tr>
<tr>
<td>-RHS_constant</td>
<td>low,high,type</td>
<td>Set the RHS according to low,high,type.</td>
</tr>
<tr>
<td>-x_max_global</td>
<td>a</td>
<td>Set the upper bound for all variables to a.</td>
</tr>
<tr>
<td>-x_min_global</td>
<td>a</td>
<td>Set the lower bound for all variables to a.</td>
</tr>
<tr>
<td>-x_bounds</td>
<td>list-of-values</td>
<td>Set the lower and upper bounds for all variables.</td>
</tr>
<tr>
<td>-x_bounds_csv</td>
<td>fname</td>
<td>Read the lower and upper bounds for all variables from the given file.</td>
</tr>
<tr>
<td>-has_sum</td>
<td>s</td>
<td>For the sum of the variables to be s.</td>
</tr>
<tr>
<td>-maximal_arc</td>
<td>s d secants subset</td>
<td>Create system for a maximal arc of size s and degree d in PG(2,q). Use the given set of two pencil lines. The subset picks the lines from the given pencils which are external.</td>
</tr>
<tr>
<td>-q</td>
<td>q</td>
<td>Use PG(2,q) for maximal arcs.</td>
</tr>
<tr>
<td>-override_polynomial</td>
<td>a</td>
<td>Use polynomial numerically coded as a for creating $\mathbb{F}_q$.</td>
</tr>
</tbody>
</table>

Table 11.3: Orbiter Commands to create Diophantine systems
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-print</td>
<td></td>
<td>Print the system.</td>
</tr>
<tr>
<td>-solve_mckay</td>
<td></td>
<td>Solve the system using McKay’s pos-solve.</td>
</tr>
<tr>
<td>-solve_DLX</td>
<td></td>
<td>Solve the system using Knuth’s dancing links.</td>
</tr>
<tr>
<td>-solve_standard</td>
<td></td>
<td>Solve the system using the standard solver.</td>
</tr>
<tr>
<td>-draw</td>
<td></td>
<td>Produce a drawing of the coefficient matrix of the system.</td>
</tr>
<tr>
<td>-draw_as_bitmap</td>
<td>$w$ $b$</td>
<td>Produce a bitmap drawing of the coefficient matrix of the system, using boxes of $w$ pixels with. Set the color bit-depth to $b$ ($b = 8$ or $b = 24$). The output is a bmp-file.</td>
</tr>
<tr>
<td>-perform_column_reductions</td>
<td></td>
<td>Eliminate variables which must be zero and write a reduced system.</td>
</tr>
<tr>
<td>-test_single_equation</td>
<td></td>
<td>For each row of the system, compute the number of solutions of the system restricted to the nonzero coefficients.</td>
</tr>
<tr>
<td>-project_to_single_equation_and_solve</td>
<td>$i$ $j$</td>
<td>Solve the system assuming the $j$th solution to the restricted system consisting of the $i$th row.</td>
</tr>
<tr>
<td>-project_to_two_equations_and_solve</td>
<td>$i$ $j$ $r$ $m$</td>
<td>Solve the system assuming any solution to the restricted system consisting of the $i$th and the $j$-th row whose number is congruent to $r$ mod $m$.</td>
</tr>
</tbody>
</table>

Table 11.4: Orbiter activities for Diophantine systems
McKay_test:

```
▷ $(ORBITER_PATH)orbiter.out:v.4:
▷▷ -define: A: -vector: -format:7: -dense: $(TEST_SYSTEM):-end:
▷▷ -define: D: -diophant:
▷▷▷ -label: test_system:
▷▷▷ -coefficient_matrix: A:
▷▷▷ -RHS: $(TEST_RHS):
▷▷▷ -x_min_global: 0: -x_max_global: 1:
▷▷ -end:
▷▷ -with: D: -do:
▷▷▷ -diophant_activity: -solve: mckay:
▷▷ -end
```

The solutions are written to the file DLX_test.sol. And now the dancing links solver:

DLX_test:

```
▷ $(ORBITER_PATH)orbiter.out:v.4:
▷▷ -define: A: -vector: -format:7: -dense: $(TEST_SYSTEM):-end:
▷▷ -define: D:
▷▷ -diophant: -label: test_system:
▷▷▷ -coefficient_matrix: A:
▷▷▷ -RHS: $(TEST_RHS):
▷▷▷ -x_min_global: 0: -x_max_global: 1:
▷▷ -end:
▷▷ -with: D: -do:
▷▷▷ -diophant_activity: -solve: DLX:
▷▷ -end
```
11.3 Combinatorial Linear Spaces

A linear space is a pair \((S, L)\) where \(S\) is a set and \(L\) is a set of subsets of \(S\) such that each set \(L \in L\) satisfies \(|L| \geq 2\) and moreover for any two \(a, b \in S\) there is exactly one element \(L \in L\) such that both \(a\) and \(b\) belong to \(L\). The usual notions of isomorphism and automorphism apply. For finite linear spaces, a first combinatorial property is the number \(a_i\) which counts the number of sets \(L \in L\) of size \(i\). The vector \((a_2, \ldots, a_n)\) is the line type of \((S, L)\). The equation
\[
\binom{n}{2} = \sum_{j=2}^{n} a_j \binom{j}{2}
\]  
(11.1)
is satisfied. The equation can be used to generate all possible line types of a putative linear space. Here is an example. For \(|S| = 6\), (11.1) becomes
\[
x_0 \frac{6}{2} + x_1 \frac{5}{2} + x_2 \frac{4}{2} + x_3 \frac{3}{2} + x_4 \frac{2}{2} = \frac{6}{2}.
\]
Here, \((x_0, x_1, \ldots, x_4)\) is the line type of a putative linear space on 6 points. That is, \(x_i = a_{6-i}\) is the number of lines of size \(6 - i\). The extended coefficient matrix of the system is
\[
[15, 10, 6, 3, 1 | 15].
\]
The Orbiter command
\[
\text{linsp6:}
\]
\[
\text{▷ $(ORBITER\_PATH)orbiter.out.-v.4.}\n\]
\[
\text{▷ ▷ -define A -vector -format 1 -dense:"15,10,6,3,1" -end}\n\]
\[
\text{▷ ▷ -define D -diophant -label linsp6}\n\]
\[
\text{▷ ▷ -coefficient matrix A}\n\]
\[
\text{▷ ▷ -RHS."15,15,1".}\n\]
\[
\text{▷ ▷ -x_min_global.0.}\n\]
\[
\text{▷ ▷ -x_max_global.15.}\n\]
\[
\text{▷ ▷ -end}\n\]
\[
\text{▷ ▷ -with D -do}\n\]
\[
\text{▷ ▷ ▷ -diophant_activity -solve mckay}\n\]
\[
\text{▷ ▷ -end}\n\]
solves the system using McKay’s program posolve [49]. The program finds 15 solutions, written to the file linsp6.sol.

Let us consider a problem from [11]. Suppose we are interested in linear spaces on 30 points, with line type \((7, 5^{27}, 4^{24})\). This notation means that we assume one 7-lines, 27 5-lines and 24 4-lines. The type of a point \(P\) is the set of integers
\[
p_j = \#j\text{-lines through } P.
\]

309
We are trying to precompute the matrix of point types

$$(p_{ij})$$

where $j = 7, 5, 4$ and $i$ belongs to an index set of all possible point types. Fixing a point $P$, counting points $Q \neq P$ collinear with $P$ yields

$$6p_7 + 4p_5 + 3p_4 = 29, \quad p_7 \leq 1, \quad p_5 \leq 27, \quad p_4 \leq 24.$$ 

Using the Orbiter commands

```
linsp30_pt_types:
  ▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot4\cdot$
  ▷ ▷ -define A -vector -format 1 -dense "6,4,3" -end ▷
  ▷ ▷ -define D -diophant ▷
  ▷ ▷ ▷ -label linsp30_pt_types ▷
  ▷ ▷ ▷ -coefficient_matrix A ▷
  ▷ ▷ ▷ -RHS "29,29,1" -x_bounds "0,1,0,27,0,24" ▷
  ▷ ▷ ▷ -end ▷
  ▷ ▷ ▷ -with D -do ▷
  ▷ ▷ ▷ -diophant_activity -solve mckay ▷
  ▷ ▷ -end ▷
```

we determine the possibilities

$$(p_7,p_5,p_4) = \begin{cases} 
1 & 5 & 1 \\
1 & 2 & 5 \\
0 & 5 & 3 \\
0 & 2 & 7 
\end{cases}$$

The rows in this matrix are called the point types ($i = 0, 1, 2, 3$). Let $b_i$ be the number of points of type $i$. By counting points, incident (point,line) pairs by $j$-lines and pairs of intersecting $j$-lines, we arrive at the following system:

$$b_0 + b_1 + b_2 + b_3 = 30$$
$$b_0 + b_1 = 7$$
$$5b_0 + 2b_1 + 5b_2 + 2b_3 = 135 = 27 \cdot 5$$
$$b_0 + 5b_1 + 3b_2 + 7b_3 = 96 = 24 \cdot 4$$
$$10b_0 + b_1 + 10b_2 + b_3 \leq 351 = \binom{27}{2}$$
$$10b_1 + 3b_2 + 21b_3 \leq 276 = \binom{24}{2}$$

Using the Orbiter commands
we determine the possibilities

\[
(b_0, b_1, b_2, b_3) = \begin{pmatrix} 2 & 5 & 23 & 0 \\ 3 & 4 & 22 & 1 \\ 4 & 3 & 21 & 2 \\ 5 & 2 & 20 & 3 \\ 6 & 1 & 19 & 4 \\ 7 & 0 & 18 & 5 \end{pmatrix}
\]
11.4 Classification of Configurations and Geometries

A partial linear space is a set system on a fixed set $V$. We write $L = (V, \mathcal{B})$, where $\mathcal{B}$ is a set of distinct subsets of $V$, called lines. The members of $V \cup \mathcal{B}$ are called elements. For two elements $x, y$, we say that $x$ is incident with $y$, written $xIy$, if either $x \in y$ or $y \in x$. We require that any line has at least two points and any two points are contained in at most one line. A decomposition of a linear space is a partition $\Pi = (C_1, \ldots, C_n)$ of $V \cup \mathcal{B}$ such that each $C_i$ either is a subset of $V$ or a subset of $\mathcal{B}$. A decomposition is called tactical if for all $i$, the incidence number

$$\iota(C_i, C_j) = \# \{ y \in C_j, xIy \}$$

does not depend on the choice of $x \in C_i$. Any linear space has a tactical decomposition, as the discrete partition (every element is in its own class) is tactical. Let $\text{Aut}(L)$ be the automorphism group of the linear space, which is the subgroup of $\text{Sym}(V)$ which preserves incidence. For $\alpha \in \text{Aut}(L)$ we say that the decomposition $\Pi$ preserves $\alpha$ if $\alpha$ fixes every class of $\Pi$. For $A \leq \text{Aut}(L)$, we say that $\Pi$ preserves $A$ if $\Pi$ preserves every element $\alpha \in A$. Mostly, we are interested in those decompositions $\Pi$ which preserve $\text{Aut}(L)$. In light of this, the discrete decomposition is not that interesting.

Any linear space has a coarsest tactical decomposition that preserves its automorphism group: The orbit partition of the automorphism group acting on $V \cup \mathcal{B}$ will do. Up to ordering of the classes, the coarsest tactical refinement is unique. Computing the orbit decomposition is challenging as it involves computing the automorphism group. Computationally, there are easier ways to get to admissible decompositions. One is by means of successive refinements. If a class $C_i$ does not have the property that $\iota(C_i, C_j)$ is well-defined for all $x \in C_i$, then a refinement of $C_i$ will do. The coarsest refinement of $C_i$ has the property that if $C_i$ preserves some group $A$ then the refinement will do, too. This shows that there is an algorithm to compute a tactical decomposition of any given linear space $\mathcal{P}$. Simply start with the decomposition of two classes, one the set of points and one the set of blocks, and refine. The output may or may not be equal to the decomposition arising from the orbit partition of $\text{Aut}(L)$.

Let us consider the opposite question. Given a tactical decomposition, does there exist a linear space whose coarsest tactical decomposition is the given one? If so, how many nonisomorphic partial linear spaces are there for a given tactical decomposition? in other words, we would like to classify the linear spaces which admit a given tactical decomposition. The $-\text{geometry\_builder}$ option can answer these kinds of questions.

The command

```
geo_10_3:
  ▸ $(\text{ORBITER\_PATH})\text{oriter.out}-v.2\$
  ▸ ▸ -define-Test_lines-set:-loop 4 11 1-end\$
  ▸ ▸ -geometry\_builder-V.10-B.10-TDO.3-fuse.1\$
  ▸ ▸ ▸ -fname_GEO.10_3\$
  ▸ ▸ ▸ -test-Test_lines\$
  ▸ ▸ -end
```

312
classifies the configurations $10_3$. It uses isomorphism tests after 4, 5, 6, 7, 8, 9 and 10 points. The positions of the tests is defined using a set called $\text{Test_lines}$. The set of test lines is defined using a loop command. The command shows that there are exactly 10 configurations of this kind. One of them is the Desargues configuration. A file $10_3.inc$ is written which contains all the partial linear spaces admitting the tactical decomposition. The file contains the incidences in increasing order. The position in the incidence matrix is given. One linear space is given per row, except for the first row and the last. The first row contains the number of points, the number of lines, and the number of incidences. The last row start with $-1$. Here is the file $10_3.inc$:

10 10 30
0 1 2 10 13 14 20 25 26 31 33 35 41 44 47 52 53 58 62 66 69 74 78 79 85 87 89 96 97 98
0 1 2 10 13 14 20 25 26 31 33 35 41 44 47 52 54 58 62 66 69 73 78 79 85 87 89 96 97 98
0 1 2 10 13 14 20 25 26 31 33 35 41 44 47 52 54 58 62 67 69 73 76 79 85 87 89 96 97 98
0 1 2 10 13 14 20 25 26 31 33 35 41 44 47 52 56 58 62 67 69 73 78 79 84 86 89 95 97 98
0 1 2 10 13 14 20 25 26 31 33 35 41 44 47 52 54 58 62 67 69 73 78 79 84 86 89 95 97 99
0 1 2 10 13 14 20 25 26 31 33 35 41 44 47 48 52 54 57 62 66 68 73 77 79 84 86 89 95 98 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 68 73 78 79 84 86 89 95 97 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 89 95 97 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 89 97 99 98
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 68 73 75 79 84 86 89 97 98 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 68 73 79 79 84 86 89 95 97 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 83 75 79 84 86 89 95 97 99
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 101 106 119
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 103 104 119
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 103 107 117
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 105 109 119
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 105 107 119
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 105 107 119
0 1 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 80 105 107 119
-1 10
120, 24, 12, 10, 6, 4^{-2}, 3^{-2}, 2

Two further files are written, containing the lines of the incidence geometry. The file $10_3.blocks$:

10 10 3
0 15 26 44 51 68 81 109 114 116
0 15 26 46 49 68 81 109 114 116
0 15 26 46 49 68 83 106 115 116
0 15 26 46 52 69 77 106 114 116
0 15 26 46 56 69 80 101 106 119
0 15 26 46 56 69 80 103 104 119
0 15 26 46 56 69 80 103 107 117
0 15 26 46 56 72 80 93 106 119
0 15 26 46 56 72 81 93 105 119
0 15 26 46 56 74 79 93 105 119
-1 10
120, 24, 12, 10, 6, 4^{-2}, 3^{-2}, 2

contains the blocks as ranked 3-subsets of a 10-element set. The file $10_3.blocks_long$ contains the list of blocks written out.

It is possible to create graphical representations of the search tree. The command below does so for the example that we just did. Note the additional option $\text{-search_tree}$. This option causes Orbiter to create a file containing the search tree. The name of the file is derived from the file name given with the $\text{fname_GEO}$ option. Here, the $\text{fname_GEO}$ option sets the output file to $10_3$. The $\text{-search_tree}$ option then creates the file $10_3_tree.txt$. 

313
In a second invocation of Orbiter, the `-tree_draw` command is used to draw a tree from the file 10_3_tree.txt that was just created. The green nodes are nodes that are accepted. The red nodes are nodes that are rejected. This means they represent geometries that have been seen before. The 10 green nodes at the very bottom of the diagram represent the 10 \(10_3\) configurations.

```
geo_10_3_tree:
  > $(ORBITER_PATH)orbiter.out-v.2\n  > -define:Test_lines-set:-loop-0.11.1-end\n  > -geometry_builder:-V.10.-B.10.-TDO.3.-fuse.1\n  > -fname:GEO.10_3\n  > -search_tree\n  > -test:Test_lines\n  > -end
  > $(ORBITER_PATH)orbiter.out-v.2\n  > -draw_options:-embedded:-radius.50\n  > -xin:10000.-yin:10000\n  > -xout:100000.-yout:500000\n  > -nodes_empty\n  > -scale:0.5.-line_width:0.3-end\n  > -tree_draw:10_3_tree.txt
  > pdflatex:10_3_tree_draw.tex
  > open:10_3_tree_draw.pdf
```

The resulting tree is shown in Figure 11.4.

Any incidence structure defines a graph on its underlying set of points. The vertices are the points of the incidence structure. Two vertices are adjacent if and only if the incidence structure contains a block which contains the associated points. In a geometric context, the graph is known as the collinearity graph of the geometry. The distance between two points is the distance of the associated vertices in the collinearity graph. The girth if the length of the shortest cycle. It is often desired to classify incidence structures with a given girth. This means that we are given an integer \(g\) (the girth), and that we are looking for incidence structures whose collinearity graph has no cycles of length less than \(g\). For instance, the following example classifies all cubic graphs on 10 vertices with girth at least 5:

```
geo_petersen:
  > $(ORBITER_PATH)orbiter.out-v.8\n  > -define:Test_lines-set:-loop-3.11.1-end\n  > -geometry_builder\n  > -V.10.-B.15.-TDO.3.-fuse.1\n  > -fname:GEO:petersen.-girth.5\n  > -test:Test_lines\n  > -end
```

314
There is a unique graph with these properties. It is the Petersen graph. Its automorphism group is $\text{Sym}(5)$ of order 120.

We can classify configurations with a given girth. For instance, while there are 245342 isomorphism classes of configurations $15_3$, only one of them has girth 4. This is the Cremona Richmond configuration. It is associated to a cubic surface. The following command classifies all configurations $15_3$:

```
geo_15_3:
  $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot 2\cdot \\
  \ -\text{define}\cdot \text{Test}\cdot \text{lines}\cdot \text{-set}\cdot \text{-loop}\cdot 4\cdot 16\cdot 1\cdot \text{-end}\cdot \\
  \ -\text{geometry}\cdot\text{builder}\cdot \\
  \ -v\cdot 15\cdot B\cdot 15\cdot TDO\cdot 3\cdot \\
  \ -\text{test}\cdot\text{Test}\cdot\text{lines}\cdot \\
  \ -\text{end}
```

This command takes about 8 minutes of time to complete. The next command classifies the $15_3$ with girth 4. Only one configuration arises, the Cremona Richmond, with automorphism group $\text{Sym}(6)$ of order 720.

```
geo_15_3\_g4:
  $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot 2\cdot \\
  \ -\text{define}\cdot \text{Test}\cdot \text{lines}\cdot \text{-set}\cdot \text{-loop}\cdot 4\cdot 16\cdot 1\cdot \text{-end}\cdot 
```
-geometry_builder:\
  -V.15-B.15-TD0.3:\
  -fuse.1-fname_GEO.15.3.g4:\
  -girth.4:\
  -test.Test_lines:\
  -end
11.5 Design Theory

A design is a special kind of incidence structure. The elements of the ground set are called points. The sets forming the design are called blocks. The incidence matrix of a design has rows corresponding to the points and columns corresponding to the blocks. An entry in a certain row and column is one if and only if the point associated with the row is contained in the block associated with the column, zero otherwise. A decomposition of the design is a partition of the points and blocks such that each class consists either exclusively of points or exclusively of blocks.

A decomposition is point-tactical if for all points, the number of incident lines in the \( j \)th block class depends only on the class of the point. If the point belongs to class \( i \), this number is denoted as \( a_{ij} \). A decomposition is block-tactical if for all blocks, the number of incident points in the \( i \)th point class depends only on the class of the block. If the block belongs to class \( j \), this number is denoted as \( b_{ij} \).

A projective plane of order \( n \) is a design with \( n^2 + n + 1 \) points and equally many blocks (also called lines), each of size \( n + 1 \) such that any two points lie in exactly one block and any two blocks have exactly one point in common. Projective planes are known to exist for all \( n = q \) which are a power of a prime. This follows from a construction which utilizes the projective geometry PG(2, \( q \)). Points are the one-dimensional subspaces of \( \mathbb{F}_q^3 \), blocks are the two-dimensional subspaces of \( \mathbb{F}_q^3 \), and incidence is natural (inclusion of subspaces). The automorphism group of this design is the collineation group of the projective space. Projective planes other than these exist, though none are known when \( n \) is not a prime power. The number of lines through a point equals the number of points on a line. The fact that these numbers exist imply that there is a tactical decomposition. Namely, the trivial decomposition with two classes, one containing all points and one containing all lines. The structure constants of the decomposition are the numbers just described.

The command

```
$design\_PG\_2\_3:
  $(ORBITER\_PATH)orbiter.out\(-v\-8\-\)
  \(>\) -create\_design\(--q\-3\--family\_PG\_2\_q\--)end
```

creates the design PG(2, 3) and its automorphism group:

We have created the following design:

\{19, 79, 126, 219, 256, 284, 371, 392, 465, 541, 619, 627, 653\}

The stabilizer is generated by:
Strong generators for a group of order 5616:

\[
\begin{bmatrix}
2 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad
\begin{bmatrix}
1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 1
\end{bmatrix}, \\
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 1
\end{bmatrix}, \quad
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix}, \quad
\begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

1,0,0,0,2,0,0,0,2, \\
1,0,0,0,2,0,0,0,1, \\
1,0,0,0,1,0,1,0,1, \\
1,0,0,0,1,0,0,1,1, \\
1,0,0,0,0,1,0,1,0, \\
0,1,0,1,0,0,0,1,

The blocks of the design are encoded in the lexicographic ordering of \(k\)-subsets (here \(k = 4\)).

The program also displays the tactical decomposition schemes of the design, which are

\[
\begin{array}{c|c|c}
\rightarrow & 13_1 & \\
\hline
13_0 & 4 & 13_1 \\
\hline
13_0 & 4 & 13_0
\end{array}
\]

One way to construct designs is by assuming a suitable group of symmetries. Let us consider an example. It is possible to construct \(t-(v, k, \lambda)\) designs invariant under a permutation group \(G\) acting on a set \(V\) with \(|V| = v\) as follows: Classify the orbits of \(G\) on subsets of size \(k\) and less. Construct a matrix which describes the relationship between the orbits on \(t\)-sets and the orbits on \(k\)-sets. This matrix is often referred to as the Kramer-Mesner matrix (cf. [39]). For each pair of \(t\)-orbit and \(k\)-orbit, for instance with representatives \(T\) and \(K\), say, we count the number of elements in the orbit of \(K\) which contain \(T\). The rows of the matrix are in correspondence to the \(t\)-orbits, while the columns are in correspondence to the \(k\)-orbits. The matrix entry \(a_{ij}\) is the number just defined where \(T\) is the representative of the \(i\)-th orbit on \(t\)-sets, and where \(K\) is the representative of the \(j\)-th orbit on \(k\)-sets. Let \(M_{t,k}(G)\) be the Kramer-Mesner matrix for the group \(G \leq \text{Sym}(V)\) defined in this way. The \(t-(v, k, \lambda)\) designs invariant under \(G\) are in one-to-one correspondence to the solutions of

\[M_{t,k}(G) \cdot x = \lambda \mathbf{1},\]

where \(x\) is a column vector of zeros and ones and \(\mathbf{1}\) is the column vector of all ones. The length of \(x\) is the number of \(k\)-orbits of \(G\) on \(V\), while the length of \(\mathbf{1}\) is the number of \(t\)-orbits of \(G\) on \(V\). Any vector \(x\) satisfying the matrix equation corresponds to a design
invariant under $G$. Simply take the blocks of the design to be the union of those orbits of $G$ on $k$-subsets whose associated entry in $x$ is one. We assume the group $\text{PGL}(2, 32)$ in the action on points of the projective line $\text{PG}(1, 32)$ over the field $\mathbb{F}_{32}$. The parameters of the design are $7-(33, 8, 10)$, that is, each 7-subset of $\text{PG}(1, 32)$ is covered exactly 10 times by the chosen 8-subsets comprising the design. The first orbiter command creates the group $\text{PGL}(2, 32)$ and computes the Kramer-Mesner matrix

$$M_{7,8}(\text{PGL}(2, 32)).$$

The number of 7-orbits is 32. The number of 8-orbits is 97. Correspondingly, the Kramer-Menser matrix has 32 rows and 97 columns. The matrix is stored in the csv-file

$$\text{KM\_PGGL\_2\_32\_KM\_7\_8.csv}.$$ 

The second command produces the graphical representation of the matrix shown in Figure 11.5 (different colors represent different values of entries in the matrix). The third Orbiter command creates the diophantine system associated with the Kramer-Mesner matrix.

```
KM_PGLGL_2_32:
  $(\text{ORBITER\_PATH})\text{orbiter.out}\text{-v.3}\text{-}\\
  \text{-define}\text{-linear\_group}\text{-PGL}\text{-2}\text{-32}\text{-end}\text{-}\\
  \text{-with}\text{-G\_do}\text{-}\\
  \text{-group\_theoretic\_activity}\text{-}\\
  \text{-poset\_classification\_control}\text{-}\\
  \text{-problem\_label}\text{KM\_PGGL}\text{-2}\text{-32}\text{-W\_depth}\text{-8}\text{-}\\
  \text{-Kramer\_Mesner\_matrix}\text{-7}\text{-8}\text{-}\\
  \text{-draw\_poset}\text{-}\\
  \text{-draw\_options}\text{-embedded}\text{-sideways}\text{-radius}\text{-50}\text{-}\\
  \text{-scale}\text{-0.5}\text{-line\_width}\text{-0.3}\text{-end}\text{-}\\
  \text{-end}\text{-}
```
The last command performs a complete enumeration of all solutions by solving the system and producing the solution vectors $x$ which correspond to the designs.
11.6 Design Theory – Large Sets

Fix a set of size \( v \) and an integer \( k \) with \( 1 < k < v \). Is it possible to partition the set of \( k \)-subsets of \( v \) into designs, all with the same parameters? If so, the resulting set of designs is called a large set (of designs). So, a large set of designs is a set of designs, all of the same types, on a fixed \( v \)-element set whose block sets are pairwise disjoint and partition the set of \( k \)-subsets. Let us see how Orbiter can help construct and classify small large sets.

Suppose we consider \( AG(2,3) \), the affine plane of order 3. It is a configuration with 9 points, 12 lines, 4 lines on each point and 3 points on each line. To see if it is unique, we use the following command:

```
geo 9 4 12 3:
  ▷ $(ORBITER_PATH)orbiter.out-v.2-\
  ▷ -geometry_builder-\
  ▷ ▷ -V.9-B.12-\
  ▷ ▷ -TDO.4-fuse.1-\
  ▷ ▷ -fname_GEO AG 2 3-\
  ▷ ▷ -test 3,4,5,6,7,8,9-\
  ▷ ▷ -end
```

The command produces the file \( AG_2_3.inc \), which contains the following lines:

```
9 12 36
0 1 2 3 12 16 18 24 31 32 33 37 40 43 46 49 53 56 59 62 64 69 71 74 78 80 82 87 89 93 94 99 102 103 107
-1 1
432
```

This shows that the design is unique, and has an automorphism group of order 432. We can use the following command to check the automorphism group:

```
geo 9 4 12 3.c:
  ▷ $(ORBITER_PATH)orbiter.out-v.2-\
  ▷ -define C-combinatorial_objects-\
  ▷ ▷ -file_of_incidence_geometries-\
  ▷ ▷ -AG_2_3.inc 9 12 36-\
  ▷ ▷ -end-\
  ▷ ▷ -with C-do-\
  ▷ ▷ -combinatorial_object_activity-\
  ▷ ▷ -canonical_form-\
  ▷ ▷ -classification_prefix AG 2 3-\
  ▷ ▷ -save ago-\
  ▷ ▷ -end
```

This command writes several files: \( AG_2_3.blocks \) contains the list of blocks as ranks of \( k \)-subsets. \( AG_2_3.blocks_long \) contains the list of blocks as \( k \)-subsets. \( AG_2_3.ago.csv \) contains the automorphism group order of the design. For the following commands, we will
treat blocks of the design as sets of ranks of \( k \)-subsets. We can now create a table of all designs \( AG(2, 3) \), as orbit under the group \( \text{Sym}(9) \). The following command does that:

\[
AG_{2,3}\text{BLOCKS}=0,13,22,27,35,41,47,53,55,59,71,76
\]

The number of designs is \( |\text{Sym}(9)|/432 = 362880/432 = 840 \). To find all large sets, we establish the block-disjointness graph on this set of designs and find all cliques of size 7:

\[
\text{LS}_AG_{2,3}\text{disjoint sets graph and cliques:}
\]

\[
\text{define } \Gamma \text{-graph}
\]

\[
\text{with } \Gamma \text{-do}
\]

\[
\text{graph theoretic activity}
\]

\[
\text{find cliques -target size 7 -end}
\]

\[
\text{print symbols}
\]

The files \( AG_{2,3}\text{design_table_disjoint_sets_sol.txt} \) and \( AG_{2,3}\text{design_table_disjoint_sets_sol.csv} \) are created, each containing the cliques of size 7. There are exactly 15360 cliques of size 7. It remains to classify the resulting 15360 large sets up to isomorphism. To do that, we first need to create the actual large sets from the cliques. The following command does that:

\[
\text{LS}_AG_{2,3}\text{export solutions:}
\]

\[
\text{define } \mathcal{D} \text{-design -list of blocks 9-3}
\]

\[
\text{with } \mathcal{D} \text{-do}
\]

\[
\text{define } \text{Sym}9 \text{-permutation group -symmetric group 9 -end}
\]

\[
\text{define } \mathcal{T} \text{-design table-D:"AG_2_3"-Sym9}
\]

\[
\text{with } \mathcal{T} \text{-do}
\]

\[
\text{graph theoretic activity}
\]

\[
\text{find cliques -target size 7 -end}
\]

\[
\text{print symbols}
\]

322
The final step to classify the large sets up to isomorphism will be discussed in Section 15.4.
11.7 Design Theory – Delandtsheer-Doyen

Delandtsheer and Doyen in [22] study line-transitive and point-imprimitive designs and show that they are rare in a certain sense. Orbiter can be used to construct such designs assuming that there is a grid structure on the set of points and assuming that the design is invariant under a chosen group $G$. The group $G$ is assumed to be a subgroup of the group $\text{AGL}(d_1,q_1) \times \text{AGL}(d_2,q_2)$ acting on a grid of size $q_1^{d_1} \times a_2^{d_2}$ in product action.

Finite projective planes often arise in this context. However, not all examples are projective planes. Orter can help to classify small examples. Let us consider an example. Suppose we want to classify all designs on 21 points with blocks of size $k = 5$ invariant under a cyclic group of order 21 preserving a grid of type $3 \times 7$. To this end, we consider the group $\text{AGL}(1,3) \times \text{AGL}(1,7)$. The subgroup is generated by the map

$$(\tau_1, \tau_2), \mathbb{Z}_3 \times \mathbb{Z}_7 \to \mathbb{Z}_3 \times \mathbb{Z}_7,$$

where

$$\tau_1 : \mathbb{Z}_3 \to \mathbb{Z}_3, x \mapsto x + 1 \mod 3, \quad \tau_2 : \mathbb{Z}_7 \to \mathbb{Z}_7, y \mapsto y + 1 \mod 7.$$

With blocks of size 5, we cover 10 pairs each. The group of order 21 allows to cover each of the $210 = \binom{21}{2}$ pairs exactly once using a single orbit of a block. The question remains to construct all blocks and to classify the resulting designs. The Desarguesian plane $\text{PG}(2,4)$ provides a solution. The question is to decide whether there are any other, nonisomorphic designs. The following Orbiter commands can be used:

```bash
PP4=-d1.1-q1.3-d2.1-q2.7-K5-search_control-W-end-problem_label-PP4
PP4_GROUP1=-subgroup"1,1,1,1,"."21"-group_label."cyclic21"
PP4_MASK1=\n  -nb_orbits_on_blocks.1:\
  -depth.5:\
  -mask_label."no_mask"

DD_PP4:
  $(\text{ORBITER_PATH})\text{oriter.out}-.v.6-.\$
  $(\text{-Delandtsheer_Doyen-.\$(PP4)$-\$(PP4\_GROUP1)$-\$(PP4\_MASK1)$-\$)\$
  $(\text{-end-.})\$

DD_PP4_system:
  $(\text{ORBITER_PATH})\text{oriter.out}-.v.4-.\$
  $(\text{-define-\-diophant-\-label-PP4-\$} \$
  $(\text{-problem_of_Steiner_type.10-\PP4_pair_covering.csv-\$} \$
  $(\text{-has_sum.1-\$} \$
  $(\text{-end-\$} \$
  $(\text{-with-\-do-\$} \$
  $(\text{-diophant_activity-\solve_mckay-\$} \$
  $(\text{-end-\$} \$
```

324
The command DD_PP4 sets up the orbits of the group on pairs and writes the file PP4_pair_covering.csv. The command DD_PP4_system creates a diophantine system of Steiner type and solves it. It finds exaclty one solution. This must be the PG(2, 4) design. Since there are no more designs, isomorphism testing is not needed.
## 11.8 Tactical Decompositions

Table 11.5 lists the Orbiter commands for decomposition refinement.

Suppose we want to study projective planes of order 16. It is a linear space with $16^2 + 16 + 1 = 273$ points and equally many lines. Each point lies on 17 lines and each line contains 17 points. Any two points lie on exactly one line and any two lines intersect in exactly one point.

We decide to study maximal arcs of degree 4 in this plane (the degree has to divide the order of the plane). A maximal arc of degree $d$ is a set of points so that each line intersects in either $d$ or zero points. A line which intersects in $d$ points is called a secant. A line which intersects in no point is called an external line. The command

```bash
max_arc_16_4_start:
> $(ORBITER_PATH)orbiter.out -v 4 -maximal_arc_parameters 16 4
```

creates a decomposition stack for the parameters of the arc and writes the file `max_arc_q16_r4.stack`

```xml
<HTDO type=pt ptanz=2 btanz=2 fuse=simple>
  221 52
  52 17 0
  221 13 4
  1 1
</HTDO>
```

This is a point-tactical decomposition with 2 point-classes and 2 block-classes. The point classes are associated with the rows. The block-classes are associated with the columns. The first row and column indicates the size of the classes. The entries $a_{ij}$ count the number of blocks in the column class $j$ that are incident with a given point in the $i$th row class. The fuse information at the bottom (1 1) is a partition of the row classes which indicates the ancestor decomposition which was column tactical. The next step is to convert the stack file to a tdo file. The command

```bash
max_arc_16_4_convert_stack.tdo:
> $(ORBITER_PATH)orbiter.out -v 4 -convert_stack_to_tdo max_arc_q16_r4.stack
```

does that. It creates the file `max_arc_q16_r4.tdo`. It also prints the decomposition stack:

```
lambda_scheme at level 2 :
is 1 x 1
  | 273_{ 1}
  ===========
273_{ 0} |
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-lambda3</td>
<td>$\lambda_3 \ s$</td>
<td>Refine as 3-design with $\lambda_3$ and with block-size $s$</td>
</tr>
<tr>
<td>-solution</td>
<td>$i \ \text{fname}$</td>
<td>Use solutions to system $i$ from file fname.</td>
</tr>
<tr>
<td>-range</td>
<td>$f \ l$</td>
<td>Refine cases $i$ with $f \leq i &lt; f + l$ only.</td>
</tr>
<tr>
<td>-select</td>
<td>label</td>
<td>Select the case for refinement by label.</td>
</tr>
<tr>
<td>-o1</td>
<td>$s$</td>
<td>Omit $s$ variables from the first refinement system.</td>
</tr>
<tr>
<td>-o2</td>
<td>$s$</td>
<td>Omit $s$ variables from the second refinement system.</td>
</tr>
<tr>
<td>-D1_upper_bound_x0</td>
<td>$b$</td>
<td>Add the bound $x_0 \leq b$ in the first refinement.</td>
</tr>
<tr>
<td>-reverse</td>
<td></td>
<td>Sort the distributions in reverse order.</td>
</tr>
<tr>
<td>-reverse_inverse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-nopacking</td>
<td></td>
<td>Do not use packing inequalities.</td>
</tr>
<tr>
<td>-dual_is_linear_space</td>
<td></td>
<td>Assume that the dual incidence structure is a linear space also. This is valid for projective planes, for instance.</td>
</tr>
<tr>
<td>-geometric_test</td>
<td></td>
<td>Subject the distributions to the geometric test.</td>
</tr>
<tr>
<td>-once</td>
<td></td>
<td>Find at most one refinement in each case. This can be used to test which cases can be refined.</td>
</tr>
<tr>
<td>-mckay</td>
<td></td>
<td>Use McKay’s solver instead (by default, a lexicographic solver is used).</td>
</tr>
<tr>
<td>-input_file</td>
<td>fname</td>
<td>Specify the input TDO-file for refinement.</td>
</tr>
</tbody>
</table>

Table 11.5: TDO refinement options
Next, we can compute all coarsest column-tactical refinements of the decomposition. To this end, the command

\texttt{max\_arc\_16\_4\_refine:}

$\div (\text{ORBITER PATH})orbiter.out\cdot-v\cdot4\cdot-tdo\_refinement\backslash$

$\div \div -input\_file-max\_arc\_q16\_r4.tdo\cdot-dual\_is\_linear\_space\cdot-end$

is used. Because the incidence structure is a projective plane, the dual is a linear space also. Hence the option \texttt{-dual\_is\_linear\_space} can be used, which is helpful to reduce possibilities. As it turns out, there is exactly one refinement, and it is tactical. The file \texttt{max\_arc\_q16\_r4r.tdo} is produced. Note the added letter \texttt{r} at the end of the file name (\texttt{r} for refinement). We can use the following command to display the decomposition stack in the file:

\texttt{max\_arc\_16\_4r\_print:}

$\div (\text{ORBITER PATH})orbiter.out\cdot-v\cdot4\cdot-tdo\_print-max\_arc\_q16\_r4r.tdo\cdot$

This produces the following output:

decomposition 0.1:
lambda\_scheme at level 2 : is 1 x 1

------------------------
273\_{}{ 1} | 
------------------------
273\_{}{ 0} | 

row\_scheme at level 4 : is 2 x 2

------------------------
221\_{}{ 1} 52\_{}{ 2} | 
------------------------
52\_{}{ 0} | 17 0 
221\_{}{ 3} | 13 4 

328
col_scheme at level 4:
is 2 x 2

|   | 221_{ 1} | 52_{ 2} |
---|---------|---------|
52_{ 0} | 4       | 0       |
221_{ 3} | 13      | 17      |

extra_col_scheme at level 3:
is 1 x 2

|   | 221_{ 1} | 52_{ 2} |
---|---------|---------|
273_{ 0} | 17      | 17      |

Chapter 12

Finite Geometry

12.1 Spreads

A $t$-spread of $\text{PG}(n,q)$ is a set of disjoint $\text{PG}(t,q)$ that cover all of $\text{PG}(n,q)$ pointwise. $t$-spreads in $\text{PG}(n,q)$ exist if $t+1$ divides $n+1$. The reason is the existence of the Desarguesian spread (also called the regular spread). The Desarguesian spread is created from $\text{PG}(m - 1, Q)$ where $Q = q^s$ for some integer $s$. The spread elements are the subspaces which arise by considering the elements of $\text{PG}(m - 1, Q)$ as vector spaces over $\mathbb{F}_q$. As such, they are rank $s$ subspaces in $\text{PG}(n - 1, q)$. So, with $t = s - 1$, we have a $t$-spread in $\text{PG}(n - 1, q)$. The following command creates the Desarguesian line-spread in $\text{PG}(3, 2)$ (so $s = 2, t = s - 1 = 1, m = 2, q = 2$, and $Q = 4$):

\begin{verbatim}
desarguesian_spread_in_PG_3_2:
$($ORBITER\_PATH$)$orbiter.out\ -v\ 3\$
\-define\_FQ\ -finite\_field\ -q\ 4\ -end\$
\-define\_Fq\ -finite\_field\ -q\ 2\ -end\$
\-with\_FQ\ -and\_Fq\ -do\ -finite\_field\_activity\$
\-cheat\_sheet\_desarguesian\_spread\_2\ -end
pdflatex\_Desarguesian\_Spread\_3\_2.tex
open\_Desarguesian\_Spread\_3\_2.pdf
\end{verbatim}

The cheat sheet contains the following spread:

| Spread element 0 is $(1, 0)$ = | \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \end{bmatrix} |
| Spread element 1 is $(0, 1)$ = | \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \end{bmatrix} |
| Spread element 2 is $(1, 1)$ = | \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \end{bmatrix} |
Spread element 3 is \((2, 1) = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}_{17}\)

Spread element 4 is \((3, 1) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}_{22}\)

Spread elements by rank: \((0, 34, 9, 17, 22)\).

The following command creates the Desarguesian plane-spread in \(\text{PG}(5, 2)\):

desarguesian\_spread\_in\_PG\_5\_2:
▷ \$(\text{ORBITER\_PATH})\text{orbi}\text{te}r\_\text{out}.\text{-v.3.}\$
▷ ▷ -\text{define}\_F\text{Q}-\text{finite}\_\text{field}-q\_8\_\text{-end.}\$
▷ ▷ -\text{define}\_F\text{q}-\text{finite}\_\text{field}-q\_2\_\text{-end.}\$
▷ ▷ -\text{with}\_F\text{Q}-\text{and}\_F\text{q}-\text{do}-\text{finite}\_\text{field}\_\text{activity.}\$
▷ ▷ ▷ -\text{cheat}\_\text{sheet}\_\text{desarguesian}\_\text{spread}\_2\_\text{-end}
▷ pdflatex Desarguesian\_Spread\_5\_2.tex
▷ open Desarguesian\_Spread\_5\_2.pdf
Spread element 6 is \((5, 1) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \)

Spread element 7 is \((6, 1) = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \)

Spread element 8 is \((7, 1) = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \)

Spread elements by rank: \((0, 1394, 189, 671, 562, 1040, 792, 1161, 373)\)

Apart from the first spread element, the left halves of the generator matrices of the subspaces in the Desarguesian spread are the elements of \(\mathbb{F}_8\) in a matrix representation over \(\mathbb{F}_2\).

Two \(t\)-spreads are isomorphic if there is a collineation which maps one to the other. The classification problem for \(t\)-spreads is the problem of determining a complete set of pairwise non-isomorphic \(t\)-spreads. Orbiter can be used to classify spreads for small parameters. For instance, the command

```
spreads16_4:
  > $(ORBITER_PATH)orbiter.out -v 6 \\
  >   -orbiter_path$(ORBITER_PATH) \\
  >   -define F -finite_field -q 4 -end \\
  >   -define P -projective_space -3 F -end \\
  >   -with P -do \\
  >   -projective_space_activity \\
  >   -spread classify 2 -problem_label spreads_4_2 \\
  >     -W -depth 17 -draw poset \\
  >     -draw_options -radius 20 \\
  >     -nodes_empty -line_width 0.2 -embedded \\
  >     -end \\
  >   -report \\
  >   -end
```

classifies the line-spreads of \(\text{PG}(3,4)\) under the action of \(\text{PGL}(4,4)\). Under the André, Bruck-Bose construction \([3, 16]\), these spreads correspond to translation planes of order 16 with kernel \(\mathbb{F}_4\). Up to isomorphism, there are exactly three line-spreads in \(\text{PG}(3,4)\). They are the dearguesian spread, the Hall spread, and the semifield spread. Here is the relevant output taken from the latex report:
There are 3 orbits at level 17.

**Orbit 0 / 3 at Level 17**

Node number: 1126

\[
\{0, 25, 50, 75, 90, 107, 122, 140, 144, 157, 179, 204, 213, 238, 268, 334, 345\}_{1200}
\]

Strong generators for a group of order 1200:

\[
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
1 & \omega & 0 & 1 \\
\omega^2 & \omega^2 & \omega & 1
\end{bmatrix}_0,
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega & \omega^2 & 0 & 0 \\
\omega & \omega^2 & \omega & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}_1,
\begin{bmatrix}
\omega & 1 & \omega & \omega \\
\omega^2 & \omega^2 & \omega^2 & 0 \\
\omega & 0 & 0 & 1 \\
0 & \omega & \omega & 1
\end{bmatrix}_0
\]

1,0,0,0,0,1,0,0,2,3,0,2,1,1,3,2,0,
1,0,0,0,3,1,0,0,3,3,2,1,0,2,2,0,1,
1,3,1,1,2,2,0,1,0,0,3,0,1,1,3,0,
There are 0 extensions
Number of generators 3

**Orbit 1 / 3 at Level 17**

Node number: 1127

\[
\{0, 25, 50, 75, 90, 107, 140, 157, 179, 204, 213, 238, 265, 282, 299, 316, 356\}_{81600}
\]

Strong generators for a group of order 81600:

\[
\begin{bmatrix}
\omega & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_0,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \omega & \omega^2 \\
0 & 0 & 1 & 1
\end{bmatrix}_0,
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega & \omega^2 & 0 & 0 \\
\omega & \omega^2 & 1 & \omega^2 \\
0 & 0 & \omega & 1
\end{bmatrix}_0,
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega^2 & 0 & 0 & \omega \\
\omega^2 & 1 & \omega^2 & 1 \\
0 & 0 & \omega^2 & 1
\end{bmatrix}_1,
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega^2 & 0 & 0 & \omega \\
\omega^2 & 1 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_0
\]
1,0,0,0,0,1,0,0,0,0,3,0,0,0,0,3,0,  
1,0,0,0,0,1,0,0,0,0,2,3,0,0,1,1,0,  
1,0,0,0,0,1,0,0,2,1,3,1,2,3,2,2,0,  
1,0,0,0,3,1,0,0,0,0,1,0,0,3,2,1,  
1,0,0,3,3,1,2,1,0,0,2,0,0,0,1,2,1,  
0,1,1,0,2,0,1,1,0,0,2,1,0,0,0,2,0,  
0,0,0,1,0,0,2,1,0,1,2,3,2,1,1,1,0,  
There are 0 extensions  
Number of generators 7

**Orbit 2 / 3 at Level 17**

Node number: 1128

\{0, 25, 50, 75, 90, 108, 122, 140, 158, 183, 199, 217, 233, 250, 268, 312, 345\}_{576}

Strong generators for a group of order 576:

\[
\begin{bmatrix}
\omega & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}^{1},
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
\omega & 0 & \omega^1 & 0 \\
\omega^2 & 0 & 0 & 1
\end{bmatrix}^{0},
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega^2 & 0 & 0 & 0 \\
\omega & 0 & 1 & 1 \\
\omega^2 & 0 & \omega^2 & 1
\end{bmatrix}^{0}
\]

1,0,0,0,0,2,0,0,0,0,2,0,0,0,0,3,1,  
1,0,0,0,0,1,0,0,3,0,3,2,1,0,0,2,0,  
1,0,0,0,3,1,0,0,3,0,2,2,2,1,0,1,2,0,  
1,1,1,1,2,1,0,2,2,0,2,1,0,2,2,3,0,  
1,0,3,1,3,1,0,1,0,2,2,0,0,0,1,1,  
0,1,1,0,0,0,1,2,0,2,1,3,2,3,2,0,  
There are 0 extensions  
Number of generators 6
The three spreads in PG(3, 4) can be distinguished by their stabilizer orders. Table 12.1 lists the line spreads in PG(3, 4) according to their orbiter catalogue number (OCN). Table 12.2 lists the solid spreads in PG(7, 2) according to their orbiter catalogue number (OCN).
12.2 Translation Planes

Via the André, Bruck, Bose construction (cf. [3, 16]), spreads give rise to translation planes. The orbiter command

```
-Andre_Bruck_Bose_construction
```

constructs a projective plane from a spread. We rely on the catalogue of spreads contained in the knowledge base of Orbiter.

For instance, the command

```
TP_16_4:
```

```
|$\text{(ORBITER_PATH)orbiter.out:}-v.3:\$
|$\text{define}\,\text{F:}-\text{finite\_field:}-q.4-\text{end:}\$
|$\text{define}\,\text{PGGL}\,\text{linear\_group:}-\text{PGGL}\,\,\,\text{4.F:-end:}\$
|$\text{define}\,\text{PGGL}\,\text{5.linear\_group:}-\text{PGGL}\,\,\,\text{5.F:-end:}\$
|$\text{with}\,\text{PGGL}\,\,\text{4.and}\,\text{PGGL}\,\,\text{5.do:}\$
|$\text{-group\_theoretic\_activity:}\$
|$\text{Andre_Bruck_Bose_construction.0:"TP16-4-HALL".}\$
|$\text{-end}\$
|$\text{ORBITER_PATH)orbiter.out:}-v.2-\text{draw\_matrix:}\$
|$\text{\text{\text{define}\_csv\text{\_file:}-TP16-4\_HALL\_incma.csv:}\}
|$\text{\text{-box\_width:6-\text{bit\_depth:8-\text{partition:6.273.273.-end}}}
|$\text{open\_TP16-4\_HALL\_incma\_draw.bmp}
|$\text{pdflatex\_TP16-4-HALL\_report.tex}
|$\text{open\_TP16-4\_HALL\_report.pdf}
```

creates the Hall plane of order 16. Remember from Table 12.1 that the Hall spread has Orbiter Catalogue Number 0. The report lists the spread first, then the automorphism group of the plane and then the tactical decomposition of the incidence matrix:

```
The spread:
subspace 0 / 17 is 0:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\]

subspace 1 / 17 is 356:

\[
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

subspace 2 / 17 is 25:

```

337
<table>
<thead>
<tr>
<th>Subspace</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/17</td>
<td>50</td>
</tr>
<tr>
<td>4/17</td>
<td>75</td>
</tr>
<tr>
<td>5/17</td>
<td>97</td>
</tr>
<tr>
<td>6/17</td>
<td>114</td>
</tr>
<tr>
<td>7/17</td>
<td>127</td>
</tr>
<tr>
<td>8/17</td>
<td>153</td>
</tr>
<tr>
<td>9/17</td>
<td>179</td>
</tr>
<tr>
<td>10/17</td>
<td>191</td>
</tr>
</tbody>
</table>

- Subspace 3/17 is 50:
  \[ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \]

- Subspace 4/17 is 75:
  \[ \begin{bmatrix} 1 & 0 & \omega & 0 \\ 0 & 1 & 0 & \omega \end{bmatrix} \]

- Subspace 5/17 is 97:
  \[ \begin{bmatrix} 1 & 0 & \omega^2 & 0 \\ 0 & 1 & 0 & \omega^2 \end{bmatrix} \]

- Subspace 6/17 is 114:
  \[ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & \omega \end{bmatrix} \]

- Subspace 7/17 is 127:
  \[ \begin{bmatrix} 1 & 0 & \omega & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \]

- Subspace 8/17 is 153:
  \[ \begin{bmatrix} 1 & 0 & \omega^2 & 1 \\ 0 & 1 & 1 & \omega \end{bmatrix} \]

- Subspace 9/17 is 179:
  \[ \begin{bmatrix} 1 & 0 & 0 & \omega \\ 0 & 1 & \omega^2 & \omega \end{bmatrix} \]

- Subspace 10/17 is 191:
  \[ \begin{bmatrix} 1 & 0 & 1 & \omega \\ 0 & 1 & \omega & 0 \end{bmatrix} \]
subspace 11 / 17 is 224:
\[
\begin{bmatrix}
1 & 0 & \omega & \omega \\
0 & 1 & \omega & \omega^2
\end{bmatrix}
\]

subspace 12 / 17 is 236:
\[
\begin{bmatrix}
1 & 0 & \omega^2 & \omega \\
0 & 1 & 1 & 1
\end{bmatrix}
\]

subspace 13 / 17 is 262:
\[
\begin{bmatrix}
1 & 0 & 0 & \omega^2 \\
0 & 1 & \omega & \omega
\end{bmatrix}
\]

subspace 14 / 17 is 288:
\[
\begin{bmatrix}
1 & 0 & 1 & \omega^2 \\
0 & 1 & \omega^2 & \omega^2
\end{bmatrix}
\]

subspace 15 / 17 is 297:
\[
\begin{bmatrix}
1 & 0 & \omega & \omega^2 \\
0 & 1 & \omega^2 & 0
\end{bmatrix}
\]

subspace 16 / 17 is 322:
\[
\begin{bmatrix}
1 & 0 & \omega^2 & \omega^2 \\
0 & 1 & \omega^2 & 1
\end{bmatrix}
\]

Automorphism group:

Strong generators for a group of order 921600:

\[
\begin{bmatrix}
\omega & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & \omega \\
0 & 0 & 0 & 1
\end{bmatrix}_0, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_0, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_0
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & \omega & 0
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & \omega^2 & 0
\end{bmatrix}
\]
\[
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega & \omega^2 & 0 & 0 \\
0 & \omega^2 & \omega & 0 \\
\omega^2 & 0 & \omega^2 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & \omega & 0 & 0 \\
1 & 0 & \omega^2 & 0 \\
\omega & 1 & 1 & \omega^2
\end{bmatrix}
\begin{bmatrix}
1 & \omega^2 & \omega & 1 \\
0 & \omega^2 & 0 & 1 \\
0 & 1 & 0 & \omega^2 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,3,0,
1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,
1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,2,0,0,0,1,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,3,2,2,2,1,0,
1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,3,1,0,
1,0,0,0,0,0,1,0,0,0,0,2,1,3,0,1,1,3,3,0,0,0,0,0,1,0,1,0,0,0,0,0,1,0,0,0,0,1,1,3,2,1,0,0,3,0,1,0,2,1,1,3,0,0,1,0,3,0,0,0,0,0,1,1,

Tactical decomposition schemes:

<table>
<thead>
<tr>
<th></th>
<th>80₁</th>
<th>192₅</th>
<th>1₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>256₀</td>
<td>5</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>8₀³</td>
<td>16</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1₂²</td>
<td>0</td>
<td>16</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>80₁</th>
<th>192₅</th>
<th>1₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>256₀</td>
<td>16</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>8₀³</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1₂²</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>
12.3 Packings

A packing of $\text{PG}(3,q)$ is a set of pairwise line-disjoint spreads of $\text{PG}(3,q)$ of size $q^2 + q + 1$. Each spread contains $q^2 + 1$ lines. A simple counting argument shows that every line is contained in exactly one spread of the packing. The classification problem for packings is the problem of determining a complete set of pairwise non-isomorphic packings. Orbiter can be used to classify packings for small parameters. It is sometimes useful to make a symmetry assumption. This means that only those packings will be found that satisfy the symmetry assumption. The reason for making such an assumption is that the problem becomes easier and hence more tractable. Often, an assumption is made that the packings are invariant under a (nontrivial) group $H$. This section describes various ways in which Orbiter can help find and classify packings, with or without symmetry assumption.

Table 12.3 list Orbiter commands related to the construction of packings with assumed symmetry.

Table 12.4 list Orbiter commands related to the construction of packings with assumed symmetry by picking long orbits.

A packing is regular if it consists solely of regular spreads. The smallest regular packings exist in $\text{PG}(3,5)$. They were first described by Prince [56] and later placed into an infinite family by Penttila and Williams [53]. Up to isomorphism, there are exactly two regular packings in $\text{PG}(3,5)$. Let us construct these packings. We start by making a table of all regular packings:

```
spread_table_PG_3_5_regular:
▷ -mkdir SPREAD_TABLES_5_REG
▷ $(\text{ORBITER}\_\text{PATH})\text{orbiter.out} -v 6 \\
▷ ▷ -define F finite_field -q 5 -end \\
▷ ▷ -define P projective_space 3 F -end \\
▷ ▷ -define T spread_table P 2 "12" "SPREAD_TABLES_5_REG/".
▷ ▷ -print_symbols
```

There are 155,000 packings. In the command, we rely on the classification of spreads in $\text{PG}(3,5)$ which is built into Orbiter. The spread with orbiter catalogue number 12 is the regular spread.

We consider the projectivity of order 31 given by the matrix

$$
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 3 & 4 & 3 \\
0 & 3 & 4 & 3 \\
0 & 3 & 2 & 3
\end{bmatrix}
$$

The next command computes the normalizer of the cyclic subgroup of order 31 generated by this element:
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-H</td>
<td>description</td>
<td>Specify the assumed group $H$ of symmetries. The orbits of $H$ on the set of spreads are considered. The packings will be constructed as union of orbits.</td>
</tr>
<tr>
<td>-N</td>
<td>description</td>
<td>Specify the normalizer of $H$.</td>
</tr>
<tr>
<td>-cliques_on_fixpoint_graph $s$</td>
<td>s</td>
<td>Using poset classification, classify the orbits of $N$ on cliques of size $\leq s$ in the graph on fixed points.</td>
</tr>
<tr>
<td>-cliques_on_fixpoint_graph_control descr</td>
<td>descr</td>
<td>Specify poset classification options related to the classification of cliques on the fixed point graph as in Tables 6.2-6.3.</td>
</tr>
<tr>
<td>-fixp_clique_types_save_individually</td>
<td></td>
<td>Sort the cliques on fixed points by the type of their spreads and write one csv file for each possible type containing the index of the cliques of the given type.</td>
</tr>
<tr>
<td>-process_long_orbits descr</td>
<td>descr</td>
<td>Proceed on to long orbits using Table 12.4.</td>
</tr>
<tr>
<td>-spread_tables_prefix $P$</td>
<td>$P$</td>
<td>Use prefix $P$ to access spread tables.</td>
</tr>
<tr>
<td>-report</td>
<td></td>
<td>Create a report of the classification process.</td>
</tr>
<tr>
<td>-regular_packing</td>
<td></td>
<td>Initialize Klein correspondence and identify (regular) spreads with external lines to the Klein quadric using the polarity of the Klein quadric.</td>
</tr>
</tbody>
</table>

Table 12.3: Orbiter commands related to the construction of packings with assumed symmetry
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-list_of_cases_</td>
<td>fname</td>
<td>Define a subset of cases of fixed point cliques to be worked on. Only the cases listed the given file are considered.</td>
</tr>
<tr>
<td>from_file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-split</td>
<td>r m</td>
<td>Define a subset of cases of fixed point cliques to be worked on. Only those cases whose number is congruent to r modulo m are considered.</td>
</tr>
<tr>
<td>-orbit_length</td>
<td>l</td>
<td>Use orbits of length l.</td>
</tr>
<tr>
<td>-clique_size</td>
<td>s</td>
<td>Use exactly s orbits of length l.</td>
</tr>
<tr>
<td>-solution_path</td>
<td>P</td>
<td>Use P as a prefix for all solution files.</td>
</tr>
<tr>
<td>-create_graphs</td>
<td></td>
<td>For each case, create the graph that describes whether two orbits of length l are compatible.</td>
</tr>
<tr>
<td>-solve</td>
<td></td>
<td>Perform clique finding and write solutions to file.</td>
</tr>
<tr>
<td>-read_solutions</td>
<td></td>
<td>Read solutions from file.</td>
</tr>
</tbody>
</table>

Table 12.4: Orbiter commands related to the construction of packings with assumed symmetry related to picking long orbits
The normalizer is a group of order 372. We encode the group and its normalizer as makefile variables:

```
PGL_4_5_SUBGROUP_31_ME=-PGL_4_5:\ 
  -subgroup_by_generators:"31"."31.1":\ 
  "1,0,0,0,0,3,4,3,0,3,3,4,0,3,2,3":\ 
PGL_4_5_SUBGROUP_31_ME_NORMALIZER=-PGL_4_5:\ 
  -subgroup_by_generators:"normalizer_31"."372"."4":\ 
  "1,0,0,0,0,4,0,0,0,0,4,0,0,0,4":\ 
  "1,0,0,0,0,3,0,0,0,3,0,0,0,3":\ 
  "1,0,0,0,0,4,0,0,0,0,2,1,0,3,2,4":\ 
  "1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,3":.
```

Let $H$ be the subgroup of order 31 and let $N$ be its normalizer. Then we compute the orbits of $H$ on the regular spreads:

```
PG_3_5_assume_31_graph: 
  $(ORBITER_PATH)orbiter.out:-v:5:\ 
  -define:F:-finite_field:-q:5:-end:\ 
  -define:F:-projective_space:-3:F:-end:\ 
  -define:T:-spread_table:P:2:"12"."SPREAD_TABLES_5_REG/":\ 
  -define:PW:-packing_with_symmetry_assumption:T:\ 
  -H:"H31".$(PGL_4_5_SUBGROUP_31_ME):-end:\ 
  -N:"N31".$(PGL_4_5_SUBGROUP_31_ME_NORMALIZER):-end:\ 
  -end:\ 
  -define:PWF:-packing_choose_fixed_points:PW:0:-end:\ 
  -define:L:-packing_long_orbits:PW:0:-end:\ 
  -orbit_length:31:-clique_size:1:-create_graphs:-end:\ 
  -print_symbols 
  pdflatex:H31_reduced_spread_orbits_orbits_report.tex 
  open:H31_reduced_spread_orbits_orbits_report.pdf 
  pdflatex:H31_line_orbits_orbits_report.tex
```
The command produces reports about the orbits of both $H$ and $N$ on points, lines and spreads. The following command searches all cliques of size 1 in the graph on long orbits. This is not very difficult!

```plaintext
PG_3_5_assume_31_fpc0_lo_cliques:
\$ (ORBITER_PATH)orbiner.out\$v.2:
\$ G
\$ do:
\$ -graph.theorectic_activity
\$ find_cliques
\$ print_symbols
```

There are exactly 8 cliques of size 1. The next command builds the packings arising from these 8 cliques:

```plaintext
PG_3_5_assume_31_read:
\$ (ORBITER_PATH)orbiner.out\$v.5:
\$ F
\$ P
\$ T
\$ PW
\$ H
\$ N
\$ PWF
\$ L
\$ orbit_length 31
\$ read_solutions:
```

The next command classifies the 8 packings up to isomorphism, using Nauty:

```plaintext
PG_3_5_assume_31_classify:
\$ (ORBITER_PATH)orbiner.out\$v.2:
```
There are exactly 2 isomorphism classes of packings. These are of course the examples found by Prince and generalized by Penttila and Williams. The packings are invariant under a group of order 93.
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-catalogue</td>
<td>$i$</td>
<td>Create BLT-set number $i$ from the Orbiter catalogue ($i$ is zero-based).</td>
</tr>
<tr>
<td>-family</td>
<td>$F$</td>
<td>Create a BLT-set from family $F$. See Table 12.6 for possibilities for $F$.</td>
</tr>
</tbody>
</table>

Table 12.5: Commands for creating BLT-sets

### 12.4 BLT-Sets

A BLT-set of $Q(4, q)$ is a set of $q + 1$ point on the quadric such that no point on the quadric is collinear to more than two points of the set. BLT sets are related to spreads of $PG(3, q)$, to flocks of the quadratic cone in $PG(3, q)$, and to many other objects in combinatorics and finite geometry. They exist whenever $q$ is odd. BLT-sets have been defined in [4]. It is an interesting problem to classify BLT-sets of $Q(4, q)$ under the orthogonal group. Some references are Law [41], Penttila-Royle [52], Penttila-Law [42, 43], Betten [9], AlAzemi-Betten-Chowdhury [1].

Orbiter can be used to create members of known families of BLT-sets and sets from a catalogue of BLT-sets over small fields. Besides that, Orbiter can be used to classify all BLT-sets for a given value of $q$. We will see how we create known examples of BLT-sets either from the catalogue or from known families. Afterwards, we will consider the problem of classification.

Table 12.5 shows options to create known BLT-sets. Table 12.6 shows options for known families or sporadic sets. For instance, the command

**BLT_11.0:**

```latex
\$\text{(ORBITER\_PATH)}\text{orbiter.out}\,-v\,-2\,\$
\$-\text{define}\,F\,-\text{finite\_field}\,-q\,11\,-\text{end}\,\$
\$-\text{define}\,O\,-\text{orthogonal\_space}\,-0\,5\,F\,-\text{end}\,\$
\$-\text{with}\,O\,-\text{do}\,-\text{orthogonal\_space\_activity}\,\$
\$-\text{create}\_\text{BLT\_set}\,-\text{-catalogue}\,0\,-\text{end}\,\$
\$-\text{end}\$
\$\text{\#pdflatex}\,0\,1\,6\,2\,\text{report.tex}\,\$
\$\text{\#open}\,0\,1\,6\,2\,\text{report.pdf}\$
```

creates the BLT-set #0 in $Q(4, 11)$. The command

**BLT_11_Mondello:**

```latex
\$\text{(ORBITER\_PATH)}\text{orbiter.out}\,-v\,-2\,\$
\$-\text{define}\,F\,-\text{finite\_field}\,-q\,11\,-\text{end}\,\$
\$-\text{define}\,O\,-\text{orthogonal\_space}\,-0\,5\,F\,-\text{end}\,\$
\$-\text{with}\,O\,-\text{do}\,-\text{orthogonal\_space\_activity}\,\$
```

347
<table>
<thead>
<tr>
<th>$F$</th>
<th>Condition</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td></td>
<td>Linear BLT-set.</td>
</tr>
<tr>
<td>Fisher</td>
<td></td>
<td>Fisher BLT-set [26].</td>
</tr>
<tr>
<td>Mondello</td>
<td>$q \equiv \pm 1$ mod 10</td>
<td>Mondello BLT-set due to Penttila [51].</td>
</tr>
<tr>
<td>FTWKB</td>
<td>$q \equiv \pm 2$ mod 3</td>
<td>Fisher, Thas, Walker [65], Kantor, Betten [13] BLT-set.</td>
</tr>
<tr>
<td>Kantor1</td>
<td></td>
<td>Kantor’s first family.</td>
</tr>
<tr>
<td>Kantor2</td>
<td>$q \equiv \pm 2$ mod 5</td>
<td>Kantor’s second family.</td>
</tr>
<tr>
<td>LP_37_72</td>
<td>$q = 37$</td>
<td>BLT-set for $q = 37$ with ago=72 due to Law and Penttila [43].</td>
</tr>
<tr>
<td>LP_37_41a</td>
<td>$q = 37$</td>
<td>First BLT-set for $q = 37$ with ago=4, due to Law and Penttila [43].</td>
</tr>
<tr>
<td>LP_37_41b</td>
<td>$q = 37$</td>
<td>Second BLT-set for $q = 37$ with ago=4, due to Law and Penttila [43].</td>
</tr>
<tr>
<td>LP_71</td>
<td>$q = 71$</td>
<td>BLT-set for $q = 71$ due to Law and Penttila [43].</td>
</tr>
</tbody>
</table>

Table 12.6: Families of BLT-sets

\[ X_0^2 + X_1X_2 + X_3X_4 = 0 \]

The BLT-set is:
\[
\begin{array}{|c|c|c|c|}
\hline
i & \text{Rank} & \text{Point} & (a, b, c) \\
\hline
0 & 846 & (1, 6, 4, 10, 3) & (22, 11, 1) \\
1 & 851 & (1, 5, 7, 10, 3) & (22, 110, 1) \\
2 & 1234 & (1, 5, 1, 7, 7) & (37, 11, 1) \\
3 & 613 & (1, 6, 10, 5, 1) & (73, 110, 1) \\
4 & 1307 & (1, 1, 3, 8, 5) & (59, 36, 1) \\
5 & 1418 & (1, 3, 9, 6, 10) & (95, 36, 1) \\
6 & 1022 & (1, 9, 5, 10, 2) & (99, 96, 1) \\
7 & 835 & (1, 2, 6, 3, 3) & (99, 36, 1) \\
8 & 950 & (1, 10, 8, 2, 9) & (95, 96, 1) \\
9 & 789 & (1, 8, 2, 4, 4) & (59, 36, 1) \\
10 & 611 & (1, 7, 7, 5, 1) & (73, 11, 1) \\
11 & 1236 & (1, 4, 4, 7, 7) & (37, 110, 1) \\
\hline
\end{array}
\]

Plane intersection type is $4^{18} 3^{148}$
Plane invariant is too big (18 planes)

\[
\begin{array}{c|c|c}
\rightarrow & 18_1 & \downarrow \\
12_0 & 6 & 12_0 \\
\end{array}
\]

\[
C_0 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}_{12}
\]

\[
C_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}_{18}
\]

\[
\begin{array}{c|c|c}
\rightarrow & 18_1 & \downarrow \\
12_0 & 6 & 12_0 \\
\end{array}
\]

\[
C_0 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}_{12}
\]

\[
C_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}_{18}
\]

The classification of BLT-sets proceeds via the poset of partial BLT-sets. For more details, see [1, 9, 41]. The following command classifies the BLT-sets in $Q(4, 13)$:

\textbf{BLT\_13\_deep\_14:}
- $(\text{ORBITER\_PATH})$orbiter.out.-v.2.\$
- \text{-define:\ M\_finite_field\_q.13.-end\$
- \text{-define:\ M\_orthogonal_space\_0.5-F.-end\$
- \text{-with:\ M\_do\_orthogonal_space\_activity\$
- \text{-BLT\_set\_starter\_14.-problem_label\_BLT\_q13.-W.-depth\_14.-end\$

349
Chapter 13

Graph Theory

13.1 Creating Graphs

Table 13.1 shows some Orbiter commands to create graphs.

For instance, the command

```
Cycle_13:
  $(ORBITER_PATH)orbiter.out.-v.2.
  -define Gamma.-graph.
  -cycle 13
  -end
```

creates the cycle graph of degree 13.

There are many ways to read graphs from file. One way is by means of an adjacency matrix stored as a csv file. Consider an example. The `-load_csv_no_border` command can be used to create a graph from a csv file containing the adjacency matrix. Here is the command:

```
TRIANGLE_GRAPH="0,1,1
               n1,0,1
               n1,1,0"

triangle_graph:
  echo $(TRIANGLE_GRAPH) >triangle_graph.csv
  $(ORBITER_PATH)orbiter.out.-v.6.
  -define G.-graph.
  -load_csv_no_border.
  triangle_graph.csv.
  -end
```

This will create the three-cycle graph.

The command
<table>
<thead>
<tr>
<th>Key</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-load</td>
<td>filename</td>
<td>Read a graph from file in Orbiter format.</td>
</tr>
<tr>
<td>-load_csv_no_border</td>
<td>filename</td>
<td>Read a graph from a csv file. Ignore the first row and first column.</td>
</tr>
<tr>
<td>-load_dimacs</td>
<td>filename</td>
<td>Read a graph from file in dimacs format.</td>
</tr>
<tr>
<td>-edge_list</td>
<td>n list-of-edges</td>
<td>Create a graph on n vertices from a list of edges as ranked pairs.</td>
</tr>
<tr>
<td>-edges_as_pairs</td>
<td>n edges-as-pairs</td>
<td>Create a graph on n vertices from a list of edges as pairs.</td>
</tr>
<tr>
<td>-cycle</td>
<td>n</td>
<td>Cycle graph on n vertices.</td>
</tr>
<tr>
<td>-Hamming</td>
<td>n q</td>
<td>Hamming graph $H(n,q)$</td>
</tr>
<tr>
<td>-Johnson</td>
<td>n k s</td>
<td>Johnson graph</td>
</tr>
<tr>
<td>-Paley</td>
<td>q</td>
<td>Paley graph</td>
</tr>
<tr>
<td>-Sarnak</td>
<td>p q</td>
<td>Lubotzky-Phillips-Sarnak graph [45]</td>
</tr>
<tr>
<td>-Schlaefli</td>
<td>q</td>
<td>Schlaefli graph</td>
</tr>
<tr>
<td>-Shrikhande</td>
<td></td>
<td>Shrikhande graph</td>
</tr>
<tr>
<td>-Winnie_Li</td>
<td>q i</td>
<td>Winnie-Li graph [44]</td>
</tr>
<tr>
<td>-Grassmann</td>
<td>n k q r</td>
<td>Grassmann graph</td>
</tr>
<tr>
<td>-coll_orthogonal</td>
<td>$\epsilon$ d q</td>
<td>Collinearity graph of $O^\epsilon(d,q)$</td>
</tr>
<tr>
<td>-triherald_pair_disjointness_graph</td>
<td></td>
<td>Triheral pair disjointness graph</td>
</tr>
<tr>
<td>-non_attacking_queens_graph</td>
<td>n</td>
<td>Create the graph for non-attacking queens on a $n \times n$ chess board.</td>
</tr>
<tr>
<td>-subset</td>
<td>label labeltex subset</td>
<td>Define vertex coloring with two colors based on a subset of vertices.</td>
</tr>
<tr>
<td>-disjoint_sets_graph</td>
<td>fname</td>
<td>Define a graph on a system of sets. Adjacency is when two sets are disjoint. The sets are taken from the given file.</td>
</tr>
<tr>
<td>-orbital_graph</td>
<td>$G$ i</td>
<td>Define orbital graph from the $i$-th orbit of the group $G$ acting on pairs.</td>
</tr>
<tr>
<td>-collinearity_graph</td>
<td>inc-matrix</td>
<td>Collinearity graph of the given incidence matrix.</td>
</tr>
<tr>
<td>-chain_graph</td>
<td>P1 P2</td>
<td>Chain graph with respect to the partitions P1 and P2.</td>
</tr>
<tr>
<td>-Cayley_graph</td>
<td>$G$ gens</td>
<td>Cayley graph with respect to group $G$ and generating set gens.</td>
</tr>
</tbody>
</table>

Table 13.1: Orbiter commands to define graphs

352
Chain_232:

- $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot v.2\$
- $-\text{define}\cdot P1\cdot-\text{vector}\cdot-\text{dense}\cdot 2,3,2\cdot-\text{end}\$
- $-\text{define}\cdot P2\cdot-\text{vector}\cdot-\text{dense}\cdot 2,3,2\cdot-\text{end}\$
- $-\text{define}\cdot \Gamma\cdot-\text{graph}\$
- $-\text{chain}\cdot\text{graph}\cdot P1\cdot P2\$
- $-\text{end}$

creates the chain graph with respect to the partitions $(2, 3, 2)$ and $(2, 3, 2)$.

The command

Paley_13_graph:

- $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot v.2\$
- $-\text{define}\cdot \Gamma\cdot-\text{graph}\cdot-\text{Paley}\cdot 13\cdot-\text{end}\$

creates the Paley graph on 13 vertices.

The command

trihedral_pair_graph:

- $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot v.2\$
- $-\text{define}\cdot \Gamma\cdot-\text{graph}\$
- $-\text{trihedral}\cdot\text{pair}\cdot\text{disjointness}\cdot\text{graph}\$
- $-\text{end}$

creates the graph of trihedral pairs. Two vertices are adjacent if the associated trihedral pairs are line-disjoint.

The command

small_graph:

- $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot v.2\$
- $-\text{define}\cdot G\cdot-\text{graph}\cdot-\text{edges}\cdot\text{as}\cdot\text{pairs}\cdot 5\cdot"0,1,0,2,0,3,0,4,1,3,1,4,2,4"\cdot-\text{end}$
creates a graph by listing the edges in pairs. In this case, the graph

![Graph Diagram]

is created.

The command

```bash
petersen:
  ▶ $(ORBITER_PATH)orriter.out -v.2-
  ▶ ▶ -define G:-graph-Johnson 5 2 0 -end
```

creates the Petersen graph.

The command

```bash
Johnson_6_2_0:
  ▶ $(ORBITER_PATH)orriter.out -v.2-
  ▶ ▶ -define G:-graph-Johnson 6 2 0 -end
```

creates the Johnson graph $J(6, 2, 0)$.

The command

```bash
Hamming_graph_3:
  ▶ $(ORBITER_PATH)orriter.out -v.2-
  ▶ ▶ -define G:-graph-Hamming 3 2 -end
```

creates the Hamming graph of order 3.

There is a graph on 315 vertices that arises from the Cohen-Tits near octagon (see [15]). The graph was first constructed in [18] and has automorphism group equal to $\text{Aut}(HJ)$, the automorphism group of the Higman-Sims sporadic simple group. The graph is distance-regular. An incidence matrix can be found in Ascii format on the web site [6]. In the following, we assume that a file `halljanko315.csv` is present, containing the incidence matrix of the graph. The following command creates the graph from the file:
<table>
<thead>
<tr>
<th>Key</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-complement</td>
<td></td>
<td>Complementary graph.</td>
</tr>
<tr>
<td>-distance_2</td>
<td></td>
<td>Distance two graph: Two vertices are adjacent if and only if they have distance two in the original graph.</td>
</tr>
</tbody>
</table>

Table 13.2: Orbiter commands to modify graphs

HJ_graph:

```
▷ $(ORBITER_PATH)orbiter.out:-v.6\n▷ ▷ -define:G:-graph\n▷ ▷ ▷ -load_csv_no_border\n▷ ▷ ▷ halljanko315.csv\n▷ ▷ -end
```

In Section 15.7, we will compute the automorphism group of the graph (of order 1209600). This will create a file `halljanko315_gens.csv` which we use here to create an orbital graph. An orbital graph is a graph whose adjacency matrix corresponds to an orbit of a permutation group in the action on pairs. The group is the automorphism group of the graph. The following command creates the third orbital graph:

```
HJ315_orbital_graph_3:
▷ $(ORBITER_PATH)orbiter.out:-v.2\n▷ ▷ -define:gens:-vector:-file\n▷ ▷ ▷ halljanko315_gens.csv:-end\n▷ ▷ -define:G:-permutation_group\n▷ ▷ ▷ -bsgs:halljanko315:"File\halljanko315"\n▷ ▷ ▷ 315:1209600:"0,1,2".6:gens\n▷ ▷ -end\n▷ ▷ -define:Gamma:-graph\n▷ ▷ ▷ -orbital_graph:G:3\n▷ ▷ -end\n```

Table 13.2 shows some Orbiter commands to modify graphs. The commands replace the given graph by the graph obtained after applying the specified modification.

For a graph $\Gamma$, the distance 2 graph $\Delta$ has the same vertices as $\Gamma$, and two vertices in $\Delta$ are adjacent if and only if the distance in $\Gamma$ is two. The following command creates the distance 2 graph of the Cohen-Tits graph.

HJ_d2_graph:

```
▷ $(ORBITER_PATH)orbiter.out:-v.6\n```

355
Let us look at some examples of Cayley graphs. The first graph has $G = \mathbb{Z}_{11}$ and connection set all elements congruent 1 mod 3. We create the group as a subgroup of the one-dimensional affine group over $F_{11}$. This means that the map $x \mapsto ax + b \mod 11$ is encoded as a pair $(a, b)$.

Cayley_Z11_1mod3:
\begin{verbatim}
▷ $(ORBITER_PATH)orbiter.out-v.2\$
▷ -define:F--finite_field--q.11--end\$
▷ -define:S--vector--dense\$
▷ "1,1,1,4,1,7,1,10"--end\$
▷ -define:G--linear_group--AGL.1_F\$
▷ -subgroup_by_generators:"Z11":11:1:"1,1"\$
▷ -end\$
▷ -define:Gamma--graph\$
▷ -Cayley_graph:G:S\$
▷ -end.
\end{verbatim}

The vertices of the Cayley graph are ordered. The ordering is determined by the stabilizer chain. This is a total ordering.

In the following example, we create a Cayley graph based on the symmetric group on 4 things. We take the Coxeter generators as connection set:

Cayley_Sym4_coxeter:
\begin{verbatim}
▷ $(ORBITER_PATH)orbiter.out-v.2\$
▷ -define:S--vector--dense:"1,0,2,3,0,2,1,3,0,1,3,2"--end\$
▷ -define:G--permutation_group--symmetric_group:4\$
▷ -end\$
▷ -define:Gamma--graph\$
▷ -Cayley_graph:G:S\$
▷ -end.
\end{verbatim}

And one more example, using the same group but a different connection set. This graph is called the star graph:

Cayley_Sym4_star:
\begin{verbatim}
▷ $(ORBITER_PATH)orbiter.out-v.2\$
\end{verbatim}
\begin{verbatim}
\define S \vector\dense"1,0,2,3,\cdot,2,1,0,3,\cdot,3,1,2,0"\end{verbatim}

\begin{verbatim}
\define G \permutation\group \symmetric\group 4 \end{verbatim}

\begin{verbatim}
\define \Gamma \graph \Cayley\graph G.S \end{verbatim}

\begin{verbatim}
\end{verbatim}
<table>
<thead>
<tr>
<th>Key</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-find_cliques</td>
<td>options</td>
<td>Find all cliques. See Section 13.4.</td>
</tr>
<tr>
<td>-export_magma</td>
<td></td>
<td>Export to Magma [14].</td>
</tr>
<tr>
<td>-export_maple</td>
<td></td>
<td>Export to Maple [48].</td>
</tr>
<tr>
<td>-export_csv</td>
<td></td>
<td>Export to csv-file.</td>
</tr>
<tr>
<td>-export_graphviz</td>
<td></td>
<td>Export to graphviz-file.</td>
</tr>
<tr>
<td>-print</td>
<td></td>
<td>Print the graph.</td>
</tr>
<tr>
<td>-sort_by_colors</td>
<td></td>
<td>Sort the vertices by color classes.</td>
</tr>
<tr>
<td>-split</td>
<td>file</td>
<td>Split the graph into subgraphs.</td>
</tr>
<tr>
<td>-split_by_starters</td>
<td>file</td>
<td>Split the graph into subgraphs according to starters.</td>
</tr>
<tr>
<td>-split_by_clique</td>
<td>label clique</td>
<td>Compute the neighborhood graph of the given clique.</td>
</tr>
<tr>
<td>-save</td>
<td></td>
<td>Save the graph to file in binary format.</td>
</tr>
<tr>
<td>-automorphism_group</td>
<td></td>
<td>Compute the automorphism group and write a report. See Section 15.7.</td>
</tr>
<tr>
<td>-properties</td>
<td></td>
<td>Compute properties of the graph.</td>
</tr>
<tr>
<td>-eigenvalues</td>
<td></td>
<td>Compute the eigenvalues of the graph.</td>
</tr>
<tr>
<td>-draw</td>
<td></td>
<td>Draw the graph.</td>
</tr>
</tbody>
</table>

Table 13.3: Graph Theoretic Activities

13.2 Graph Theoretic Activities

Graph theoretic activities allow us to perform tasks on graphs. Table 13.3 shows the commands for graph theoretic activities. These are activities that can be applied to objects of type graph.

Continuing the example of the three-cycle, the command

```bash
triangle_graph_properties:
  echo $(TRIANGLE_GRAPH)>triangle_graph.csv
  $(ORBITER_PATH)orbiter.out.-v.6\`
  -define G:-graph-
  -load_csv_no_border-
  triangle_graph.csv-
  -end-
  -with G:-do-
  -graph_theoretic_activity:-properties-
  -end
```

358
computes the degree type of the graph. This is the distribution of degrees in the degree sequence of the graph. It prints the distribution of degree values in exponential notation. Multiplicities are indicated as exponent. Since there are three vertices of degree 2, the degree sequence is printed as $2^3$.

We can export the adjacency matrix and create a bitmap drawing like so:

```
Cycle_13.draw:
  > $(ORBITER_PATH)orbiter.out-v.2-
  > -define Gamma -graph -cycle 13 -end-
  > -with Gamma -do-
  > -graph_theoretic_activity -export_csv -end-
  > -with Gamma -do-
  > -graph_theoretic_activity -export_graphviz -end
  > $(ORBITER_PATH)orbiter.out-v.2 -draw_matrix-
  > -input_csv_file Cycle_13.csv-
  > -box_width 20 -bit_depth 8 -partition 4 13 13 -end
  > #dot -Tpng Cycle_13.gv > Cycle_13.png
  > #twopi -Tpng Cycle_13.gv > Cycle_13.png
  > #open Cycle_13_draw.bmp
  > #pdflatex Cycle_13_report.tex
  > #open Cycle_13_report.pdf
```

The command first creates the cycle graph of order 13, and then exports the adjacency matrix as csv file. It then draws the adjacency matrix as a bitmap graphics.

Suppose we want to compute the eigenvalues of the adjacency matrix of a graph. The command `-eigenvalues` can help:

```
Cycle_9_eigenvalues:
  > $(ORBITER_PATH)orbiter.out-v.2-
  > -define Gamma -graph-
  > -cycle 9-
  > -with Gamma -do-
  > -graph_theoretic_activity -eigenvalues -end
  > pdflatex Cycle_9_eigenvalues.tex
  > open Cycle_9_eigenvalues.pdf
```

computes the eigenvalues (both regular and Laplace) of the 9-cycle. The following output is produced:
The energy is 11.5175
Eigenvalues: \( \lambda_i \)
Laplace eigenvalues: \( \theta_i \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( \lambda_i )</th>
<th>( \theta_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>3.87939</td>
</tr>
<tr>
<td>1</td>
<td>1.53209</td>
<td>3.87939</td>
</tr>
<tr>
<td>2</td>
<td>1.53209</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0.347296</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0.347296</td>
<td>1.6527</td>
</tr>
<tr>
<td>5</td>
<td>−1</td>
<td>1.6527</td>
</tr>
<tr>
<td>6</td>
<td>−1</td>
<td>0.467911</td>
</tr>
<tr>
<td>7</td>
<td>−1.87939</td>
<td>0.467911</td>
</tr>
<tr>
<td>8</td>
<td>−1.87939</td>
<td>−2.26243e−16</td>
</tr>
</tbody>
</table>

The command

\texttt{petersen:}

\texttt{\$(ORBITER\_PATH)orbiter.out\:−v\:2\:\}
\texttt{\:\:−define\:G\:\:−graph\:−Johnson\:5\:2\:0\:−end\:\}
\texttt{\:\:−with\:G\:−do\:\}
\texttt{\:\:−graph\:theoretic\:activity\:−export\:csv\:−end\:\}
\texttt{\:\:−with\:G\:−do\:\}
\texttt{\:\:−graph\:theoretic\:activity\:−export\:graphviz\:−end\:\}
\texttt{\:\:−with\:G\:−do\:\}
\texttt{\:\:−graph\:theoretic\:activity\:−save\:−end
\texttt{\$(ORBITER\_PATH)orbiter.out\:−v\:2\:−draw\:matrix\:\}
\texttt{\:\:−input\:csv\:file\:Johnson\:5\:2\:0\:csv\:\}
\texttt{\:\:−box\:width\:40\:−bit\:depth\:24\:−partition\:4\:"10"\:"10"\:−end}
\texttt{\:\:do\:\:−Tpng\:Johnson\:5\:2\:0.gv\:\:>\:Johnson\:5\:2\:0.png}

creates the Johnson graph \( J(5, 2, 0) \) also known as the Petersen graph.

The command

\texttt{small\_graph\_draw:}

\texttt{\$(ORBITER\_PATH)orbiter.out\:−v\:2\:\}
\texttt{\:\:−define\:G\:\:−graph\:−edges\:_as\:_pairs\:5\:"0,1,0,2,0,3,0,4,1,3,1,4,2,4"\:−end\:\}
\texttt{\:\:−with\:G\:−do\:\}
\texttt{\:\:−graph\:theoretic\:activity\:−export\:csv\:−end\:\}
draws the small graph created in Section 13.1 using the external tool graphviz.

It is possible to create the collinearity graph of an incidence structure. The incidence structure must be encoded by means of an incidence matrix. Let us continue an example from Section 4.7, where the incidence matrix of $Q(4,2)$ was created. At that time, we wrote the incidence matrix to file. Here, we read the incidence matrix from file and create the collinearity graph of it:

```
PG0_5_2_collinearity_graph:0_5_2_incidence_matrix.csv
```

The command also computes properties of the graph. The graph has 15 vertices of degree 6. This is because in the geometry, each point lies on three lines, and hence is collinear with 6 other points.

Let is consider again the Cayley graphs from Section 13.1. Here is a command that draws the first graph:

```
Cayley_Z11_1mod3_draw:
```

361
The drawing is shown in Figure 13.1. Let us draw the Cayley graph of Sym(5) with respect to the Coxeter generators:

```
Cayley_Sym5_coxeter_draw:
  $(ORBITER_PATH)orbiter.out-v.2\n  -draw_options-xin:1000000-yin:1000000\n  -embedded-radius:10000-nodes_empty\n  -define:S:vector:dense\n  -define:G:permutation_group:symmetric_group:5\n  -end\n  -define:Gamma:graph\n  -Cayley_graph:G:S\n  -end\n  -with:Gamma:do\n  -graph_theoretic_activity:draw\n  pdflatex:Cayley_graph_AGL.1_11_draw.tex
  open:Cayley_graph_AGL.1_11_draw.pdf
```

Figure 13.1: The Cayley graph in $\mathbb{Z}_{11}$
Figure 13.2: The Cayley graph of $\text{Sym}(5)$ w.r.t. the Coxeter generators

The drawing is shown in Figure 13.2.
<table>
<thead>
<tr>
<th>Option</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-girth</td>
<td>$d$</td>
<td>Girth at least $d$</td>
</tr>
<tr>
<td>-regular</td>
<td>$r$</td>
<td>Regular of degree $r$</td>
</tr>
<tr>
<td>-no_transmitter</td>
<td></td>
<td>Tournament without transmitter (requires -tournament)</td>
</tr>
</tbody>
</table>

Table 13.4: Options for classifying graphs

### 13.3 Classification of Graphs and Tournaments

Table 13.4 lists the Orbiter commands to classify graphs and tournaments. The following command classifies all graphs on 4 vertices:

```bash
graph_classify_4:
  ▶ $(ORBITER_PATH)orbiter.out -v 2
  ▶ ▶ -define GC -graph_classification
  ▶ ▶ ▶ -n 4
  ▶ ▶ ▶ -poset_classification_control
  ▶ ▶ ▶ ▶ -problem_label:graphs_v4 -depth 6 -draw_poset
  ▶ ▶ ▶ ▶ -draw_options: -radius 250 -embedded -end
  ▶ ▶ ▶ ▶ -end
  ▶ ▶ ▶ -end
  ▶ ▶ -with GC -do
  ▶ ▶ -graph_classification_activity
  ▶ ▶ ▶ -draw_graphs_at_level 3
  ▶ ▶ ▶ -end
  ▶ ▶ -print_symbols
  ▶ pdflatex:graphs_v4_rep_3_2.tex
  ▶ open:graphs_v4_rep_3_2.pdf
  ▶ #pdflatex:graphs_v4_poset_detailed_lvl_6.tex
  ▶ #open:graphs_v4_poset_detailed_lvl_6.pdf
  ▶ #pdflatex:graphs_v4_poset_lvl_6.tex
  ▶ #open:graphs_v4_poset_lvl_6.pdf
```

The next command classifies all tournaments on 4 vertices:

```bash
tournament_classify_4:
  ▶ $(ORBITER_PATH)orbiter.out -v 2
  ▶ ▶ -define GC -graph_classification
  ▶ ▶ ▶ -n 4 -tournament
```

364
Figure 13.3: The four isomorphism types of tournaments on 4 vertices

```
$>$ $>$ $>$ -poset_classification_control.\$
$>$ $>$ $>$ $>$ -problem_label.tournament_4..-depth.6..-draw_poset.\$
$>$ $>$ $>$ $>$ -draw_options.-radius.250.-embedded.-end.\$
$>$ $>$ $>$ -end.\$
$>$ $>$ -end.\$
$>$ $>$ -with.GC..do.\$
$>$ $>$ -graph_classification_activity.\$
$>$ $>$ $>$ -draw_options.-embedded.-radius.400.\$
$>$ $>$ $>$ -line_width.2.-scale.0.15.-end.\$
$>$ $>$ -draw_graphs_at_level.6.\$
$>$ $>$ -end.\$
$>$ $>$ -print_symbols
```

Figure 13.3 shows the resulting list of 4 tournaments.

The next command classifies all cubic graphs on 8 vertices:

```
graph_classify_8_r3:
$>$ $(ORBITER\_PATH)orbiter.out.-v.3.\$
$>$ $>$ -define.GC.-graph_classification.\$
$>$ $>$ $>$ -n.8.-regular.3.\$
$>$ $>$ $>$ -poset_classification_control.\$
$>$ $>$ $>$ $>$ -problem_label.graphs_v8_r3..-depth.12..-draw_poset.\$
$>$ $>$ $>$ $>$ -draw_options.-radius.250.\$
$>$ $>$ $>$ $>$ $>$ -line_width.0.2.-embedded.-end.\$
$>$ $>$ $>$ -end.\$
$>$ $>$ -end.\$
$>$ $>$ -with.GC..do.\$
$>$ $>$ -graph_classification_activity.\$
$>$ $>$ $>$ -draw_options.-embedded.-radius.400.\$
$>$ $>$ $>$ $>$ -line_width.2.-end.\$
$>$ $>$ $>$ -draw_graphs_at_level.12.\$
$>$ $>$ -end.\$
$>$ $>$ -print_symbols
```
<table>
<thead>
<tr>
<th>Key</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-rainbow</td>
<td></td>
<td>Find all rainbow cliques. The size of the cliques is the number of vertex colors.</td>
</tr>
<tr>
<td>-target_size</td>
<td>$s$</td>
<td>Find all cliques of size $s$.</td>
</tr>
<tr>
<td>-weighted</td>
<td>$s$</td>
<td>Find weighted cliques.</td>
</tr>
<tr>
<td>-Sajeeb</td>
<td></td>
<td>Use the implementation by Sajeeb Chowdhury.</td>
</tr>
<tr>
<td>-output_file</td>
<td>$\text{fname}$</td>
<td>Write cliques to the named file.</td>
</tr>
<tr>
<td>-restrictions</td>
<td>$l , r , m$</td>
<td>Restricted search: At level $l$, restrict to all branches congruent to $r$ modulo $m$. Here, $0 \leq r &lt; m$.</td>
</tr>
</tbody>
</table>

Table 13.5: Clique Finding Options

### 13.4 Clique Finding

A clique in a graph $\Gamma = (V, E)$ is a subset $S$ of the vertices such that any two elements of $S$ are adjacent in $\Gamma$. Finding and classifying cliques in graphs is important for many applications of graph theory. Orbiter can help. The command `-find_cliques` command from Table 13.3 can be used to find all cliques in a graph. Additional options for this command are shown in Table 13.5. For instance, the cliques of size 3 in the graph `graph_v5_e7.colored_graph` from Section 13.1 can be found using

```plaintext
small_graph_cliques:
\>$\text{\$(ORBITER\_PATH)orbiter.out:\text{-v.10\backslash}$}
\>$\text{\text{-define\text{-G-g}\text{-graph-load_from_file-graph_v5_e7.colored_graph-end\backslash}$}
\>$\text{\text{-with\text{-G-g}\text{-do\backslash}$}
\>$\text{\text{-graph_theoretic_activity\text{-find_cliques\text{-target_size.3-end$}}}
```

This command finds three cliques of size 3.

It is also possible to classify all cliques under the automorphism group of the graph. This is a multi-step process, though. At first, the automorphism group of the graph has to be computed. Then, poset classification can be invoked to classify the cliques of a certain size. Here is an example. We wish to classify the cliques in the Paley graph of order 13. The first command creates the graph and computes the automorphism group:

```plaintext
Paley_13.aut:
\>$\text{\$(ORBITER\_PATH)orbiter.out:\text{-v.2\backslash}$}
\>$\text{\text{-define\Gamma-g\text{-graph-Paley.13-end\backslash}$}
\>$\text{\text{-with\Gamma-g\text{-do\backslash}$}
\>$\text{\text{-graph_theoretic_activity\text{-automorphism_group$}}
```
The command writes a file `Paley_13_group.makefile`, shown below:

Paley_13:

```
> $(ORBITER_PATH)orbiter.out -v 2 \\
> -define gens -vector -file Paley_13 gens.csv -end \\
> -define G -permutation_group \\
> -bsgs Paley_13 "Paley_13".13.78 "0,1".3 gens -end \\
```

The group is given using a base and strong generating set. The base consists of the two points 0, 1. Three strong generators with respect to this base are given in a csv file. Using this group, the next command classifies all cliques of size at most 5 in the Paley graph of order 13 under the action of the automorphism group. It turns out that there are no 5-cliques, and that the largest cliques have size 3. The command shows that there is a unique orbit of 3-cliques:

Paley_13_cliques_classify:

```
> $(ORBITER_PATH)orbiter.out -v 4 \\
> -define gens -vector -file Paley_13 gens.csv -end \\
> -define G -permutation_group \\
> -bsgs Paley_13 "Paley_13".13.78 "0,1".3 gens -end \\
> -define Gamma -graph -Paley_13 -end \\
> -with G -do \\
> -group_theoretic_activity \\
> -poset_classification_control \\
> -problem_label Paley_13_cliques \\
> -clique_test Gamma \\
> -depth 5 \\
> -end \\
> -orbits_on_subsets 5 \\
> -report \\
```

For comparison, the command

Paley_13_cliques:

```
> $(ORBITER_PATH)orbiter.out -v 10 \\
> -define Gamma -graph -Paley_13 -end \\
> -with Gamma -do \\
> -graph_theoretic_activity -find_cliques -target_size 3 -end
```

finds all cliques of size 3. There are exactly 26 of them. Because of the previous command, we know that they are all equivalent under the automorphism group of the graph.
Let us consider the orbital graph created in Section 13.1. We wish to study the 5-cliques. We first determine the number of 5-cliques, and then the number of orbits of 5-cliques under the automorphism group. The following command computes all 5-cliques:

```
HJ64_cliques5:
  ▶ $(ORBITER_PATH)orbiter.out -v 6 \
  ▶     -define Gamma -graph \n  ▶     -load \n  ▶     Group_Perms15_Orbital_3.colored_graph \n  ▶     -end \n  ▶     -with Gamma -do \n  ▶     -graph_theoretic_activity \n  ▶     -find_cliques -target_size 5 -end \n  ▶     -end
```

It turns out that there are exactly 1008 5-cliques. Concerning the classification with respect to the automorphism group of the graph, we apply the following command:

```
HJ64_cliques5_classify:
  ▶ $(ORBITER_PATH)orbiter.out -v 6 \
  ▶     -define Gamma -graph \n  ▶     -load \n  ▶     Group_Perms15_Orbital_3.colored_graph \n  ▶     -end \n  ▶     -define gens -vector \n  ▶     -file halljanko315_gens.csv \n  ▶     -end \n  ▶     -define G -permutation_group \n  ▶     -bsgs halljanko315 "File\halljanko315" \n  ▶     315 1209600 0 1 42 95 6 gens -end \n  ▶     -with G -do \n  ▶     -group_theoretic_activity \n  ▶     -poset_classification_control \n  ▶     -problem_label HJ64_cliques \n  ▶     -clique_test Gamma \n  ▶     -depth 5 \n  ▶     -end \n  ▶     -orbits_on_subsets 5 \n  ▶     -report \n  ▶     -end
```

This command shows that all of the 1008 5-cliques lie in one orbit under the group. The orbit representative picked by Orbiter is \{0, 8, 31, 110, 283\}. These numbers refer to the vertices of the graph. They are zero-based. The stabilizer of the clique has order 1200.
Let us look at the collinearity graph of $Q(4, 2)$ one more time. The following command computes the cliques of size 3:

```
PGO_5_2_cliques:0_5_2_incidence_matrix.csv.
  $(ORBITER_PATH)orbiter.out.-v.3\n  -define:Inc.-vector.-file:0_5_2_incidence_matrix.csv.-end\n  -define:Gamma.-graph.-collinearity_graph.Inc.-end\n  -with:Gamma.-do\n  -graph_theoretic_activity\n  -find_cliques.-target_size:3.-end\n  -end
```

There are 15 cliques of size 3. They correspond to the lines in the geometry.
Chapter 14

Combinatorial Objects

14.1 Combinatorial Objects

Combinatorial objects are objects that are defined by means of a finite group action. The isomorphism problem for combinatorial objects is the question to decide whether two objects of the same type belong to the same orbit under the relevant group action. Orbiter offers a unified treatment of such questions for various types of objects. The main tool is the computation of a canonical form, as well as the automorphism group.

Combinatorial objects are coded as sequences of integers. Each type of object has its own coding. Coding of objects as integer sequences allows easy handling of objects. For instance, objects can be specified in a command line argument, or they can be stored in a file. Large numbers of objects can be stored in files.

In order to apply Orbiter commands, an input stream is defined. An input stream is a sequence of objects, all of the same kind. The objects can be defined using any of the commands listed in Table 14.1. The file types will be discussed in more detail in the next section. Here are some examples. First, we create the Hirschfeld surface. Since the Hirschfeld surface is a cubic surface, the object is defined using point ranks in the relevant projective space as described in Section 4.2. For instance, the Hirschfeld surface in PG(3, 4) is defined as 45 points, coded as 45 integers which are point ranks. A makefile variable is employed to define the set. The makefile variable is then used to define a set-object:

HIRSCHFELD_SURFACE_Q4_SET_OF_POINTS="0,1,2,3,4,5,6,7,8,9,\
10,11,12,13,14,23,26,27,30,31,34,35,38,39,42,47,48,51,52,\
53,54,59,60,61,62,67,68,69,70,75,76,79,80,81,82"

Hirschfeld_q4_from_set:
▷ $(ORBITER_PATH)orbiter.out:-v.4:\
▷ ▷ -define:H:-set:-here:\
▷ ▷ ▷ $(HIRSCHFELD_SURFACE_Q4_SET_OF_POINTS)\
▷ ▷ ▷ -end:\
▷ ▷ -define:C:-combinatorial_objects\
▷ ▷ ▷ -set.of_points:H\"
### Table 14.1: Defining Combinatorial Objects

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-set_of_points</td>
<td></td>
<td>A set consisting of points.</td>
</tr>
<tr>
<td>-set_of_lines</td>
<td></td>
<td>A set consisting of lines.</td>
</tr>
<tr>
<td>-set_of_points_and_lines</td>
<td></td>
<td>A set consisting of points and a second set consisting of lines.</td>
</tr>
<tr>
<td>-set_of_packings</td>
<td></td>
<td>A set of packings.</td>
</tr>
<tr>
<td>-file_of_points</td>
<td></td>
<td>A set consisting of points read from file.</td>
</tr>
<tr>
<td>-file_of_lines</td>
<td></td>
<td>A set consisting of lines read from file.</td>
</tr>
<tr>
<td>-file_of_packings</td>
<td></td>
<td>A set consisting of packings read from file.</td>
</tr>
<tr>
<td>-file_of_packings_through_spread_table</td>
<td></td>
<td>A file of packings.</td>
</tr>
<tr>
<td>-file_of_point_set</td>
<td></td>
<td>A file containing point sets.</td>
</tr>
<tr>
<td>-file_of_designs</td>
<td></td>
<td>A file containing designs or large sets.</td>
</tr>
<tr>
<td>-file_of_incidence_geometries</td>
<td>$v \ b \ f$</td>
<td>A file of incidence geometries defined by their set of flags. Here, $v$ is the number of points, $b$ is the number of blocks and $f$ is the number of flags.</td>
</tr>
<tr>
<td>-file_of_incidence_geometries_by_row_ranks</td>
<td></td>
<td>A file describing incidence geometries defined by their row ranks.</td>
</tr>
<tr>
<td>-incidence_geometry</td>
<td>flags $v \ b \ f$</td>
<td>An incidence geometry defined by a set of flags. Here, $v$ is the number of points, $b$ is the number of blocks and $f$ is the number of flags.</td>
</tr>
<tr>
<td>-incidence_geometry_by_row_ranks</td>
<td></td>
<td>An incidence geometry defined by row ranks.</td>
</tr>
<tr>
<td>-from_parallel_search</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The next example creates the two hyperovals in PG(2, 16). The hyperovals are stored in makefile variables:

```plaintext
HYPEROVAL.16.144="0,1,2,3,52,67,89,106,126,\n141,159,176,184,199,220,235,245,262"
HYPEROVAL.16.16320="0,1,2,3,52,70,83,109,127,\n139,156,174,186,199,217,229,256,264"
```

`hyperoval.16_create:

In the next example, we read the points of an elliptic curve from file (see Section 4.2):

```plaintext
EC_read: elliptic_curve_b1_c3_q11.txt
$ (ORBITER_PATH) orbiter.out -v 4 -define C -combinatorial_objects -file_of_points elliptic_curve_b1_c3_q11.txt -end
```

In the next example, we read a packing, using a previously defined table of spreads, stored in a csv file.

```plaintext
PG.3.5.assume.31_read:
$ (ORBITER_PATH) orbiter.out -v 2 -define C -combinatorial_objects -file_of_packings_through_spread_table -H31.packings.csv -SPREAD_TABLES.5_REG/spread.25.spreads.csv -5 -end
```

The following command reads a file of large sets of designs:

```plaintext
LS_AG.2.3_read:
$ (ORBITER_PATH) orbiter.out -v 2 -define C -combinatorial_objects -file_of_designs -solutions.csv 9:84:3:12 -end
```
The next command reads incidence geometries from a file containing the flags:

```
geo_7_3_read:
  $(ORBITER_PATH)orbiter.out -v.10\n  -draw.incidence_structure_description\n  -width.60-with.10.6-end\n  -define.C:combinatorial_objects\n  -file_of.incidence.geometries\n  7_3.inc:7.7-21\n  -end
```

The next command creates incidence geometries from a file containing row-ranks:

```
Desargues_path_lex_least_read:
  echo $(DESARGUES_PATH_LEX_LEAST)>Desargues_path_lex_least.inc
  $(ORBITER_PATH)orbiter.out -v.10\n  -draw.incidence_structure_description\n  -width.60-with.10.6-end\n  -define.C:combinatorial_objects\n  -file_of.incidence.geometries_by_row_ranks\n  -Desargues_path_lex_least.inc:10.10.3\n  -end
```
14.2 Encoding Combinatorial Objects

Combinatorial objects can be stored in text files. There can be any number of objects in one file. The objects themselves are coded. For instance, a set of points in projective space is given as a set of integers, using the Orbiter point ranks. Likewise, a set of lines is coded using Orbiter line ranks. For designs, there are several ways in which the object can be stored. One way is by listing the incidences in a numerical form. One number is one incidence. Another way is by describing the incidence matrix in a row-by-row fashion, using ranks of $k$-subsets. This assumes that the number of incidences per row is constant over all rows. Yet another way is by listing the columns of the incidence matrix, again using ranks of $k$-subsets. This version requires that the column sums of the incidence matrix are constant. Let us go over some of these file formats, using small examples to illustrate the ideas informally.

Suppose we want to work with the Pasch configuration. This is the configuration of 6 points and 4 lines shown in Figure 14.1. In the geometry, we have 4 lines, which we can identify with the index sets of the points as \{0, 1, 2\}, \{0, 3, 4\}, \{1, 3, 5\} and \{2, 4, 5\}. The incidence matrix of the configuration is shown in Figure 14.2. Row labels are on the left, column labels are on top. The $(i, j)$-entry is one if $P_i$ lies on $\ell_j$, and it is zero otherwise. There are three ways to encode the incidence structure. One way encodes the flags of the geometry. This will be described next. The flag space is the set of all possible flags in the incidence matrix between the given number of points and lines. The space is totally ordered using the row-major ordering (see Figure 14.3). The Pasch configuration can now be coded as

\[
\{0, 1, 4, 6, 8, 11, 13, 14, 17, 19, 22, 23\}.
\]

The file `pasch.inc` contains:

```
6 4 12
```

Figure 14.1: The Pasch configuration
<table>
<thead>
<tr>
<th>$\ell_0$</th>
<th>$\ell_1$</th>
<th>$\ell_2$</th>
<th>$\ell_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$P_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$P_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$P_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$P_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$P_5$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 14.2: The incidence matrix of the Pasch configuration

<table>
<thead>
<tr>
<th>$\ell_0$</th>
<th>$\ell_1$</th>
<th>$\ell_2$</th>
<th>$\ell_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$P_1$</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$P_2$</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>$P_3$</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>$P_4$</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>$P_5$</td>
<td>20</td>
<td>21</td>
<td>22</td>
</tr>
</tbody>
</table>

Figure 14.3: Row-major ordering of the flag space

0 1 4 6 8 11 13 14 17 19 22 23
-1 1
24

The first line lists the number of rows and columns of the incidence matrix, and the number of incidences. The geometry is encoded on the next line. After that, a marker of -1 shows that this is the only geometry in this file (the file format allows for any number of incidence geometries, all with the same parameters). The final row is the order of the automorphism group of the geometry. This row is optional. In case that there are several geometries in the file, the orders will all be listed. In this case, the possible values will be collected with multiplicities, and listed in decreasing order. The command

```
geo_pasch_read:
▷ $(ORBITER_PATH)orbiter.out.-v.10.\$
▷ ▷ -define-C.-combinatorial_objects\$
▷ ▷ ▷ -file_of_incidence_geometries\$
▷ ▷ ▷ ▷ pasch.inc.6.4.12.\$
▷ ▷ ▷ -end
```

reads the incidence geometry from the file pasch.inc. It is also possible to enter the incidence geometry directly from the command line. The following example uses the -incidence.geometry command to do so:
geo_pasch_given:
  $\$(ORBITER\_PATH)\texttt{orbiter.out}\$ -v 10 \\
  -\texttt{define C-}\texttt{combinatorial_objects} \\
  -\texttt{incidence}\texttt{geometry} \\
  "0,1,4,6,8,11,13,14,17,19,22,23" \\
  6\cdot4\cdot12 \\
  -\texttt{end}
Chapter 15
Canonical Forms with Nauty

15.1 Overview of Canonical Forms

What is a combinatorial object? For the purposes of Orbiter, it is any kind of object that has a representation as a set of sets, all drawn from an underlying finite set. We allow colorings of the elements of the underlying set and of the sets in the set-system. The representation is functorial. Isomorphisms between the combinatorial objects must correspond to color preserving bijections of the set-representation and vice-versa. Under these conditions, the isomorphisms between combinatorial objects and automorphisms from one object to itself correspond to the mappings between the associated set representations.

The set-representation of combinatorial objects can help us computationally approach the isomorphism problem. We simply search for color-preserving bijections that take the set-representation of the object to the set-representation of the other object. Automorphisms can be found by mapping the set-representation of the object to itself.

Canonical labelings can be used to eliminate the need to do pairwise isomorphism testing. This is particularly helpful if the number of objects to test is large. If we have \( N \) objects, say, then pairwise isomorphism testing requires \( \binom{N}{2} \) tests. With canonical forms, we only need \( N \) canonical forms computations.

Sets of sets are incidence structures. The Levi graph of an incidence structure is the bipartite graph whose two classes correspond to rows and columns of the incidence matrix. The partition of the set system (underlying point set and set of sets) reduces to a coloring of the vertices of the graph. Two combinatorial objects are isomorphic if and only if the associated colored Levi graphs are isomorphic in the sense of graph isomorphism. This allows to settle many questions associated with combinatorial object, such as isomorphism testing and determining the automorphism group.

A canonical labeling of a graph is a bijection of the vertices. The property is that if two graphs are isomorphic, then the graphs become identical if the canonical labeling permutation is applied (each graph has its own canonical labeling). It is therefore important to compute canonical forms. If there is a vertex coloring, we implicitly assume that the canonical labeling preserves the coloring.
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-max_TDO_depth</td>
<td>$d$</td>
<td>Limit TDO depth to $d$ in the report.</td>
</tr>
<tr>
<td>-classification_prefix</td>
<td>prefix</td>
<td>Use the given prefix when writing files related to the classification.</td>
</tr>
<tr>
<td>-save_ago</td>
<td></td>
<td>Save the automorphism group orders to file.</td>
</tr>
<tr>
<td>-save_transversal</td>
<td></td>
<td>Save the indices of the elements chosen for the transversal.</td>
</tr>
</tbody>
</table>

Table 15.1: Orbiter commands related to canonical labelings

The graph theory package Nauty [50] provides a canonical form algorithm for graphs. Using the Levi graph construction, this technique allows to solve the isomorphism problem for combinatorial objects in the more general sense just defined.

The technique of isomorphism testing can be lifted to combinatorial objects in projective spaces or other types of finite incidence geometries. For instance, arcs in projective planes have been classified this way (cf. [2]).

Table 15.1 list Orbiter commands related to canonical labelings of combinatorial objects.
15.2 Canonical Forms of Objects in Projective Space

Suppose we want to compute the stabilizer of an elliptic curve. In Section 4.1, we have created an elliptic curve over $\mathbb{F}_{11}$ and stored the set of $\mathbb{F}_q$-points in the file `elliptic_curve_b1_c3_q11.txt`. The following example computes the set stabilizer of the curve. This means it computes the set stabilizer of the points on the curve. In order to do so, an input stream is created which referst to the file containing the Orbiter point ranks of points on the curve.

```plaintext
EC_canon::elliptic_curve_b1_c3_q11.txt
  > $(ORBITER_PATH)orbiter.out.-v.4\n  >  -define-C:-combinatorial_objects\n  >  > -file_of_points:elliptic_curve_b1_c3_q11.txt\n  >  > -end\n  >  > -define-F:-finite_field-q.11:-end\n  >  > -define-P:-projective_space-2:F:-end\n  >  > -with-C:-do\n  >  > -combinatorial_object_activity\n  >  >  -canonical_form_PG:P\n  >  >  > -classification_prefix:EC\n  >  >  >  -label:EC\n  >  >  >  -save_ago\n  >  >  >  -max_TDO_depth:4\n  >  >  >  -end\n  >  >  >  -report\n  >  >  >  > -prefix:EC\n  >  >  >  >  -export_flag_orbits\n  >  >  >  >  -show_TDO\n  >  >  >  >  -show_TDA\n  >  >  >  >  -dont_show_incidence_matrices\n  >  >  >  >  -export_group\n  >  >  >  >  -end\n  >  >  >  end
```

Orbiter shows that the curve has a collineation stabilizer of order 6, generated by
The following example computes the canonical form and the automorphism group of the Hirschfeld surface in PG(3, 4). Using indexing of points in PG(3, 4), we encode the surface as a set of points using Orbiter ranks. We use a makefile variable to provide these ranks as input for the canonical form procedure.

HIRSCHFELD\_SURFACE\_Q4\_SET\_OF\_POINTS="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,23,\ 26,27,30,31,34,35,38,39,42,47,48,51,52,53,54,59,60,61,62,67,68,69,70,75,76,\ 79,80,81,82"

Hirschfeld\_q4\_c: Hirschfeld\_surface\_q4.txt
  $(ORBITER\_PATH)orbiter.out: -v 40:
  -define C -combinatorial_objects:
  -define -file_of_points Hirschfeld\_surface\_q4.txt:
  -end:
  -define -finite_field -q 4 -end:
  -define -projective_space -3 F -end:
  -with C -do:
  -combinatorial_object_activity:
  -canonical_form PG P:
  -classification_prefix Hirschfeld\_surface\_q4:
  -save ago:
  -max TDO depth 10:
  -end:
  -report:
  -show TDO:
  -end:
  -end
pdflatex Hirschfeld\_surface\_q4\_classification.tex
open Hirschfeld\_surface\_q4\_classification.pdf

In the next example, we look at the two hyperovals in PG(2, 16).

hyperoval\_16\_c:
  $(ORBITER\_PATH)orbiter.out: -v 2:
  -define C -combinatorial_objects:
  -set of points $(HYPEROVAL\_16\_16320):

382
-set_of_points:$(HYPEROVAL_{16,144})\$
-define:F:-finite_field:-q.16:-end:
-define:P:-projective_space:2.F:-end:
-with:C:-do:
-definend_combinatorial_object_activity:\
-canonical_form_PG-P:\

classification_prefix_hyperoval_q16:\
-label:hyperoval_q16:\
-save_ago:\
-save_transversal:\
-max_TDO_depth:10:\
-end:\
-report:\
-prefix_hyperoval_q16:\
-export_flag_orbits:\
-show_TDO:\
-show_TDA:\
-dont_show_incidence_matrices:\
-export_group:\
-end:\
-end
pdfflatex:hyperoval_q16_classification.tex
open:hyperoval_q16_classification.pdf

$($ORBITER_PATH$)oribiter.out:-v.2.-draw_matrix:\
-input_csv_file_hyperoval_q16_object0_TDA_flag_orbits.csv:\
-secondary_input_csv_file_hyperoval_q16_object0_TDA.csv:\
-box_width:4.-bit_depth:24:\
-end
open:hyperoval_q16_object0_TDA_flag_orbits_draw.bmp

$($ORBITER_PATH$)oribiter.out:-v.2.-draw_matrix:\
-input_csv_file_hyperoval_q16_object1_TDA_flag_orbits.csv:\
-secondary_input_csv_file_hyperoval_q16_object1_TDA.csv:\
-box_width:4.-bit_depth:24:\
-end
open:hyperoval_q16_object1_TDA_flag_orbits_draw.bmp
15.3 Canonical Forms of Incidence Geometries

Let us consider a system of subsets. This subsets are chosen from the same set, which we call the ground set. The elements of the group set are often called points. In many cases, there are conditions that restrict the way in which the sets can be chosen. There is a notion of isomorphism on such set systems. Two set systems are isomorphic if there is a bijection between the underlying ground sets which takes one to the other. The incidence matrix is the 0/1 matrix whose rows correspond to the elements of the group set, and whose columns correspond to the chosen subsets. An entry 1 indicates that the corresponding point belongs to the corresponding set.

An incidence geometry is a set system with the following properties: No set appears twice, and no pair of elements in the ground set appear in two different sets. The elements of the ground set are called points. The sets are called lines (or sometimes planes). A flag is an incident point-line pair. An anti-flag is a non-incident point-line pair. Two points are said to be collinear of there is a line in the geometry containing both points.

It is interesting to study the action of the automorphism group on the elements of a geometry. Properties of interest are various levels of transitivity on the elements of the geometry. For instance, a geometry is line-transitive if the automorphism group is transitive on lines. Likewise, it is flag transitive if the automorphism group is transitive on flags. The collinearity graph of a geometry is the graph whose vertices correspond to the points, with two vertices adjacent of the associated points are collinear. The girth of the incidence geometry is the girth of the associated collineation graph. A geometry is triangle free if its girth is at least 4.

A configuration \( v_r b_k \) is an incidence geometry with a ground set of size \( v \) and with \( b \) lines such that each line has size \( k \) and each point is contained in exactly \( r \) lines. In the special case where \( b = v \) and \( k = r \), the name symmetric configuration \( v_r \) is used (the term symmetric is somewhat misleading because the incidence matrix of a symmetric configuration need not be symmetric). Orbiter can be used to classify incidence geometries. One of the important steps in this process is computing a canonical form of the incidence geometry.

We will also be producing drawings of the incidence matrices of geometries. In these drawings, flags are indicated as heavy squares while anti-flags are drawn as small squares. The coloring will indicate the orbits of the automorphism group on flags and anti-flags. Objects with the same color belong to the same orbit. For a flag-transitive geometry, there is only one color for the incidences.

The following command computes the canonical form and a report of the projective plane \( \text{PG}(2,2) \), which is a configuration \( 7_3 \).

```
geo_7_3.c:
  $(ORBITER_PATH)orbiter.out-v.10-
  -draw.incidence.structure.description-
  -width.60-with.10.6.-end-
```
A bitmap drawing is produced, as shown in Figure 15.1. The command also produces the following report of the geometry:

### Summary of Orbits

<table>
<thead>
<tr>
<th>Rep</th>
<th>#</th>
<th>Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>168</td>
</tr>
</tbody>
</table>

Ago: 168
Isomorphism type 0 / 1

Isomorphism type 0 / 1 is original object 0 and appears 1 times:
This isomorphism type appears 1 times, namely for the following 1 input objects:
{0}

incidence structure:
( 0, 1, 2, 7, 10, 11, 14, 19, 20, 22, 24, 26, 29, 32, 34, 37, 38, 41, 44, 46, 47 )

Column sets of the encoded object:
{ 0, 1, 2 }
{ 0, 3, 4 }
{ 0, 5, 6 }
{ 1, 3, 5 }
{ 1, 4, 6 }
{ 2, 3, 6 }
{ 2, 4, 5 }

Row sets of the encoded object:
{ 0, 1, 2 } = 0
{ 0, 3, 4 } = 9
{ 0, 5, 6 } = 14
{ 1, 3, 5 } = 20
{ 1, 4, 6 } = 23
{ 2, 3, 6 } = 27
{ 2, 4, 5 } = 28
Generators for the automorphism group:
The stabilizer of order 168 is generated by:
\[ g_1 = (3, 5)(4, 6)(8, 9)(12, 13) \] of order 2 and with 6 fixed points.
\[ g_2 = (3, 4)(5, 6)(10, 11)(12, 13) \] of order 2 and with 6 fixed points.
\[ g_3 = (1, 2)(5, 6)(10, 12)(11, 13) \] of order 2 and with 6 fixed points.
\[ g_4 = (1, 3)(2, 4)(7, 8)(11, 12) \] of order 2 and with 6 fixed points.
\[ g_5 = (0, 1)(4, 5)(8, 10)(9, 11) \] of order 2 and with 6 fixed points.

Canonical labeling:
canonical row = 6
canonical orbit number = 0
Flags : ( 0, 1, 2, 7, 10, 11, 14, 19, 20, 22, 24, 26, 30, 31, 34, 36, 39, 41, 44, 46, 47 )

Flag orbits:
orbit length : number of orbits of that length:

\[ 21 \ 1 \]

Anti-Flag orbits:
orbit length : number of orbits of that length:

\[ 28 \ 1 \]

The following command computes the canonical form and a report of the affine plane AG(2,3), which is a configuration 9_{4123}.

\texttt{AG\_2\_3\_c::AG\_2\_3.inc}
\texttt{ $(\text{ORBITER\_PATH})\text{or} \text{biter.out} -v\cdot2\backslash$
\texttt{ \texttt{define \texttt{C::combinatorial\_objects}}$\	exttt{ \texttt{define \texttt{C::file\_of\_incidence\_geometries}}$
\texttt{ \texttt{define \texttt{C::AG\_2\_3\_inc\texttt{9\cdot12\_36}}$
\texttt{ \texttt{define \texttt{C::end}}}$

\[387\]
Figure 15.2: The affine plane $\text{AG}(2, 3)$ is a configuration $9_4 12_3$.

A bitmap drawing is produced, shown in Figure 15.2. Because the geometry is flag transitive, there is only one color being used for the incidence. In fact, all incidences are in black. The geometry is also anti-flag transitive. This can be seen from the fact that there is only one color in the picture for the smaller boxes, which represent anti-flags. Orbiter also produces
the following report of the geometry:

**Summary of Orbits**

<table>
<thead>
<tr>
<th>Rep</th>
<th># Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>432</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Ago: 432

**Isomorphism type 0 / 1**

Isomorphism type 0 / 1 is original object 0 and appears 1 times:

This isomorphism type appears 1 times, namely for the following 1 input objects:

{0}

incidence structure:

( 0, 1, 2, 3, 12, 16, 17, 18, 24, 31, 32, 33, 37, 40, 43, 46, 49, 53, 56, 59, 62, 64, 69, 71, 74, 78, 80, 82, 87, 89, 93, 94, 99, 102, 103, 107 )

Generators for the automorphism group:

The stabilizer of order 432 is generated by:

\[ g_1 = (3, 4)(5, 7)(6, 8)(11, 12)(13, 14)(16, 17)(19, 20) \]

of order 2 and with 7 fixed points.

\[ g_2 = (3, 5)(4, 6)(7, 8)(10, 11)(14, 15)(16, 18)(19, 20) \]

of order 2 and with 7 fixed points.

\[ g_3 = (1, 3)(2, 4)(7, 8)(9, 10)(14, 16)(15, 19)(18, 20) \]

of order 2 and with 7 fixed points.

\[ g_4 = (0, 1)(4, 5)(6, 7)(10, 13)(11, 14)(12, 15)(17, 18) \]

of order 2 and with 7 fixed points.

Decomposition by combinatorial refinement:

\[
\begin{array}{c|c}
\rightarrow & 12_1 \\
9_0 & 4 \\
\downarrow & 12_1 \\
9_0 & 3 \\
\end{array}
\]
Decomposition by automorphism group:

Canonical labeling:
- canonical row = 6
- canonical orbit number = 0
- Flags: (0, 1, 2, 3, 12, 16, 17, 18, 24, 31, 32, 33, 37, 40, 43, 46, 49, 53, 56, 59, 62, 64, 69, 71, 75, 78, 79, 83, 87, 89, 93, 94, 98, 102, 104, 106)

Flag orbits:
- orbit length: number of orbits of that length:
  - 36: 1

Anti-Flag orbits:
- orbit length: number of orbits of that length:
  - 72: 1

It is possible to perform isomorph classification for configurations based on incidence files. Suppose we want to check that the configurations in 10_3 are in fact all nonisomorphic. We apply the canonical form algorithm given by Nauty. This produces a transversal of the isomorphism types of incidence geometries from the given list of input objects. The objects are specified by means of the \texttt{combinatorial\_objects} command. The classification algorithm can print a report which lists the transversal and all elements in it in latex form.

\begin{verbatim}
geo_10_3.c:
  > $(ORBITER\_PATH)orbor\_path\_version-\text{"\text{-v.2\"}}\$
  >     -\text{define}\text{-combinatorial\_objects}\$
  >     -\text{-file_of_incidence\_geometries:10\text{-3.inc:10\text{-30}}\$
  >     -\text{-end}\$
  >     -\text{-with}\text{-do}\$
  >     -\text{combinatorial\_object\_activity}\$
  >     -\text{-canonical\_form}\$
  >     -\text{-classification\_prefix:10\text{-3}}$
  >     -\text{-save\_ago}$$
\end{verbatim}
The report is shown below. It is truncated for reasons of space. Only the first two geometries are shown. Note that the ordering of geometries in the report may be different from the ordering in the input file. This is because the classification program sorts the geometries according to the canonical form. Also, note that the report includes the incidence geometry in the form it is given as well as the tactical decomposition induced by the orbits of the automorphism group.

### Summary of Orbits

<table>
<thead>
<tr>
<th>Rep</th>
<th>Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Ago :2, 3^2, 4^2, 6, 10, 12, 24, 120

### Isomorphism type 0 / 10

Isomorphism type 0 / 10 is original object 9 and appears 1 times:
This isomorphism type appears 1 times, namely for the following 1 input objects:
{9}
incidence structure:
( 0, 1, 2, 10, 13, 14, 20, 25, 26, 31, 33, 35, 41, 44, 47, 52, 53, 58, 62, 66, 69, 74, 78, 79, 85, 87, 89, 96, 97, 98 )
Generators for the automorphism group:
The stabilizer of order 3 is generated by:
\[ g_1 = (0, 1, 3)(2, 5, 4)(6, 7, 8)(10, 13, 11)(12, 14, 15)(16, 18, 17) \] of order 3 and with 2 fixed points.

Decomposition by automorphism group:

Canonical labeling:
canonical row = 5
canonical orbit number = 1
Flags : 0,1,2,16,17,18,25,27,29,34,38,39,40,43,45,51,53,56,62,63,64,70,74,77,82,86,89,91,95,98,

Isomorphism type 1 / 10

Isomorphism type 1 / 10 is original object 1 and appears 1 times:
This isomorphism type appears 1 times, namely for the following 1 input objects:
incidence structure:
( 0, 1, 2, 10, 13, 14, 20, 25, 26, 31, 33, 35, 41, 44, 47, 52, 54, 58, 62, 66, 69, 73, 78, 79, 85, 87, 89, 96, 97, 98 )

Generators for the automorphism group:
The stabilizer of order 2 is generated by:
$g_1 = (0, 6)(1, 9)(3, 8)(4, 7)(10, 16)(11, 19)(13, 17)(14, 18)$ of order 2 and with 4 fixed points.

Decomposition by automorphism group:

Canonical labeling:
canonical row = 0
canonical orbit number = 0
Flags : 0,1,2,15,18,19,24,26,29,33,37,39,40,43,44,50,55,56,61,67,68,72,75,77,82,84,88,91,93,96,

The following command computes the canonical form for the three triangle free configurations
24_3 found by Abdullah Alazemi. These configurations have 24 points, 24 lines, each line consists of 3 points and each point is on 3 lines.

TFC_24_3.c:
```bash
$ echo -n $(FILE_24_3_TFC_INC) | 24_3_TFC.inc
$ (ORBITER_PATH) orbiter.out -v 6 -
  -define -c -combinatorial_objects -
  -file_of_incidence_geometries 24_3_TFC.inc 24 24 72 -
  -end -
  -with -C -do -
  -combinatorial_object_activity -
  -canonical_form -
  -classification_prefix 24_3_TFC -
  -label 24_3_TFC -
  -save_ago -
  -end -
  -report -
  -prefix 24_3_TFC -
  -export_flag_orbits -
  -show TDO -
  -show incidence_matrices -
  -end -
  -end
$ pdflatex 24_3_TFC_classification.tex
$ open 24_3_TFC_classification.pdf
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix -
  -input_csv_file 24_3_TFC_object2_TDA_flag_orbits.csv -
  -secondary_input_csv_file 24_3_TFC_object2_TDA.csv -
  -box_width 20 -bit_depth 24 -
  -end
$ open 24_3_TFC_object2_TDA_flag_orbits_draw.bmp
```

The command also computes the tactical decomposition induced by the automorphism group. In addition, the command also computes the orbits on flags and on anti-flags. The third of the three geometries is flag transitive. A bitmap drawing is produced, shown in Figure 15.3. Because the geometry is flag transitive, there is only one color being used for the incidence. In fact, all incidences are in black.
Figure 15.3: A flag transitive $24_3$ configuration
In Secton 11.6, some large sets of AG(2, 3) were constructed. The final isomorphism classification is performed using the Nauty interface. A list of combinatorial objects is created, and the \texttt{-canonical_form} command is applied as activity. This will produce a list of pairwise non-isomorphic designs. The size of this list is the number of isomorphism types of large sets of AG(2, 3).

\begin{verbatim}
LS_AG_2_3_solutions_classify:
  \$ (ORBITER_PATH) orbiter.out \$ v.2 \$
  \$ -draw_incidence_structure_description \$
  \$ -width 30 \$ -with 10 \$ -end \$
  \$ -define C \$ -combinatorial_objects \$
  \$ -file_of_designs \$
  \$ -define solutions.csv \$ 9 \$ 84 \$ 3 \$ 12 \$
  \$ -end \$
  \$ -with C \$ do \$
  \$ -combinatorial_object_activity \$
  \$ -define \$ canonical_form \$
  \$ -save ago \$
  \$ -classification_prefix large_sets_of_AG_2_3 \$
  \$ -end \$
  \$ -report \$ "classification \$
  \$ -end \$
  pdflatex large_sets_of_AG_2_3_classification.tex
  open large_sets_of_AG_2_3_classification.pdf
\end{verbatim}

It turns out that there are exactly two isomorphism types, with automorphism groups of order 54 and 42, respectively.
15.5 Canonical Forms of Linear Codes

Orbiter can compute canonical forms and automorphism groups of codes using Nauty. For linear codes, the semilinear automorphism group can be computed.

Consider the $[3, 2, 2]$ code generated by

$$
\begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 1
\end{bmatrix}
$$

The semilinear automorphism group can be computed using the following command:

```
code 3 2 aut:
  $(ORBITER_PATH)orbiter.out -v:20 \n  -define:F:-finite_field:-q:2 -end \n  -define:genma:-vector:-field:F:-format:2 \n  -dense:$\text{CODE}_3_2\text{Q}_2\text{GENMA} \n  -end \n  -define:P:-projective_space:1:F:-end \n  -with:P:-do \n  -projective_space_activity \n  -canonical_form_of_code \n  "3_2":genma:-save_ago:-label:"3_2" \n  -classification_prefix:"3_2" \n  -end \n  end
```

The code has a semilinear automorphism group of order 6. The following report is written:

```
Summary of Orbits

<table>
<thead>
<tr>
<th>Rep</th>
<th># Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
```

The code has a semilinear automorphism group of order 6. The following report is written:
Isomorphism type 0 / 1

Isomorphism type 0 / 1 is original object 0 and appears 1 times:
This isomorphism type appears 1 times, namely for the following 1 input objects:
{} 
set of points of size 3: ( 0, 1, 2 )

<table>
<thead>
<tr>
<th>i</th>
<th>Rank</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(1,0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(0,1)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>

Column sets of the encoded object:
{ 0, 1, 2 }
{ 0, 1, 2, 3 }
Row sets of the encoded object:
{ 0, 1 } = 0
{ 0, 1 } = 0
{ 0, 1 } = 0
{ 1 } = 1

Generators for the automorphism group:
The stabilizer of order 6 is generated by:
g_1 = (1, 2) of order 2 and with 4 fixed points.
g_2 = (0, 1) of order 2 and with 4 fixed points.

Generators for the automorphism group as matrix group:
The stabilizer of order 6 is generated by:
g_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \end{bmatrix} of order 2 and with 1 fixed points.
g_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 01 \\ 10 \end{bmatrix} of order 2 and with 1 fixed points.

Decomposition by combinatorial refinement:
Decomposition by automorphism group:

\[
\begin{array}{c|c}
\rightarrow & 2_1 \\
4_0 & 2 \\
\downarrow & 2_1 \\
4_0 & 3 \\
\end{array}
\]

Canonical labeling:
- canonical row = 3
- canonical orbit number = 1

Flags: (0, 1, 2, 3, 4, 5, 7)

Flag orbits:
- orbit length : number of orbits of that length:
  - 1 1
  - 3 2

Anti-Flag orbits:
- orbit length : number of orbits of that length:
  - 1 1

We distinguish the 4 codewords of the \([5, 2, 3]_2\) code amongst the vertices of the Hamming graph \(H(5, 2)\) and compute the set stabilizer in the automorphism group of the graph.

```bash
Hamming_5_2_with_5_2_3_code:
  $(ORBITER_PATH)orbiter.out--v.2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
  $\$defineG$-graph$-Hamming_5_2\$
The group has order 32. For the graph theoretic commands, see Section 13.1.

The command

```latex
\texttt{CODE\_RM\_3\_1\_GENMA}="\\
1111111\\
0101010\\
0011001\\
0000111"
```

computes the automorphism group of the Reed-Muller code, of order 1344. It is the affine group $\text{AGL}(3,2)$. A report is created, showing the automorphism group and the action on $\text{PG}(3,2)$, with the Reed-Muller code distinguished.

The following command creates a drawing of the incidence matrix between points and lines in $\text{PG}(3,2)$, with the Reed-Muller code distinguished:

```latex
\texttt{REED\_MULLER\_3\_1\_CODEWORDS}="0,255,170,85,204,51,102,\\
153,240,15,90,165,60,195,150,105"
```

computes the automorphism group of the Reed-Muller code, of order 1344. It is the affine group $\text{AGL}(3,2)$. A report is created, showing the automorphism group and the action on $\text{PG}(3,2)$, with the Reed-Muller code distinguished.
Figure 15.4: PG(3, 2) with the Reed-Muller code distinguished

\begin{verbatim}
> > -end;
> > -define P.-projective_space.3:F.-end;
> > -with P.-do;
> > -projective_space_activity;
> > -canonical_form_of_code;
> > -define genma.-vector_field F.-format 4;
> > -compact $(CODE RS 6 4 7);
> > -end;

pdflatex RM_3.1_classification.tex
open RM_3.1_classification.pdf

$(ORBITER_PATH)orbiter.out --v.2 -draw_matrix -input_csv_file RM_3.1_object0_INP_flag_orbits.csv -secondary_input_csv_file RM_3.1_object0_INP.csv -box_width 16 -bit_depth 24

$(ORBITER_PATH)orbiter.out --v.2 -draw_matrix -input_csv_file RM_3.1_object0_TDA_flag_orbits.csv -secondary_input_csv_file RM_3.1_object0_TDA.csv -box_width 16 -bit_depth 24

open RM_3.1_object0_INP_flag_orbits_draw.bmp
open RM_3.1_object0_TDA_flag_orbits_draw.bmp
\end{verbatim}

The drawing in Figure 15.4 is created.

The command

\texttt{RS 6.4.7 group:}

\begin{verbatim}
$ (ORBITER_PATH)orbiter.out --v.20 -define F.-finite_field.-q.7 -end;
> > -define genma.-vector_field F.-format 4;
> > -compact $(CODE RS 6.4.7);
> > -end;
\end{verbatim}
shows that the automorphism group has order 12. After some shortening, the output is:

Isomorphism type 0 / 1 is original object 0 and appears 1 times:
set of points of size 6: \{(0, 9, 51, 344, 253, 3)\}

<table>
<thead>
<tr>
<th>i</th>
<th>Rank</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(1,0,0,0)</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>(5,1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>51</td>
<td>(6,5,1,0)</td>
</tr>
<tr>
<td>3</td>
<td>344</td>
<td>(0,6,5,1)</td>
</tr>
<tr>
<td>4</td>
<td>253</td>
<td>(0,0,4,1)</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>(0,0,0,1)</td>
</tr>
</tbody>
</table>

Group order 12
This isomorphism type appears 1 times, namely for the following 1 input objects:
\{0\}
Stabilizer:
Strong generators for a group of order 12:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
2 & 0 & 0 & 1 \\
5 & 0 & 6 & 0 \\
5 & 1 & 0 & 0 \\
\end{bmatrix}
\]
The command

\texttt{GV\_n15\_k6\_d5\_group:}
\begin{verbatim}
▷ $(\text{ORBITER\_PATH})\text{orbiter.out\_v.20}\$
▷ ▷ -define:F\_finite_field\_q.2\_end\$
▷ ▷ -define:genma\_vector\_field\_F\_format.6\$
▷ ▷ ▷ -compact:$(\text{CODE\_GV\_N15\_K6})\$
▷ ▷ -end\$
▷ ▷ -define:P\_projective\_space\_5\_F\_end\$
▷ ▷ -with:P\_do\$
▷ ▷ -projective\_space\_activity\$
▷ ▷ ▷ -canonical\_form\_of\_code\$
▷ ▷ ▷ ▷ "GV\_n15\_k6\_d5\_genma\_save\_ago\_label\_GV\_n15\_k6\_d5\$
▷ ▷ ▷ ▷ -classification\_prefix\_"GV\_n15\_k6\_d5\_end\$
▷ ▷ ▷ ▷ -end\$
▷ ▷ pdflatex\_GV\_n15\_k6\_d5\_classification.tex
▷ open\_GV\_n15\_k6\_d5\_classification.pdf
\end{verbatim}

computes the automorphism group of the Gilbert-Varshamov code from Section 10.8. It has order 12.
15.6 Canoncial Forms of General Codes

The command

```
HAMMING_CODE_CODEWORDS="0,·67,·37,·102,·22,·85,·51,·112,·15,·76,·42,·105,·25,·90,·60,·127"
```

computes the set stabilizer of the Hamming code inside the automorphism group of the Hamming graph. The group has order $2688 = 16 \cdot 168$. 
15.7 Canonical Forms of Graphs

Orbiter can compute isomorphism and automorphism between graphs. orbiter relies on the canonical labelings of graphs computed by Nauty [50], which is integrated into Orbiter. Here are some examples.

Suppose we want to compute the automorphism group of the cycle graph of order 13:

```plaintext
Cycle_13_aut:
▷ $(ORBITER_PATH)orbiter.out.-v.2.-define:Gamma.-graph:.-cycle.13.-end\n▷ ▷ -with:Gamma.-do:\n▷ ▷ -graph:theoretic:activity.-automorphism:group.-end\n```

The output is two files: The first one, `Cycle_13_group.makefile` is a makefile containing an Orbiter command to create the automorphism group:

```plaintext
Cycle_13:
▷ $(ORBITER_PATH)orbiter.out.-v.2\n▷ ▷ -define:gens.-vector.-file:Cycle_13.gens.csv.-end\n▷ ▷ -define:G.-permutation_group\n▷ ▷ -bsgs:Cycle_13:"Cycle\_13".i3.26."0,5".2.gens.-end\n```

The second file is `Cycle_13_gens.csv`, which contains the permutation representation of the group, and which is needed for the makefile:

```
Row,C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12
0,0,12,11,10,9,8,7,6,5,4,3,2,1
1,1,2,3,4,5,6,7,8,9,10,11,12,0
END
```

The next command computes the automorphism group of the chain graph with respect to the partition (2,3,2).

```plaintext
Chain_232_aut:
▷ $(ORBITER_PATH)orbiter.out.-v.2\n▷ ▷ -define:P1.-vector.-dense:2,3,2.-end\n▷ ▷ -define:P2.-vector.-dense:2,3,2.-end\n▷ ▷ -define:Gamma.-graph\n▷ ▷ ▷ -chain:graph:P1,P2\n▷ ▷ -end\n▷ ▷ -with:Gamma.-do\n▷ ▷ ▷ -graph:theoretic:activity.-automorphism:group\n▷ ▷ -end
▷ pdflatex:chain_graph_report.tex
▷ open:chain_graph_report.pdf
```

405
The following report is written:

The automorphism group of \textit{chain\_graph} has order 1152 and is generated by:

Strong generators for a group of order 1152:

\[(12,13),\]
\[(3,4),\]
\[(2,3),\]
\[(10,11),\]
\[(9,10),\]
\[(5,6),\]
\[(7,8),\]
\[(0,1),\]
\[(0,12)(1,13)(2,9)(3,10)(4,11)(5,7)(6,8)\]

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 12, 14,
0, 1, 2, 4, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
0, 1, 3, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 10, 12, 13, 14,
0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 9, 11, 12, 13, 14,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
0, 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 14,
1, 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
12, 13, 9, 10, 11, 7, 8, 5, 6, 2, 3, 4, 0, 1, 14,

Junttila and Kaski maintain a collection of graphs that can be used as test cases. The graphs are stored in Dimacs format and can be read in through the Orbiter \texttt{-load_dimacs} command. For instance, the following command computes the automorphism group of the Levi graph of the desarguesian projective plane:

\texttt{JK\_graph\_pp16\_1:}
\begin{verbatim}
  $(ORBITER\_PATH)orbiter.out-v.2\$
  -define:Gamma=graph-load_dimacs$
  -..\_JUNTTILA\_KASKI/benchmarks/pp/pp16\_1$
  -end$
  -with:Gamma-do$
  -graph\_theoretic\_activity-save-end$
  -with:Gamma-do$
  -graph\_theoretic\_activity-automorphism\_group-end$
\end{verbatim}
The command shows a group of order 34217164800. As a measurement of the complexity, the number of backtrack nodes in Nauty is recorded:

\[
\begin{align*}
\text{nb\_backtrack1} &= 6 \\
\text{nb\_backtrack2} &= 134 \\
\text{nb\_backtrack3} &= 134 \\
\text{nb\_backtrack4} &= 1
\end{align*}
\]

Here, \text{nb\_backtrack1} is the number of calls to \text{firstpathnode}, \text{nb\_backtrack2} is the number of calls to \text{othernode}, \text{nb\_backtrack3} is the number of calls to \text{processnode}, \text{nb\_backtrack4} is the number of calls to \text{firstterminal}. These are the four recursive functions in Nauty.

Unfortunately, the complexity of graph isomorphism is not well-understood. We can see this here. While the first projective plane of order 16 can be handled relatively easily, the second one causes problems. The following command hardly finishes:

\begin{verbatim}
JK_graph_pp16_2:
  $ (ORBITER\_PATH)orbiter.out\-v\-2\-
  -define\-Gamma\-graph\-load\_dimacs\-
  ../JUNTTILA\_KASKI/benchmarks/pp/pp16\-2\-
  -end\-
  -with\-Gamma\-do\-
  -graph\_theoretic\_activity\-save\-end\-
  -with\-Gamma\-do\-
  -graph\_theoretic\_activity\-automorphism\_group\-end\-
\end{verbatim}

The difference between the two planes is that the first plane has a very large automorphism group, while the second one has not. For any \( q \), the Desarguesian plane \( \text{PG}(2,q) \) has the largest automorphism group of all projective planes of order \( q \).

The following example considers the block intersection graph of a Steiner triple system ("STS") of order 13. There are exactly two STS(13). The one we consider here has a group of order 39. The block intersection graph has the same automorphism group.

\begin{verbatim}
JK_graph_sts_13:
  $ (ORBITER\_PATH)orbiter.out\-v\-2\-
  -define\-Gamma\-graph\-load\_dimacs\-
  ../JUNTTILA\_KASKI/benchmarks/srg/sts\-13\-
  -end\-
  -with\-Gamma\-do\-
  -graph\_theoretic\_activity\-save\-end\-
  -with\-Gamma\-do\-
  -graph\_theoretic\_activity\-automorphism\_group\-end
  make\-ORBITER\_PATH=$(ORBITER\_PATH)\-f\-sts\-13\_group\_makefile\-sts\-13
\end{verbatim}

407
The automorphism group has order 39 and is generated by:

\[(1, 25, 16)(2, 18, 20)(3, 7, 15)(4, 13, 11)(5, 6, 17)
(9, 12, 19)(10, 14, 24)(21, 23, 22),
(0, 1, 2)(3, 4, 5)(7, 8, 9)(11, 12, 13)(14, 16, 18)
(15, 17, 19)(20, 22, 24)(21, 23, 25)\]

Graphs can be created from groups by means of orbitals. An orbital is an orbit of a permutation group \(G\) on the set of pairs. Here is an example. We start from the Coxeter-Tits graph on 315 vertices, whose automorphism group is the Hall-Janko group \(J_2 : 2\). We first read the graph from file `halljanko315.csv` and compute the automorphism group using Nauty:

```
HJ_aut:
▷ $(ORBITER_PATH)orbiter.out -v 6:
  ▷ -define G: -graph:\n  ▷ -load_csv_no_border:\n  ▷ halljanko315.csv:
```

The next step is to compute the orbits of the automorphism group on pairs, using the following command:

```
HJ_group_and_orbits:
▷ $(ORBITER_PATH)orbiter.out -v 2:
  ▷ -define gens: -vector: -file:\n  ▷ -load_csv_no_border:\n  ▷ halljanko315 gens.csv: -end:\n  ▷ -define G: -permutation_group:\n  ▷ -bsgs halljanko315 "File\halljanko315":\n  ▷ 315 1209600 "0,1,2".6 gens:\n  ▷ -end:\n  ▷ -with G: -do:\n  ▷ -graph_theoretic_activity:\n  ▷ -poset_classification_control:\n  ▷ -W:\n  ▷ -problem_label: HJ_orbits:\n  ▷ -depth 2:\n  ▷ -end:\n  ▷ -orbits_on_subsets 2:\n```

408
There are 4 orbits on pairs. We decide to pick the fourth orbit to create a new graph. Because indexing is zero-based, we give the orbit index of 3:

HJ_orbital_graph_3:
\begin{verbatim}
> $(ORBITER_PATH)orbiter.out.-v.2\n> -define.gens.-vector.-file\n> > halljanko315.gens.-end\n> > -define.G.-permutation_group\n> > > -bsgs.halljanko315."File\_halljanko315".\n> > > 315:1209600."0,1,2":6.gens.\n> > > -end\n> > > -define.Gamma.-graph.\n> > > -orbital_graph.G.3.\n> > > -end\n> > > -with.Gamma.-do.\n> > > > -graph_theoretic_activity.\n> > > > -properties.\n> > > > -end\n> > > > -with.Gamma.-do.\n> > > > -graph_theoretic_activity.\n> > > > -save.\n> > > > -end
\end{verbatim}

The graph is regular of degree 64.

The next command computes the automorphism group of the collinearity graph of the $Q(4,2)$ quadric.

PGO_5_2_graph_group:0_5_2_incidence_matrix.csv.
\begin{verbatim}
> $(ORBITER_PATH)orbiter.out.-v.3\n> > -define.Inc.-vector.-file0_5_2_incidence_matrix.csv.-end\n> > -define.Gamma.-graph.-collinearity_graph.Inc.-end\n> > -with.Gamma.-do\n> > > -graph_theoretic_activity.\n> > > > -automorphism_group.\n> > > > -end
\end{verbatim}

The group is PGO(5,2) of order 720. The command creates the group as a permutation group on the 15 vertices of the graph. The group is no longer treated as a matrix group.
Chapter 16

Interfaces

16.1 Graphical Output

Orbiter can produce graphical output in a variety of formats:

1. TikZ / Latex [63],
2. Metapost [33],
3. Bitmap files (.bmp) [66],
4. Povray, see Section 16.2.

Bitmaps can be created using the `-draw_matrix` command. The input is an integer-valued matrix in csv format. The matrix entries are translated into colors. The possible commands after `-draw_matrix` are shown in Table 16.1. Suppose we want to create a graphical representation of the addition table of the finite field \( \mathbb{F}_7 \). The following command sequence first creates the addition and multiplication tables of the field, and then produces a bitmap graphic for the addition table:

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-input_csv_file</code></td>
<td>csv-file</td>
<td>Specify the input csv-file</td>
</tr>
<tr>
<td><code>-partition</code></td>
<td>( w \ R \ C )</td>
<td>Specify a partitioning ( R ) of rows and ( C ) of columns. Use separating lines of with ( w ).</td>
</tr>
<tr>
<td><code>-box_width</code></td>
<td>( w )</td>
<td>Use ( w ) pixels per matrix entry.</td>
</tr>
<tr>
<td><code>-bit_depth</code></td>
<td>( d )</td>
<td>Use color bit depth of ( d ) bits ( (d = 8 ) or ( d = 24 )).</td>
</tr>
<tr>
<td><code>-invert_colors</code></td>
<td></td>
<td>Use an inverted color scheme.</td>
</tr>
</tbody>
</table>

Table 16.1: Commands to Create Bitmap Graphics
The finite field activity `-cheat_sheet_GF` creates the file

GF_q7_addition_table.csv

which is used as the input for the second command. The file content is:

```
Row,C0,C1,C2,C3,C4,C5,C6
0,0,1,2,3,4,5,6
1,1,2,3,4,5,6,0
2,2,3,4,5,6,0,1
3,3,4,5,6,0,1,2
4,4,5,6,0,1,2,3
5,5,6,0,1,2,3,4
6,6,0,1,2,3,4,5
END
```

The second command creates the diagram in Figure 16.1. The `-partition` command is used to define an outline of width 3 pixes. The all-in-one partition 7 is used as both row-partition and column-partition.

The planes PG(2,q) admit a cyclic automorphism group known as the Singer cycle. The command

```
PG_2_4_cyclic_incma:
```

```
  $(ORBITER_PATH)orbiter.out --v.2 \n  -define F --finite_field -q.4 --end \n  -define P --projective_space -2:F --end \n  -with P --do --projective_space_activity \n  -cheat_sheet_for_decomposition_by_element_PG \n```
Figure 16.1: Addition table of $\mathbb{F}_7$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

produces a cyclically ordered incidence matrix of the plane $\text{PG}(2, 4)$, shown in Figure 16.2. The Singer cycle is the projectivity defined by the companion matrix of an irreducible polynomial. We may pick the irreducible polynomial $X^2 + X + \omega$ over $\mathbb{F}_4$. The associated Singer cycle is the projectivity defined by the companion matrix

$$
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
2 & 1 & 1
\end{bmatrix}.
$$

The poset classification algorithm from Sections 6.3 and 6.4 computes partially ordered sets. The posets are created using the `-draw_poset` option in the poset classification control command package, see Table 6.2. The posets are stored in a file with extension `.layered_graph`. These files can be drawn using the `-draw_layered_graph` command. The commands in Table 16.2 and Table 16.3 show ways in which to customize the drawings. Let us consider an example. Suppose we are interested in the Schreier trees of a permutation group represented in a Stabilizer chain. We take $\text{PGL}(4, 2)$ in its action on the wedge product. The command
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-file</td>
<td>fname</td>
<td>Use the given file name for output files.</td>
</tr>
<tr>
<td>-xin</td>
<td>a</td>
<td>Assume input $x$-coordinates are in the interval $[0, a]$. Default value: 10000.</td>
</tr>
<tr>
<td>-yin</td>
<td>a</td>
<td>Assume input $y$-coordinates are in the interval $[0, a]$. Default value: 10000.</td>
</tr>
<tr>
<td>-xout</td>
<td>a</td>
<td>Assume output $x$-coordinates are in the interval $[0, a]$. Default value: 1000000.</td>
</tr>
<tr>
<td>-yout</td>
<td>a</td>
<td>Assume output $y$-coordinates are in the interval $[0, a]$. Default value: 1000000.</td>
</tr>
<tr>
<td>-spanning_tree</td>
<td></td>
<td>Place nodes according to a spanning tree. Default value: off.</td>
</tr>
<tr>
<td>-circle</td>
<td></td>
<td>Circle all nodes. Default value: on.</td>
</tr>
<tr>
<td>-corners</td>
<td></td>
<td>Draw corners at the outside of the drawing. Default value: off.</td>
</tr>
<tr>
<td>-rad</td>
<td>r</td>
<td>Use radius $r$ for drawing circles around nodes. Default value: 50.</td>
</tr>
<tr>
<td>-embedded</td>
<td></td>
<td>Create latex headers for stand-alone latex files. Default value: off.</td>
</tr>
<tr>
<td>-sideways</td>
<td></td>
<td>Create latex figure sideways. Default value: off.</td>
</tr>
<tr>
<td>-label_edges</td>
<td></td>
<td>Label the edges in Schreier trees. Default value: off.</td>
</tr>
<tr>
<td>-x_stretch</td>
<td>s</td>
<td>Apply $x$-axis scaling by a factor of $s$. Default value: $s = 1.0$. This option does not affect the drawing of Schreier trees.</td>
</tr>
<tr>
<td>-y_stretch</td>
<td>s</td>
<td>Apply $y$-axis scaling by a factor of $s$. Default value: $s = 1.0$. This option does not affect the drawing of Schreier trees.</td>
</tr>
</tbody>
</table>

Table 16.2: Drawing Options for Layered Graph Files (Part 1)
Figure 16.2: A cyclic ordering of the incidence matrix of PG(2, 4)

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-scale</td>
<td>s</td>
<td>Use tikz global scale-factor of s. Default value: $s = 0.45$.</td>
</tr>
<tr>
<td>-line_width</td>
<td>s</td>
<td>Set tikz line width to s. Default value: $s = 1.5$.</td>
</tr>
<tr>
<td>-nodes_empty</td>
<td></td>
<td>Draw nodes empty. Do not label. Default value: off.</td>
</tr>
<tr>
<td>-select_layers</td>
<td>S</td>
<td>Draw layers whose index is given in the list S only.</td>
</tr>
<tr>
<td>-paths_in_between</td>
<td>$l_1$ $i_1$ $l_2$ $i_2$</td>
<td>Draw all paths from node $(l_1, i_1)$ to node $(l_2, i_2)$. Here, $(l, i)$ is the $i$-th node at layer $l$ (counting from zero). Delete all other edges between layers $l_1$ and $l_2$.</td>
</tr>
</tbody>
</table>

Table 16.3: Drawing Options for Layered Graph Files (Part 2)
Figure 16.3: The first basic orbit of PGL(4, 2) as a subgroup of $\text{PGO}^+(6, 2)$

The command

```
PGL_4_2.Wedge_4_0_detached_graphical_output:
\texttt{\$(ORBITER\_PATH)\texttt{orbiter.out:\v.12:\}}
\texttt{\$\texttt{-define\_G\_linear\_group\_PGL\_4\_2:\}}
\texttt{\$\texttt{-wedge\_detached\_end:\}}
\texttt{\$\texttt{-with\_G\_do:\}}
\texttt{\$\texttt{-group\_theoretic\_activity:\}}
\texttt{\$\texttt{-report:\}}
\texttt{\$\texttt{-end:\}}
\texttt{\$\texttt{pdflatex\_PGL_4_2.Wedge_4_0_detached_report.tex:\}}
\texttt{\$\texttt{open\_PGL_4_2.Wedge_4_0_detached_report.pdf:\}}
```

produces a report about this group action. Figure 16.3 shows the first basic orbit in the stabilizer chain of the group in that action.

The command

```
schreier_tree_graphical_output:
\texttt{\$(ORBITER\_PATH)\texttt{orbiter.out:\v.4:\}}
\texttt{\$\texttt{-draw\_options:\}}
\texttt{\$\texttt{-yout\_500000:\}}
\texttt{\$\texttt{-radius\_15\_nodes\_empty:\}}
\texttt{\$\texttt{-line\_width\_0.5\_y\_stretch\_0.25:\}}
\texttt{\$\texttt{-end:\}}
\texttt{\$\texttt{-define\_G\_linear\_group\_PGL\_4\_2\_end:\}}
\texttt{\$\texttt{-with\_G\_do:\}}
\texttt{\$\texttt{-group\_theoretic\_activity:\}}
\texttt{\$\texttt{-orbits\_on\_polynomials\_3:\}}
```

416
Figure 16.4: A Schreier tree in the action on polynomials

\[ \text{-orbits on polynomials draw tree\-6} \]
\[ \text{-end} \]
\[ \text{pdflatex\-poly\_orbits\_d3\_n3\_q2.tex} \]
\[ \text{open\-poly\_orbits\_d3\_n3\_q2.pdf} \]

draws the 6th Schreier tree in the classification of orbits of $\text{PGL}(4,2)$ on homogeneous polynomials of degree 3 in 4 variables. The drawing is shown in Figure 16.4. This particular orbit has length 420, so there are 420 nodes in the tree.
16.2 The Povray Interface

Orbiter can be used to create raytracing 3D-graphics. Orbiter serves as a front end for the raytracing software Povray [55]. This is a multi step process: A 3D scene is defined through orbiter commands. Next, Orbiter produces Povray files. After that, the povray files are processed through povray, and turned into graphics files (png), called frames. The frames can be turned into a video by using tools like ffmpeg (see Section 16.3). By default, an rotational animation is produced.

The Orbiter Povray interface requires some general information about the animation, the camera position, the boundary box for clipping, the font size for text and others. Tables 16.4-16.5 list the commands to control the 3D-povray frontend. The main part in a 3D graphics is the scene description. This tells the system what will be in the picture. A scene is composed of objects. Various types of objects are available: points, lines, planes, faces, algebraic surfaces, reguli, 3D-text, and others. Some complex objects are predefined, for instance the Hilbert, Cohn-Vossen surface. Once the objects are defined, output commands can be invoked to draw them in various colors and with various options. At times, there are many objects in one scene. In order to make drawing easier, it is possible to group objects. All objects in a group must have the same type. One group of object can be drawn with one command. Tables 16.6 and 16.7 summarize the Orbiter commands to build objects of a 3D scene.

Building the scene itself does not create any graphical output. To this end, the commands in Table 16.8 are used. Each of these commands applies to a group of objects of the same kind. Groups of objects are created using the commands in Table 16.7 which start with group_of. Here is a simple example which combines scene building and graphical output. The example creates a cube with vertices, edges and faces:

cube:

```
$\text{(ORBITER\_PATH)}\text{orbiter.out}\text{-v.2-povray}\$

$\text{\text{-round:0-nb\_frames\_default:30}}$

$\text{\text{-output\_mask\_cube\_\%03d.pov}}$

$\text{\text{-video\_options:-W.1024-H.768}}$

$\text{\text{-global\_picture\_scale:0.5}}$

$\text{\text{-default\_angle:75}}$

$\text{\text{-clipping\_radius:2.7}}$

$\text{\text{-end}}$

$\text{-scene\_objects}$

$\text{-obj\_file-cube\_centered.obj}$

$\text{-edge:0,1}$

$\text{-edge:0,2}$

$\text{-edge:0,4}$

$\text{-edge:1,3}$

$\text{-edge:1,5}$

$\text{-edge:2,3}$

$\text{-edge:2,6}$

$\text{-edge:3,7}$

$\text{-edge:4,5}$
```

418
<table>
<thead>
<tr>
<th>Option</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-do_not_rotate</code></td>
<td></td>
<td>No rotation. By default, the animation consists of a full rotation around a vertical axis.</td>
</tr>
<tr>
<td><code>-rotate_about_z_axis</code></td>
<td></td>
<td>Rotate around z-axis.</td>
</tr>
<tr>
<td><code>-rotate_about_111</code></td>
<td></td>
<td>Rotate around (1,1,1)-axis (default).</td>
</tr>
<tr>
<td><code>-rotate_about_custom_axis</code></td>
<td>axis</td>
<td>Rotate around a custom axis. The axis is specified as a vector of length 3.</td>
</tr>
<tr>
<td><code>-boundary_none</code></td>
<td></td>
<td>Remove the clipping.</td>
</tr>
<tr>
<td><code>-boundary_box</code></td>
<td></td>
<td>Clip using a box shape.</td>
</tr>
<tr>
<td><code>-boundary_sphere</code></td>
<td></td>
<td>Clip using a sphere (default).</td>
</tr>
<tr>
<td><code>-font_size</code></td>
<td>s</td>
<td>Set font size to s.</td>
</tr>
<tr>
<td><code>-stroke_width</code></td>
<td>s</td>
<td>Set text depth to s.</td>
</tr>
<tr>
<td><code>-omit_bottom_plane</code></td>
<td></td>
<td>Remove the bottom plane.</td>
</tr>
<tr>
<td><code>-W</code></td>
<td>w</td>
<td>Set output dimension to w pixels wide.</td>
</tr>
<tr>
<td><code>-H</code></td>
<td>h</td>
<td>Set output dimension to h pixels height.</td>
</tr>
<tr>
<td><code>-nb_frames</code></td>
<td>n</td>
<td>Set number of frames to n. One revolution around the axis is split into n frames.</td>
</tr>
<tr>
<td><code>-zoom</code></td>
<td>r aₙ aₜ cₛ cₜ</td>
<td>Set zoom angle and clipping with in round r to change from aₙ to aₜ and from cₛ to cₜ, respectively.</td>
</tr>
<tr>
<td><code>-pan</code></td>
<td>r F T C</td>
<td>In round r, pan the camera from location F to location T in a rotational movement with center at C. Each of F,T,C are three dimensional coordinates.</td>
</tr>
<tr>
<td><code>-pan_reverse</code></td>
<td>r F T C</td>
<td>Same as <code>-pan</code>, but camera movement is in opposite order.</td>
</tr>
<tr>
<td><code>-no_background</code></td>
<td></td>
<td>Remove background.</td>
</tr>
<tr>
<td><code>-no_bottom_plane</code></td>
<td>r</td>
<td>Remove bottom plane in round r.</td>
</tr>
<tr>
<td><code>-camera</code></td>
<td>r S C L</td>
<td>In round r, set camera location at C, sky at S and pointing towards L. Each of S,C,L are three-dimensional coordinate vectors.</td>
</tr>
</tbody>
</table>

Table 16.4: Options for Orbiter 3D-graphics (Part 1)
<table>
<thead>
<tr>
<th>Option</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-clipping</td>
<td>r c</td>
<td>In round r, set clipping radius to c.</td>
</tr>
<tr>
<td>-text</td>
<td>r a text</td>
<td>In round r, produce running text text with sustain value a.</td>
</tr>
<tr>
<td>-label</td>
<td>r s a g text</td>
<td>In round r, produce running text text with start value s, sustain s and gravity g.</td>
</tr>
<tr>
<td>-latex</td>
<td>r s a praemable g text l fname</td>
<td>In round r, produce running latex text text with start value s, sustain s and gravity g. Put praemable in the latex source code. Use fname for the latex file names (no extension).</td>
</tr>
<tr>
<td>-global_picture_scale</td>
<td>d</td>
<td>Set scaling factor to d.</td>
</tr>
<tr>
<td>-picture</td>
<td>r d fname options</td>
<td>In round r, place picture fname scaled by d using options.</td>
</tr>
<tr>
<td>-picture</td>
<td>r d fname options</td>
<td>In round r, place picture fname scaled by d using options.</td>
</tr>
<tr>
<td>-look_at</td>
<td>L</td>
<td>Override camera look-at value to L. L is a three-dimensional vector.</td>
</tr>
<tr>
<td>-default_angle</td>
<td>a</td>
<td>Set default camera angle to a.</td>
</tr>
<tr>
<td>-clipping_radius</td>
<td>f</td>
<td>Set default clipping radius to f.</td>
</tr>
<tr>
<td>-scale_factor</td>
<td>s</td>
<td>Set default scale factor to s.</td>
</tr>
<tr>
<td>-line_radius</td>
<td>s</td>
<td>Set default line radius to s.</td>
</tr>
</tbody>
</table>

Table 16.5: Options for Orbiter 3D-graphics (Part 2)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cubic_lex</td>
<td>coeffs</td>
<td>Cubic surface given by 20 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-cubic_orbiter</td>
<td>coeffs</td>
<td>Cubic surface given by 20 coefficients in Orbiter ordering</td>
</tr>
<tr>
<td>-cubic_Goursat</td>
<td>A B C</td>
<td>Cubic surface with tetrahedral symmetry given by 3 Goursat coefficients as $Axyz + B(x^2 + y^2 + z^2) + C = 0$</td>
</tr>
<tr>
<td>-quadric_lex_10</td>
<td>coeffs</td>
<td>Quadric surface given by 10 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-quartic_lex_35</td>
<td>coeffs</td>
<td>Quartic surface given by 35 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-octic_lex_165</td>
<td>coeffs</td>
<td>Octic surface given by 165 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-point</td>
<td>coeffs</td>
<td>Point given by three coordinates</td>
</tr>
<tr>
<td>-point_list_from_csv_file</td>
<td>fname</td>
<td>List of points with coordinates given in a csv file</td>
</tr>
<tr>
<td>-line_through_two_points_recentered_from_csv_file</td>
<td>fname</td>
<td>List of lines through two points with point coordinates given in a csv file</td>
</tr>
<tr>
<td>-line_through_two_points_from_csv_file</td>
<td>fname</td>
<td>List of lines through two points with point coordinates given in a csv file</td>
</tr>
<tr>
<td>-point_as_intersection_of_two_lines</td>
<td>$i_1 \ i_2$</td>
<td>Create a point from the intersection of two lines $i_1$ and $i_2$</td>
</tr>
<tr>
<td>-edge</td>
<td>$i_1 \ i_2$</td>
<td>Create an edge (line segment) between points $i_1$ and $i_2$</td>
</tr>
<tr>
<td>-text</td>
<td>$i_1 \ s$</td>
<td>Create a label $s$ located at the point $i$</td>
</tr>
<tr>
<td>-triangular_face_given_by_three_lines</td>
<td>$i_1 \ i_2 \ i_3$</td>
<td>Create a triangular face give by three lines $i_1, i_2, i_3$</td>
</tr>
<tr>
<td>-face</td>
<td>pts</td>
<td>Create a face through the vertices pts, ordered cyclically</td>
</tr>
<tr>
<td>-quadric_through_three_skew_lines</td>
<td>$i_1 \ i_2 \ i_3$</td>
<td>Create a quadric through three skew lines</td>
</tr>
<tr>
<td>-plane_defined_by_three_points</td>
<td>$i_1 \ i_2 \ i_3$</td>
<td>Create a plane through three noncollinear points</td>
</tr>
<tr>
<td>-line_through_two_points_recentered</td>
<td>pt-coords</td>
<td>Create a line through two points given by 6 coordinates, recentered</td>
</tr>
</tbody>
</table>

Table 16.6: Scene definition commands (part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-line_through_two_points</td>
<td>pt-coords</td>
<td>Create a line through two points given by 6 coordinates</td>
</tr>
<tr>
<td>-line_through_two_existing_points</td>
<td>i₁ i₂</td>
<td>Create a line through two points</td>
</tr>
<tr>
<td>-line_through_point_with_direction</td>
<td>x y z uₓ uᵧ uₓ</td>
<td>Create a line through a point ((x, y, z)) with a given direction ((uₓ, uᵧ, uₓ))</td>
</tr>
<tr>
<td>-plane_by_dual_coordinates</td>
<td>a b c d</td>
<td>Create the plane (ax + by + cz + d = 0) given in dual coordinates</td>
</tr>
<tr>
<td>-dodecahedron</td>
<td></td>
<td>Create a Dodecahedron centered at the origin (20 points, 30 edges, 12 faces)</td>
</tr>
<tr>
<td>-Hilbert_Cohn_Vossen_surface</td>
<td></td>
<td>Create the Hilbert, Cohn-Vossen surface (1 cubic surface, 45 tritangent planes, 27 lines)</td>
</tr>
<tr>
<td>-obj_file</td>
<td>fname</td>
<td>Read points and faces from the given .obj file</td>
</tr>
<tr>
<td>-group_of_things</td>
<td>list</td>
<td>Create a group of things from the given list</td>
</tr>
<tr>
<td>-group_of_things_with_offset</td>
<td>list offset</td>
<td>Create a group of things from the given list, each value is increase by offset</td>
</tr>
<tr>
<td>-group_of_things_as_interval</td>
<td>a b</td>
<td>Create a group of things indexed by the interval (a, \ldots, a+b-1)</td>
</tr>
<tr>
<td>-group_of_things_as_interval_with_exceptions</td>
<td>a b ex</td>
<td>Create a group of things indexed by the interval (a, \ldots, a+b-1) with the exceptional elements in the list ex removed</td>
</tr>
<tr>
<td>-group_of_all_points</td>
<td></td>
<td>Create a group of things from all points currently defined</td>
</tr>
<tr>
<td>-group_of_all_faces</td>
<td></td>
<td>Create a group of things from all faces currently defined</td>
</tr>
<tr>
<td>-group_subset_at_random</td>
<td>i f</td>
<td>Create a group of things from the existing group (i) by picking a random subset with probability (f)</td>
</tr>
<tr>
<td>-create_regulus</td>
<td>i N</td>
<td>Create a regulus for quadric (i) with (N) lines</td>
</tr>
</tbody>
</table>

Table 16.7: Scene definition commands (part 2)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-spheres</td>
<td>$i \ r \ prop$</td>
<td>For each element in point group $i$, create a sphere of radius $r$ with given Povray properties.</td>
</tr>
<tr>
<td>-cylinders</td>
<td>$i \ r \ prop$</td>
<td>For each element in edge group $i$, create a cylinder of radius $r$ with given Povray properties.</td>
</tr>
<tr>
<td>-prisms</td>
<td>$i \ d \ prop$</td>
<td>For each element in face group $i$, create a prism of half-thickness $d$ with given Povray properties.</td>
</tr>
<tr>
<td>-planes</td>
<td>$i \ prop$</td>
<td>For each element in plane group $i$, create a plane with given Povray properties.</td>
</tr>
<tr>
<td>-lines</td>
<td>$i \ r \ prop$</td>
<td>For each element in line group $i$, create a line of radius $r$ with given Povray properties.</td>
</tr>
<tr>
<td>-cubics</td>
<td>$i \ prop$</td>
<td>For each element in group $i$ of cubics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-quadrics</td>
<td>$i \ prop$</td>
<td>For each element in group $i$ of quadrics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-quartics</td>
<td>$i \ prop$</td>
<td>For each element in group $i$ of quartics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-octics</td>
<td>$i \ prop$</td>
<td>For each element in group $i$ of octics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-texts</td>
<td>$i \ d \ s \ prop$</td>
<td>For each element in group $i$ of labels, create a text element with half-thickness $d$ and size $s$ with given Povray properties.</td>
</tr>
</tbody>
</table>

Table 16.8: Graphical output commands
This command instructs Orbiter to create 30 povray files (extension .pov), one for each frame of a rotating scene. The scene contains a cube whose vertices are shown in chrome, whose edges are in red, and whose faces are yellow and transparent. The cube turns around a vertical axis of symmetry. Here is the first frame of the result:

```
This command instructs Orbiter to create 30 povray files (extension .pov), one for each frame of a rotating scene. The scene contains a cube whose vertices are shown in chrome, whose edges are in red, and whose faces are yellow and transparent. The cube turns around a vertical axis of symmetry. Here is the first frame of the result:

```

```
The coordinates of the cube are stored in an object file cube_centered.obj. The content of this file is:

```
This command instructs Orbiter to create 30 povray files (extension .pov), one for each frame of a rotating scene. The scene contains a cube whose vertices are shown in chrome, whose edges are in red, and whose faces are yellow and transparent. The cube turns around a vertical axis of symmetry. Here is the first frame of the result:

```

```
The coordinates of the cube are stored in an object file cube_centered.obj. The content of this file is:

```
This command instructs Orbiter to create 30 povray files (extension .pov), one for each frame of a rotating scene. The scene contains a cube whose vertices are shown in chrome, whose edges are in red, and whose faces are yellow and transparent. The cube turns around a vertical axis of symmetry. Here is the first frame of the result:

```

```
The coordinates of the cube are stored in an object file cube_centered.obj. The content of this file is:

```
The monkey saddle is a cubic surface, given by the equation
\[ z = x^3 - 3xy^2 \]

The next example plots the surface knowns as the monkey saddle. The tangent plane at (0,0,0) is drawn as well. An animation is created by rotating the scene around the z-axis.

```
MONKEY_SADDLE_CUBIC="1,0,0,0,-3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0"

monkey:
  > $(ORBITER_PATH)orbiter.out -v.2 -povray -
  > -round.0 -nb_frames_default.30 -
  > -output_mask-monkey_%d_%03d.pov -
  > -video_options -W.1024 -H.768 -
  > -global_picture_scale.8 -
  > -default_angle.75 -
  > -clipping_radius.0.8 -
  > -camera.0.0,0,1.1,1,0.5.0,0,0 -
  > -rotate_about_z_axis -
  > -end -
  > -scene_objects -
  >   > -cubic_lex $(MONKEY_SADDLE_CUBIC) -
  >   > -plane_by_dual_coordinates.0,0,1,0 -
  >   > -group_of_things.0 -
  >   > -group_of_things.0 -
  >   > -cubics.0.texture{pigment.Gold}.finish -
  >     {ambient.0.4.diffuse.0.5.roughness.0.001 -
  >       reflection.0.1.specular.8}.} -
  >   > -planes.1.texture{pigment.color.Blue -
  >     transmit.0.5}.finish{diffuse.0.9.phong.0.2} -
  >   > -scene_objects_end -
  > > -povray_end -
  > -rm -rf POV -
  > mkdir POV -
  > mv monkey_0_* .pov POV
```

425
Here is one of the frames that are created:

The Eckardt surface is given by the equation

$$\frac{5}{2}xyz - (x^2 + y^2 + z^2) + 1 = 0.$$  

We use the following code to plot the surface and the lines on it. The Schlafli labeling of the lines is indicated.

Eckardt:

```
> mv-makefile_animation-POV

$ (ORBITER_PATH) orbiter.out -v -2 -povray \
> -round 0 -nb_frames_default 30 \
> -output_mask Eckardt_%d_%03d.pov \
> -video_options -W 1024 -H 768 \
> -global_picture_scale 0.9 \
> -default_angle 75 \
> -clipping_radius 2.4 \
> -camera 0: "1,1,1": "-3,1,3": "0.12,0.12,0.12" .. \
> -end \
> -scene_objects \
>   -Hilbert_Cohn_Vossen_surface \
>   -group_of_things "0" . \
>   -cubics 0 "texture { pigment { White * 0.5 transmit 0.5 } \
finish { ambient 0.4 diffuse 0.5 roughness 0.001 reflection 0.1 specular 0.8 } }" . \
>   -group_of_things_as_interval 0.6 \
>   -group_of_things_as_interval 6.6 \
>   -group_of_things_as_interval_with_exceptions 12 15 \
>   -lines 1 0.02 "texture { pigment { color Red } } "
```
Figure 16.5 shows the final product.
Figure 16.5: The Eckardt surface
The Endrass octic \([25]\) is the algebraic surface given by the equation

\[
x_8 = 64 (-w^2 + x^2) (-w^2 + y^2) ((x+y)^2 - 2 w^2) ((x-y)^2 - 2 w^2) - (-4 (1 + \sqrt{2}) (x^2 + y^2)^2 + (8 (2 + \sqrt{2})^2 + 2 (2 + 7 \sqrt{2}) w^2) (x^2 + y^2) - 16 z^4 + 8 (1 - 2 \sqrt{2}) z^2 w^2 - (1 + 12 \sqrt{2}) w^4)^2
\]

The following Orbiter command creates a povray graphics of the octic, shown in Figure 16.6:

```
ENDRASS_OCTIC_LEX_165="-93.2548,0,0,0,-309.019,0,0,527.529,0,395.647,\n 0,0,0,0,0,0,0,0,0,-687.529,0,0,1582.59,0,1186.94,0,0,0,0,-1055.06,0,\n-1582.59,0,-593.47,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-309.019,\n0,0,1582.59,0,1186.94,0,0,0,0,-2110.12,0,-3165.17,0,-1186.94,0,0,0,0,\n0,874.039,0,1560.63,0,1677.92,0,343.362,0,0,0,0,0,0,0,0,0,0,0,0,0,\n0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\n-1582.59,0,-593.47,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\n0,0,0,0,-1055.06,0,-1582.59,0,-593.47,0,0,0,0,0,0,874.039,0,1560.63,0,\n1677.92,0,343.362,0,0,0,0,0,0,0,-256.0,-468.077,0,-789.019,0,\n-525.726,0,0.941125"
```

```
endrass8:
▷ $(ORBITER_PATH)orbiter.out -v -povray -
▷ -round:0 -nb_frames_default:30 -
▷ -output_mask-endrass_octic_%d.png -
▷ -video_options:W:1024:H:768 -
▷ -global_picture_scale:0.75 -
▷ -default_angle:75 -
▷ -clipping_radius:3.7 -
▷ -no_bottom_plane -
▷ -camera:0:"1,1,1"."6,6,3"."0,0,0" .. -
▷ -rotate_about:111 -
▷ -end -
▷ -scene_objects -
▷ ▷ -line_through_two_points_recentered_from_csv_file -
▷ ▷ ▷ coordinate_grid.csv -
▷ ▷ ▷ -group_of_things:0 -
▷ ▷ ▷ -group_of_things:1 -
▷ ▷ ▷ -group_of_things:2 -
▷ ▷ ▷ -group_of_things:as_interval:3.39 -
▷ ▷ ▷ -lines:0.0.15:"texture{pigment{.color:Red}.} finish{diffuse:0.9-phon:1}" -
▷ ▷ ▷ -lines:1.0.15:"texture{pigment{.color:Green}.} finish{diffuse:0.9-phon:1}" -
▷ ▷ ▷ -lines:2.0.15:"texture{pigment{.color:Blue}.} finish{diffuse:0.9-phon:1}" -
▷ ▷ ▷ -lines:3.0.05:"texture{pigment{.color:Black}.} finish{diffuse:0.9-phon:1}" -
▷ ▷ ▷ -octic_lex_165:$(ENDRASS_OCTIC_LEX_165) -
```

429
Figure 16.6: The Endrass Octic

This illustration includes coordinate axes and the \( x, y \)-plane.
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-i</td>
<td>s l mask</td>
<td>Specify the input file names by running a printf command with the given mask applied to the index i where i goes from s to $s + l - 1$. This option can be repeated.</td>
</tr>
<tr>
<td>-step</td>
<td>s</td>
<td>Increment the index in steps of size s.</td>
</tr>
<tr>
<td>-o</td>
<td>mask</td>
<td>Create the output file using the given mask.</td>
</tr>
<tr>
<td>-output_starts_at</td>
<td>i</td>
<td>Start output file indices at i (default is 0).</td>
</tr>
</tbody>
</table>

Table 16.9: Prepare frames commands

16.3 Creating Animations

Orbiter can be used to create animations. This relies on the software ffmpeg. In a first step, all frames (i.e. individual graphics files) are created using Orbiter’s povray interface. After that, the frames are used to create the animation. In order to use ffmpeg, the frames should have a uniform file naming scheme, using a consecutive numbering to arrange the files in order. This is achieved by using a printf style mask, with `%d` representing the number of the current frame. In order to do so, Orbiter can be used to copy and rename files. A temporary directory can be used to collect the files. The Orbiter command `prepare_frames` can be used. For a list of commands, see Tables 16.9. For instance, the command

```
monkey_video:
  -rm-r.FRAMES
  -mkdir.FRAMES
  -rm monkey.mp4
  $(ORBITER_PATH)orbiter.out/\n  -prepare_frames/\n  -i 0-30 monkey_0_%03d.png/\n  -output_starts_at 0/\n  -o FRAMES/frame%04d.png/\n  -end
  ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png/\n  -f mp4 -q:v 0 -vcodec mpeg4 monkey.mp4
```

creates a video `monkey.mp4` from a set of 30 files. The individual filenames are created using the printf format string `monkey_0_%03d.png`, with an integer index that is drawn from the interval $[0, 29]$. The part that starts with a percent sign and ends with a “d” character defines the way in which the integer is formatted. The number three before the “d” indicates that three characters will be printed. The zero indicates the use of leading zeros. So, the first file would be `monkey_0_000.png` and the very last file is `monkey_0_029.png`. The description of the printf format string can be found in the documentation of the C standard library [38].
16.4 Continuous Function Plotter

Orbiter can plot functions using a built-in function tracker. The functions must be continuous apart from a finite number of poles. The function can have multiple components, each described using an expression. Each expression is specified in Reverse Polish Notation (RPN). Consider an example. A Lissajous curve is defined using coordinate functions of the form

\[ x = r \sin (at + c), \quad y = r \sin (bt), \quad a, b, c, r \in \mathbb{R}. \]

The terms \( r \sin (at + c), \quad r \sin (bt) \) are the expressions of the two coordinate functions. RPN means that the operator is listed after the operands. A stack data structure is used to hold temporary values. Operators are pushed to the top of the stack using the push commands. A binary operator pops the two elements from the stack, performs the operation, and pushes the resulting value back onto the stack. For a unary operator, only one element is popped and replaced by the result. Here are some examples of expressions rewritten in RPN:

\[ \sin(x) \mapsto \text{push } x \text{ sin}, \]
\[ a + b \mapsto \text{push } a \text{ push } b \text{ add}, \]
\[ a \cdot b \mapsto \text{push } a \text{ push } b \text{ mult}. \]

The coordinate functions are enclosed between \texttt{-code} and \texttt{-code_end} commands. Each coordinate function is described in RPN and terminated using a \texttt{return} keyword. By the time the \texttt{return} keyword is reached, the RPN expression must have exactly one value on the stack which is considered the value of the expression. Constants are declared between the \texttt{-const} and \texttt{-const_end} keywords. Likewise, variables are declared between the \texttt{-var} and \texttt{-var_end} keywords. Picking \( a = 3, \quad b = 2, \quad c = \pi/2 \) and \( r = 7 \), the function is computed using

\texttt{lissajous:}
\[ \texttt{\$(ORBITER\_PATH)orbiter.out-\_v.2\_}\]
\[ \texttt{\_smooth\_curve:"lissajous"\_0.07\_2000\_15\_0.18.85\_}\]
\[ \texttt{\_const\_a\:3\_b\:2\_c\:1.57\_r\:7\_\_const\_end\_}\]
\[ \texttt{\_var\_t\_\_var\_end\_}\]
\[ \texttt{\_code\_}\]
\[ \texttt{\_push\_t\_push\_a\_mult\_push\_c\_add\_sin\_push\_r\_mult\_return\_}\]
\[ \texttt{\_push\_t\_push\_b\_mult\_sin\_push\_r\_mult\_\_return\_}\]
\[ \texttt{\_code\_\_end\_}\]

The sequence
\[ \texttt{push } t \text{ push } a \text{ mult push } c \text{ add sin push } r \text{ mult} \]
is \( r \sin(at + c) \) expressed in RPN. The constants are defined in the line

\[ \texttt{-const a 3 b 2 c 1.57 r 7 -const\_end} \]
The input variable is defined using the line

```
-var t -var_end
```

The sequence

```
-smooth_curve "lissajous" 0.07 2000 15 0 18.85
```

defines the name of the output file, the fact that two consecutive points are never further than $\epsilon = 0.07$ away, the fact that points that are 15 or more away from the origin should be ignored, and the fact that the variable $t$ loops over the range $[0, 18.85]$ with a default of 2000 steps. The evaluator automatically reduces the step-size if consecutive image points are more than $\epsilon$ apart. The code to produce the plot is

```
lißajous_plot:
  ▶ $(ORBITER_PATH)orbiter.out -v.2 -povray -
  ▶ -round:0 -nb_frames_default:1 -
  ▶ -output_mask:lißajous_%d_%03d.pov -
  ▶ -video_options:-W:1024 -H:768 -
  ▶ -global_picture_scale:0.40 -
  ▶ -default_angle:45 -
  ▶ -clipping_radius:5 -
  ▶ -omit_bottom_plane -
  ▶ -camera:0:0,-1,0:0,0,12:0,0,0 -
  ▶ -rotate_about_z_axis -
  ▶ -end -
  ▶ -scene_objects -
  ▶ ▶ -line_through_two_points_recentered_from_csv_file -
  ▶ ▶ coordinate_grid.csv -
  ▶ ▶ -group_of_things:0 -
  ▶ ▶ -group_of_things:1 -
  ▶ ▶ -group_of_things:2 -
  ▶ ▶ -lines:0:0.09:"texture{pigment{color:Yellow}.}" -
  ▶ ▶ -lines:1:0.09:"texture{pigment{color:Yellow}.}" -
  ▶ ▶ -lines:2:0.09:"texture{pigment{color:Yellow}.}" -
  ▶ ▶ -group_of_things_as_interval:3.39 -
  ▶ ▶ -lines:3:0.02:"texture{pigment{color:Black}.}" -
  ▶ ▶ -point_list_from_csv_file -
  ▶ ▶ function_lissajous_N2000_points.csv -
  ▶ ▶ -group_of_things_as_interval:0.6524 -
  ▶ ▶ -spheres:4:0.1:"texture{pigment{color:Red}.}" -
  finish{diffuse:0.9:phong:1}"
  ▶ ▶ -plane_by_dual_coordinates:0,0,1,0:0 -
  ▶ ▶ -group_of_things:0 -
  ▶ ▶ -planes:5:"texture{pigment{color:Blue*0.5:transmit:0.5}.}" -
```

433
The plot is shown in Figure 16.7.

We can turn it into a 3D plot by using the $t$ value for the $z$ coordinate. The function is computed using the command

```
lissajous_3d:
  $(ORBITER_PATH)orbiter.out -v 2 -povray
  -smooth_curve:"lissajous_3d"-0.07:2000:50:0:18.85
  -var:t:-var_end
  -code
  push:t:return
  -code_end
```

The code to produce the 3D plot is

```
lissajous_3d_plot:
  $(ORBITER_PATH)orbiter.out -v 2 -povray
  -round:0 -nb_frames_default:30
```
The 3D curve is shown in Figure 16.8.
Figure 16.8: Lissajous Spacecurve
Chapter 17

Miscellaneous

17.1 Miscellaneous

Table 17.1 lists miscellaneous Orbiter commands. The command `-csv_file_select_rows` can be used to select rows from a csv file. The command `-csv_file_select_cols` can be used to select columns from a csv file. The command `-csv_file_select_rows_and_cols` selects rows and columns. Here is an example. We create the multiplication table of the finite field \( \mathbb{F}_7 \), ordered according to the powers of a primitive element:

\[
\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5.
\]

After that, we pull the rows and columns corresponding to even powers \( \alpha^0, \alpha^2, \alpha^4 \).

```plaintext
misc_select:
▷ $(ORBITER_PATH)orbiter.out -v 3 \$
▷▷ -define F -finite_field -q 7 -end \$
▷▷ -with F -do finite_field_activity cheat_sheet_GF -end
▷ $(ORBITER_PATH)orbiter.out -v 4 -csv_file_select_rows_and_cols \$
▷▷ GF_q7_multiplication_table_reordered.csv \$
▷▷ "0,2,4" "0,2,4".
```

The even powers of \( \alpha \) create a multiplicative subgroup. Figure 17.1 shows the table of the multiplicative group \( \mathbb{F}_7^* \) and the subgroup of squares (compare with Figure 3.4 in Section 3.2). Here is the file `GF_q7_multiplication_table_reordered.csv`:

<table>
<thead>
<tr>
<th>Row, C0, C1, C2, C3, C4, C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1, 3, 2, 6, 4, 5</td>
</tr>
<tr>
<td>1, 3, 2, 6, 4, 5, 1</td>
</tr>
<tr>
<td>2, 2, 6, 4, 5, 1, 3</td>
</tr>
<tr>
<td>3, 6, 4, 5, 1, 3, 2</td>
</tr>
<tr>
<td>4, 4, 5, 1, 3, 2, 6</td>
</tr>
<tr>
<td>Option</td>
</tr>
<tr>
<td>------------------------------</td>
</tr>
<tr>
<td>-csv_file_select_rows</td>
</tr>
<tr>
<td>-csv_file_select_cols</td>
</tr>
<tr>
<td>-csv_file_select_rows_and_cols</td>
</tr>
<tr>
<td>-csv_file_join</td>
</tr>
<tr>
<td>-csv_file_latex</td>
</tr>
<tr>
<td>-store_as_csv_file</td>
</tr>
</tbody>
</table>

| Table 17.1: Miscellaneous Orbiter Commands |

Figure 17.1: Cyclic multiplication table of $\mathbb{F}_7$ and subgroup of squares
and next the file that is created by selecting rows and columns 0, 2, 4:

```
Row, "C0", "C2", "C4"
0, "1", "2", "4"
1, "2", "4", "1"
2, "4", "1", "2"
END
```
17.2 Limitations

Several limitations exist in Orbiter. Here is a list:

1. Field elements are encoded as int. This limits the size of fields that can be handled to $2^{8s-1}$ where $s = \text{sizeof(int)}$.

2. The ranks of elements in the permutation domain are encoded as long int. This limits the size of permutation domains that can be handled. The degree of a permutation group must be less that $2^{8s-1}$ where $s = \text{sizeof(long int)}$.

3. The finite field class builds tables for the addition and multiplication of field elements. This restricts the size of the fields that can be created.

4. The projective geometry class tries to build a bitmatrix for the adjacency matrix if the number of lines is less than MAX_NUMBER_OF_LINES_FOR_INCIDENCE_MATRIX which is defined in src/lib/foundations/geometry/projective_space.cpp. If the number of lines is too big, the table is not created. In this case, the projective geometry class may behave slower.

5. The projective geometry class tries to build a table for the lines if the number of points is less that MAX_NUMBER_OF_POINTS_FOR_POINT_TABLE and the number of lines is less than MAX_NUMBER_OF_LINES_FOR_LINE_TABLE, both of which are defined in src/lib/foundations/geometry/projective_space.cpp. If the number of points is too big, the table is not created. In this case, the projective geometry class may behave slow.

6. The projective geometry class tries to build a table for the lines through any two points if the number of points is less than MAX_NB_POINTS_FOR_LINE_THROUGH_TWO_POINTS_TABLE which is defined in src/lib/foundations/geometry/projective_space.cpp. If the number of points is too big, the table is not created. In this case, the projective geometry class may behave slow.

7. The projective geometry class tries to build a table for the intersection points of pairs of lines if the number of points is less than MAX_NB_POINTS_FOR_LINE_INTERSECTION_TABLE which is defined in src/lib/foundations/geometry/projective_space.cpp. If the number of points or lines is too big, the table is not created. In this case, the projective geometry class may behave slow.

8. For Windows users: Cygwin by default uses 32 bit integers for both int and long int. Using Cygwin 64 to compile Orbiter recommended.

9. A limited list of primitive polynomials are hard-coded in Orbiter. For large fields, the user must provide their own primitive polynomial. The polynomials encoded in orbiter are not guaranteed to be compatible with the subfield relationship.
Chapter 18

Orbiter on Windows

18.1 Using Windows Subsystem Linux

The following quote from https://docs.microsoft.com/en-us/windows/wsl/ summarizes the function of the Windows Subsystem for Linux:

Windows Subsystem for Linux (WSL) lets developers run a GNU/Linux environment – including most command-line tools, utilities, and applications – directly on Windows, unmodified, without the overhead of a traditional virtual machine or dual-boot setup. You can:

1. Choose your favorite GNU/Linux distributions from the Microsoft Store.
2. Run common command-line tools such as grep, sed, awk, or other ELF-64 binaries.
3. Run Bash shell scripts and GNU/Linux command-line applications including:
4. Tools: vim, emacs, tmux
5. Languages: NodeJS, Javascript, Python, Ruby, C/C++, C# & F#, Rust, Go, etc.
6. Services: SSHD, MySQL, Apache, lighttpd, MongoDB, PostgreSQL.
7. Install additional software using your own GNU/Linux distribution package manager.
8. Invoke Windows applications using a Unix-like command-line shell.
9. Invoke GNU/Linux applications on Windows.

The following set of slides will illustrate the installation of Orbiter under WSL.
Resources

• Many of the steps will be taken from the following sources:
  • https://okunhardt.github.io/documents/Installing_WSL.pdf
  • https://docs.microsoft.com/en-us/windows/wsl/basic-commands
• Consult the two links for further help and suggestions.

Installing WSL

• Search “Turn Windows features on or off” in the Windows search bar
• Search for “Windows Subsystem for Linux”, the box must be checked
• Restart the computer
Update

• The Windows Subsystem for Linux kernel does not automatically update due to system settings
• Updates must be done manually
• To update, first you need to command prompt as admin
  • Press Windows + R to open the “Run” box
  • Type “cmd” into the box
  • Press Ctrl + Shift + Enter
  • When the window prompt opens, click “Yes”
  • Command prompt will now open as admin
• In command prompt
  • Type `wsl --update`
  • Type `wsl --shutdown`

WSL1, WSL2

• When using WSL, you can adjust the configurations according to the Linux distribution that you are using
• To run Ubuntu distribution, we need the WSL1 configuration
• To check the status, in the command prompt enter
  • `wsl --status`
• To change WSL configuration type
  • `wsl --set-default-version 1`
  • `wsl --shutdown`
Ubuntu - installation

- Generally, the Ubuntu distribution is installed by default when WSL is installed
  - `wsl --status`
    - Displays the default distribution
- If you find that Ubuntu was not installed, you can find it in the Microsoft store
- Launch Ubuntu after installation

Ubuntu - launching

- After launching Ubuntu, allow the installation to be initiated
- If you receive an error, this could be a result of the configuration
  - Set configuration to WSL1
    - `wsl --set-default-version 1`
  - Make sure to terminate Ubuntu and reboot
    - `wsl --terminate Ubuntu`
  - Start Ubuntu again
- Once Ubuntu starts correctly
  - Create Username & Password to complete installation
  - Note: the password will not appear when you type it
Ubuntu - update

- Ubuntu does not update automatically, to update run the command
  - `sudo apt update && sudo apt upgrade`
- You will be prompted to enter your password
- When update is ready to be installed the message will appear
  - Do you want to continue? [Y/n]
  - Y + enter

Ubuntu – g++ and make

- At this point, you have successfully installed and setup WSL, and now you can use the terminal as you would on Ubuntu
- Terminate and reboot Ubuntu
- Run the command in Ubuntu
  - `sudo apt install g++`
  - You can now compile C++ in WSL
- Run the command in Ubuntu
  - `sudo apt install make`
  - You can now use makefiles in WSL
Orbiter - installation

• The easiest way to run make is through the command prompt, not Ubuntu
• To run WSL commands in command prompt, use either
  • wsl <command>
  • wsl.exe <command>
• Open command prompt
• Change directory to Users\username
  • cd C:\Users\"your username"

Orbiter - installation

• In web, go to https://github.com/abetten/orbiter
• Click on the green icon “Code” that opens a drop-down menu
• You want to copy HTTPS URL
Orbiter - installation

• In command prompt, once you are in C:\Users\Joel type the command
  • wsl.exe git clone https://github.com/abetten/orbiter.git
  • Hit enter

• Now, orbiter will begin the cloning process

Orbiter - compile

• After cloning orbiter, run the command
  • dir

• You will find a new directory created called “orbiter”
• Change directory to “orbiter”
  • cd orbiter
Orbiter - compile

• Now that you are in C:\Users\"your username\"\orbiter, run the command
  • wsl.exe make
• The orbiter library will now be compiled, give it some time

![Image showing command output](image1)

Makefile

• Now that orbiter has been successfully compiled, in the directory C:\Users\"your username\"\orbiter
  • Change directory to C:\Users\"your username\" and create a new directory
  • Ex: mkdir CPP_Workspace
• Change directory into CPP_Workspace
  • cd CPP_Workspace
• In C:\Users\"your username\"\"new directory\”, run the command
  • wsl.exe vim makefile
• Vim (an IDE) will create the file “makefile”
• For Vim commands, go to [https://vim.rtorr.com/](https://vim.rtorr.com/)
• Remember: all Ubuntu commands must begin with either
  • wsl or wsl.exe
Makefile

- To edit file in vim, click “i”
- You will see --insert-- in the lower left-hand corner
- The example to the right demonstrates a simple test to assure that orbiter is running correctly
- Assuming that orbiter directory is located in C:\Users\"your username" then the variable OP and ORBITER_PATH should work just fine
- Note were wsl.exe is inserted
- Makefile contains Ubuntu commands not windows commands

Running makefile

- Now that you have created the makefile, click “esc” to finish editing in vim
- Run the command
  - :wa + enter
  - This saves & closes the makefile in vim
- You will be returned to
  - C:\Users\"your username"\“new directory”
- In the directory run,
  - wsl.exe make test
  - Hit “enter”
- If everything runs correctly, you will see
Orbiter - notes

• Now that everything runs correctly, visit https://www.math.colostate.edu/~betten/orbiter/users_guide.pdf
• This is the Orbiter User’s guide
• Remember that you must use “wsl.exe make <target>” or “wsl make <target>” to run make correctly on linux distribution
• Also, note how “wsl.exe” is used inside of the makefile
• Ubuntu commands are used in makefile

Orbiter - update

• To update orbiter, change directories to
  • C:\Users\"your username\"\orbiter
• Run the commands
  • wsl.exe make clean ; wsl.exe make
• Good luck!
Chapter 19

The Makefile

19.1 The Makefile

#MY_PATH=./orbiter
MY_PATH="/DEV.22/orbiter"
#MY_PATH=/scratch/betten/COMPILE/orbiter

# uncomment exactly one of the following two lines.
# uncomment the first if you want to run orbiter through docker.
# uncomment the second if you have an installed copy of orbiter and you want to run it directly.
#ORBITER_PATH=docker run -it --volume ${PWD}:/mnt -w /mnt betten/orbiter/
ORBITER_PATH=$(MY_PATH)/src/apps/orbiter/

#############################################################################
# additional configurations for when you want to compile automatically generated code
#############################################################################
SRC=$(MY_PATH)/src
MY_CPP=g++
MY_CC=gcc
CPPFLAGS=-Wall -I .. /DEV.22/orbiter/src/lib -std=c++14
LIB=$(SRC)/lib/liborbiter.a -lpthread
LFLAGS=-lm -Wl, -rpath -Wl, /usr/local/gcc-8.2.0/lib64

#############################################################################
# End of configuration part
#############################################################################
GINAC_PATH=$(MY_PATH)/src/apps/ginac
SANDBOX_PATH=$(MY_PATH)/src/apps/sandbox

update:
  ▶ cd $(ORBITER_PATH); make clean;
  ▶ cd $(MY_PATH); make cleana; git pull; make

update_all:
  ▶ cd $(MY_PATH); make clean; git pull; make

sandbox:
  ▶ $(SANDBOX_PATH)/sandbox.out

******************************************************************************
# Makefile Variables
******************************************************************************

# MAGMA_PATH=/usr/local/magma
MAGMA_PATH=

V7_VANDERMONDE_EXTENDED="
1,0,0,0,0,0,1,0,0,0,0,0,0,.
1,1,1,1,1,0,1,0,0,0,0,0,0,.
1,2,4,1,2,4,1,0,0,1,0,0,0,0,.
1,3,2,6,4,5,1,0,0,0,1,0,0,0,.
1,4,2,1,4,2,1,0,0,0,0,1,0,0,0,.
1,5,4,6,2,3,1,0,0,0,0,0,1,0,0,0,.
1,6,1,6,1,6,1,0,0,0,0,0,0,0,1"

# Co3 from Conway et al., 1985 (ATLAS)
# order = 495766656000
# Co3 from the paper by Suleiman and Wilson 1997

CONWAY_GEN1="
1101110001000001010000
1111010111110100001011
0000001000000100010101
1111100110110001001110
0101010000000010011101
0000010000000100010101"
CONWAY
  GEN2="\n  0101000101111010111111\n  011010100011110110000\n  011101000011111010111\n  00110111000101100111\n  100010010010010110010\n  11010000001010100011\n  110010101000111100101\n  10001101001101010101\n  0100110001010000001111\n  11000000101001010010\n  0101110110011000000101\n  0101111101010011111001\n  1000010101010101010001\n  001010000111100010111\n  0011010010111011001111\n  0100110010110011111010\n  1101011001111001100011\n  010010010010010001000001\n  1100101000100111001111\n  0101110110010100000001\n  0000001101111000101110\n  1101011001111001100011\n  1101010101010101010001"

HIRSCHFELD SURFACE EQUATION="0,0,0,0,0,1,0,1,0,0,0,0,0,0,0"
EC_11_EQUATION="1,0,3,0,0,0,10,1,0,0"


GEN_C13="1,2,3,4,5,6,7,8,9,10,11,12,0"
#(0,1,2,3,4,5,6,7,8,9,10,11,12)

GENERATORS_HESSE_GROUP="\n3000300030\n2000201230\n1000100111\n1000220200\n1002312010\n0331003211\n2200011331"

GENERATORS_WEYL_GROUP_E8="\n-1,-1,-1,-1,0,0,0,0,\n0,0,0,1,0,0,0,0,\n1,0,0,0,0,0,0,0,\n0,0,1,0,0,0,0,0,\n0,1,0,1,1,0,0,0,\n0,0,0,0,1,0,0,0,\n0,0,0,0,0,1,0,0,\n0,0,0,0,0,0,1,0,\n-1,0,-1,-1,-1,-1,-1,-1,\n0,1,0,1,1,1,1,1,\n1,0,0,0,0,0,0,0,\n0,0,1,0,0,0,0,0,\n0,0,0,1,0,0,0,0,\n0,0,0,0,1,0,0,0,\n0,0,0,0,0,1,0,0,\n0,0,0,0,0,0,1,0"

 Ree_gen1="21,5,1,6,17,1,1,3,13,5,21,6,6,18,21,3,21,21,22,6,14,\n14,18,1,5,13,6,7,3,3,2,1,24,16,3,17,3,22,10,16,24,26,\n21,21,6,18,20,2,5"
ELEMENTARY_SYMMENTRIC_4_3="x0*x1*x2+x0*x1*x3+x0*x2*x3+x1*x2*x3"

ELEMENTARY_SYMMENTRIC_4_4="x0*x1*x2*x3"

CODE_5_2_3_CODEWORDS="0,7,25,30"

SURFACE_F7_15_LINES_POINTS="0,1,2,3,5,6,7,8,9,10,12,18,20,26,28,34,36,42,44,45,50,53,58,59,60,61,62,63,64,67,77,80,90,93,103,107,115,118,122,125,127,142,147,155,157,162,165,170,172,204,208,219,229,240,244,246,251,253,259,277,278,281,286,298,300,302,303,310,312,316,340,343,351,354,358,365,369,372,373,379,384,386,388,393,399"

SURFACE_MCKEAN_15_LINES_POINTS="0,1,2,3,4,5,6,7,8,9,10,16,17,28,35,36,59,60,61,62,63,64,65,76,80,91,95,106,107,111,119,121,122,130,138,139,141,146,150,154,155,158,170,175,184,186,199,203,204,206,226,231,234,239,240,252,253,254,278,279,282,287,299,301,302,319,320,330,338,343,345,350,351,357,364,370,371,376,378,382,385,388,392,394,399"

SURFACE_MCKEAN_15_LINES="1,5,1,-1,16,-2,17,1,10,2,18,1,8,1,9,-1,11,-1,19,-1,14"

CODE_RS_6_4_7="621000\n062100\n006210\n000621"

CODE_RS_10_8_11="7,2,1,0,0,0,0,0,0,0,\n0,7,2,1,0,0,0,0,0,0,\n0,0,7,2,1,0,0,0,0,0,\n0,0,0,7,2,1,0,0,0,0,\n0,0,0,0,7,2,1,0,0,0,\n0,0,0,0,0,7,2,1,0,0,\n0,0,0,0,0,0,7,2,1,0,\n0,0,0,0,0,0,0,7,2,1."

#Dickson:

D1="1,4,1,7,1,12,1,15"
D2="1,1,1,2,1,3,1,4,1,7,1,11,1,14,1,15,1,19"
D3="1,4,1,7,1,12,1,14,1,15,1,19"
293 F_ALPHA_BETA_GAMMA_DELTA="beta*(gamma+1)*x0*x0*x2\
294 + (alpha+delta-beta*gamma+alpha-beta-delta-1)*x0*x1*x2\
295 -1*(alpha*beta-alpha*delta+delta)*(gamma+1)*x0*x1*x3\
296 + (alpha+delta+alpha*gamma-beta*gamma-b*gamma+delta-delta)*x0*x2*x2\
297 - (alpha+delta+beta-delta)*(gamma+1)*x0*x2*x3\
298 - (delta+1)*(alpha-1)*x1*x1*x2\
299 - (delta+1)*(alpha-1)*x1*x1*x3\
300 + (alpha+delta-alpha*gamma+beta*gamma+beta-delta-gamma)*x1*x2*x2\
301 + (alpha+beta*gamma-alpha*beta+alpha+delta)*\x1*x3*x3"
302
303 #general-normal-form-for-surfaces-with-27-lines:
304 F_abcd_eqn="-(a*b*c-a*b*d-a*c*d+a*d-b*c)*(b-d)*X0*X0*X2\n305 +(a*b*c-a*b*d-a*c*d+a*d-b*c)*(a+b-c-d)*X0*X1*X2\n306 + (a*a*c-a*a*d-a*c*c+b*c*c+a*d-b*c)*(b-d)*X0*X1*X3\n307 -(a*d-b*c)*(a*b+c-a*b-d-a*c*d-b*c*d+a*d-b*c)*X0*X2*X2\n308 -(a*a*c-d-a*b*c-c-a*d+a*b*d+b*c*c-b*c*d)*(b-d)*X0*X2*X3\n309 -(a*c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X1*X1*X2\n310 -(a*c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X1*X1*X3\n311 +(a*d-b*c)*(a*b*c-a*b*d-a*c*d+b*c*d+a*d-b*c)*X1*X2*X2\n312 +(1+1)*(a*a*b*c*d-a*a*b*d-d-1+1)*a*a*c*d*d\n313 -(1+1)*(a*b*b*c*c-a*b*b*c*d+a*a*c*d+1+1)*a*b*c*c*d+a*a*b*c*d\n314 -b*b*c*c*d-a*a*b*c*c+a*a*c*d+a*a*d*d+a*b*b*c*c+a*b*c\n315 -(1+1+1+1)*a*b*c*d-a*c*c*d+a*c*d+a*b*b*c*c)*X1*X2*X3\n316 +c*a*(a*d-b*c-a*b+c-d)*(b-d)*X1*X3*X3"
317
318 #proposed-normal-form-for-smooth-cubic-surfaces-with-9-lines:
319 F_a_b_c_d_f_g="g*b*(c*f+d*f-d*g-c+f-a)*x0*x0*x2\n320 + (a*d+g+g-b*c*f+*b*d+f+b*d+g*g\n321 -a*c*f+a*c*g-a+d*f+a*g+g+b*c*g-b+f+f+b*g+g-a+f)*x0*x1*x2\n322 +(1+c+d)*f*(a*g+b+f+b*g-a)*x0*x1*x3\n323 +g*c+f+d*f-d*g-c+f-g)*a*x0*x2*x2\n324 -b*a*g*(c+f+*d+f-d*g-c+f-g)*x0*x2*x3\n325 -(1+d)*f*(a*g+*b+f+b*g-a)*x1*x1*x2-(1+d)*f*(a*g+*b+f+b*g-a)*x1*x1*x3\n326 -c*f*(a*g+b+f+b*g-a)*x1*x2*x2\n327 + (d-1)*c*f*(a*g+b+f+b*g-a)*x1*x2*x3\n328 +c*d*f*(a*g+*b+f+b*g-a)*x1*x3*x3"
KNECHT.13.1_AS_PAIRS="1,0,1,1,2,12,9"
KNECHT.13.1_AS_VECTOR="1,1,0,0,0,0,0,0,12,0,0,0,0,0"
KNECHT.13.2_AS_PAIRS="1,0,1,1,2,8,9,8,10,8,11"
KNECHT.13.2_AS_VECTOR="1,1,0,0,0,0,0,8,0,8,0,0,0"

GOLAY.23_COLUMN_RANKS_PROJECTIVELY="0,1,2,3,4,5,6,7,8,9,10,11,132,913,1460,1750,1898,2518,2787,2874,3320,3357,3662"

# [23,12,8]
# 0,1,2,3,4,5,6,7,8,9,10,11,132,913,1460,1750,1898,2518,2787,2874,3320,3357,3662

CODE.RM.3.1_GENMA="\n11111111\n01010101\n00110011\n00011111"

CODE.RM.4.1_GENMA="\n1111111111111111\n0101010101010101\n0011001100110011\n0001111100001111\n0000000011111111"

CODE.RS.8="\n5610000\n0561000\n0056100\n0005610\n0000561"

CODE.RS.11_RREF="\n1,0,0,0,0,0,0,7,2,\n1,0,0,0,0,0,0,8,3,\n0,1,0,0,0,0,0,1,2,\n0,0,1,0,0,0,0,0,8,8,\n0,0,1,0,0,0,0,8,8,\n
ago=12
CODE_15_6_6_B="\n 11111111100000-\n 11110000010000-\n 111001100001000-\n 101011010000100-\n 101010110000010-\n 111111111000001"  

# ago=720:
CODE_15_6_6_C="\n 11111111100000-\n 11110000010000-\n 111001100001000-\n 101010101000010-\n 10101001000010-\n 100010111000001"  

# ago=96:
CODE_15_6_6_D="\n 11111111110000-\n 11110000010000-\n 111001100001000-\n 101010100100010-\n 10101001000010-\n 011001011000001"  

# ago=360
CODE_15_6_6_E="\n 11111111110000-\n 11110000010000-\n 111001100001000-\n 100111010000100-\n 01010110000010-\n 01011010100001"  

BCH_21_15_PROJ="0, 1, 19, 37, 113, 420, 1651, 6577, 26284, 105115, 420442, 1681753, 6727000, 26907991, 107631958, 27874647, 111498582, 43341143, 173364566, 156587350, 14."  

461
476
477
478

479
480

481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497

498

BCH 21 15 GENERATOR MATRIX="1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,
·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,
·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,
·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,
·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,
·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,
·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,
·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,
·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,
·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,
·0,·0,·0,·1,·0,·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,
·1,·0,·1,·1,·1,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·1,·0,·1,·0,·1,·1,·1"

BCH 21 6 GENERATOR MATRIX="·1,·0,·0,·0,·0,·0,·1,·1,·0,·1,·0,·0,·1,·1,·0,·0,·1,·0,
·0,·1,·0,·0,·1,·0,·0,·0,·0,·0,·1,·1,·0,·1,·0,·0,·1,·1,·0,·0,·1,·0,·0,·1,·0,·0,·1,
·0,·0,·0,·1,·1,·1,·0,·0,·1,·1,·1,·1,·1,·1,·0,·1,·1,·0,·0,·0,·0,·1,·0,·0,·0,·1,·1,
·1,·0,·0,·1,·1,·1,·1,·1,·1,·0,·1,·1,·0,·0,·0,·0,·1,·0,·1,·1,·1,·0,·1,·0,·1,·0,·1,
·1,·0,·1,·1,·1,·1,·0,·0,·0,·0,·0,·1,·1,·0,·1,·0,·0,·1,·1,·0,·0,·1,·0,·0,·1,·0,·1·
"
POLY Q256 DEG30 SPARSE="1,0,26,1,210,2,24,3,\
138,4,148,5,160,6,58,7,108,8,199,9,95,10,56,\
11,9,12,205,13,194,14,193,15,3,16,248,17,110,\
18,150,19,24,20,169,21,192,22,212,23,112,24,\
144,25,97,26,109,27,174,28,253,29,1,30"
POLY Q256 DEG30 DENSE="1,26,210,24,138,148,\
160,58,108,199,95,56,9,205,194,193,3,248,110,\
150,24,169,192,212,112,144,97,109,174,253,1"

#·created·in·the·combinatorics·section:
ELEMENTARY SYMMETRIC 8 1="x0·+·x1·+·x2·+·x3·+·x4·+·x5·+·x6·+·x7"
ELEMENTARY SYMMETRIC 8 2="x0*x1·+·x0*x2·+·x0*x3·+·x0*x4·+·x0*x5·+·x0*x6·+·x0*x7·+
·x1*x2·+·x1*x3·+·x1*x4·+·x1*x5·+·x1*x6·+·x1*x7·+·x2*x3·+·x2*x4·+·x2*x5·+·x2*x6·+·
x2*x7·+·x3*x4·+·x3*x5·+·x3*x6·+·x3*x7·+·x4*x5·+·x4*x6·+·x4*x7·+·x5*x6·+·x5*x7·+·x
6*x7"
ELEMENTARY SYMMETRIC 8 3="x0*x1*x2·+·x0*x1*x3·+·x0*x1*x4·+·x0*x1*x5·+·x0*x1*x6·+·
x0*x1*x7·+·x0*x2*x3·+·x0*x2*x4·+·x0*x2*x5·+·x0*x2*x6·+·x0*x2*x7·+·x0*x3*x4·+·x0*x
3*x5·+·x0*x3*x6·+·x0*x3*x7·+·x0*x4*x5·+·x0*x4*x6·+·x0*x4*x7·+·x0*x5*x6·+·x0*x5*x7
·+·x0*x6*x7·+·x1*x2*x3·+·x1*x2*x4·+·x1*x2*x5·+·x1*x2*x6·+·x1*x2*x7·+·x1*x3*x4·+·x
1*x3*x5·+·x1*x3*x6·+·x1*x3*x7·+·x1*x4*x5·+·x1*x4*x6·+·x1*x4*x7·+·x1*x5*x6·+·x1*x5

462


H6="1,24"
H7="1,24,-1,4"
H8="1,21,-1,22"
H9="1,21,-1,23"
H10="1,22,-1,24"
H11="1,23,-1,24"
H12="1,61,-1,62"
H13="1,61,-1,21,-1,24,-1,62,1,22,1,23"
H14="-1,67,-1,68,1,61,1,66,1,69,-1,62"

Orb0="0"
Orb1="4"
Orb2="16"
Orb3="20"
Orb4="21"
Orb5="24"
Orb6="31"
Orb7="34"
Orb8="35"
Orb9="38"
Orb10="43"
Orb11="46"
Orb12="68"
Orb13="69"
Orb14="76"
Orb15="77"
Orb16="139"
Orb17="140"
Orb18="156"
Orb19="192"
Orb20="264"
Orb21="265"
Orb22="331"
Orb23="337"
Orb24="346"
Orb25="355"
Orb26="362"
Orb27="487"
Orb28="10566"
Orb29="65540"
Orb30="65542"
Orb31="65546"
Orb32="65547"
Orb33="65548"
Orb34="65550"
604 Orb35="65554"
605 Orb36="65561"
606 Orb37="65562"
607 Orb38="65569"
608 Orb39="65570"
609 Orb40="65603"
610 Orb41="65605"
611 Orb42="65609"
612 Orb43="65611"
613 Orb44="65612"
614 Orb45="65613"
615 Orb46="65614"
616 Orb47="65617"
617 Orb48="65618"
618 Orb49="65633"
619 Orb50="65634"
620 Orb51="65665"
621 Orb52="65666"
622 Orb53="65687"
623 Orb54="65695"
624 Orb55="65735"
625 Orb56="65743"
626 Orb57="65744"
627 Orb58="65759"
628 Orb59="65760"
629 Orb60="65831"
630 Orb61="65839"
631 Orb62="65843"
632 Orb63="65850"
633 Orb64="65851"
634 Orb65="65858"
635 Orb66="65859"
636 Orb67="65863"
637 Orb68="65865"
638 Orb69="65867"
639 Orb70="65868"
640 Orb71="65869"
641 Orb72="65871"
642 Orb73="65872"
643 Orb74="65873"
644 Orb75="65874"
645 Orb76="65887"
646 Orb77="65899"
647 Orb78="65900"
648 Orb79="65903"
649 Orb80="65904"
650 Orb81="65915"
651 Orb82="65919"
652 Orb83="65921"
653 Orb84="65922"
654 Orb85="66755"
655 Orb86="66763"
656 Orb87="66764"
657 Orb88="67107"
658 Orb89="67115"
659 Orb90="67117"
660 Orb91="67150"
661 Orb92="67243"
662 Orb93="73731"
663 Orb94="73733"
664 Orb95="73737"
665 Orb96="73753"
666 Orb97="73795"
667 Orb98="73796"
668 Orb99="73797"
669 Orb100="73798"
670 Orb101="73801"
671 Orb102="73802"
672 Orb103="73987"
673 Orb104="73993"
674 Orb105="74007"
675 Orb106="74051"
676 Orb107="74052"
677 Orb108="74053"
678 Orb109="74055"
679 Orb110="74057"
680 Orb111="74099"
681 Orb112="74105"
682 Orb113="74106"
683 Orb114="74243"
684 Orb115="74247"
685 Orb116="74248"
686 Orb117="74263"
687 Orb118="74264"
688 Orb119="74275"
689 Orb120="74276"
690 Orb121="74279"
691 Orb122="74280"
692 Orb123="74295"
693 Orb124="74296"
694 Orb125="74499"
695 Orb126="74500"
696 Orb127="74531"
697 Orb128="74532"
# elements of order 2:
# conjugacy class reps:
# elt order, class size, centralizer order:

#2A: 2\cdot 48960\cdot 40320 Baer involution
#2B: 2\cdot 5355\cdot 368640 \cdot one block of 10,11
#2C: 2\cdot 64260\cdot 30720 \cdot two blocks of 10,11 (problem group)
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787

CLASS 2B=-centralizer of element·"1,0,0,0,·1,1,0,0,·0,0,1,0,·0,0,0,1,·0"·-label·"
2B"
CLASS 2C=-centralizer of element·"1,0,0,0,·1,1,0,0,·0,0,1,0,·0,0,1,1,·0"·-label·"
2C"
#·problem·group

#·3·classes·of·elements·of·order·3
#·4·classes·of·elements·of·order·4

#·Baer·involution:
PGGL 4 4 SUBGROUP 2A=-PGGL·4·4·\
. -subgroup by generators·"2A"·2·1·"1,0,0,0,·0,1,0,0,·0,0,1,0,·0,0,0,1,·1"
PGGL 4 4 SUBGROUP 2A NORMALIZER=-PGGL·4·4·\
. -subgroup by generators·"centralizer 2A"·"40320"·10·\
. "1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1,0,·\
1,0,0,0,0,1,0,0,1,1,0,1,1,1,1,0,0,·\
1,0,0,0,0,1,0,0,1,0,1,0,0,1,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,1,1,0,1,0,0,·\
1,0,0,0,1,1,1,1,1,0,1,0,1,1,1,0,0,·\
1,0,0,0,0,1,1,0,0,0,1,1,1,0,1,0,0,·\
0,1,0,0,0,1,0,1,1,1,0,1,0,1,1,1,1

#·the·problem·group,·two·blocks·of·10,11:
PGGL 4 4 SUBGROUP 2C=-PGGL·4·4·\
. -subgroup by generators·"2C"·2·1·\
. "1,0,0,0,·1,1,0,0,·0,0,1,0,·0,0,1,1,·0"
PGGL 4 4 SUBGROUP 2C NORMALIZER=-PGGL·4·4·\
. -subgroup by generators·"centralizer 2C"·"30720"·9·\
. "1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,2,0,0,0,0,2,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,2,0,3,1,0,·\
1,0,0,0,0,1,0,0,0,0,1,0,1,0,3,1,0,·\
1,0,0,0,0,1,0,0,1,0,1,0,1,1,1,1,1,·\
1,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0,·\
1,0,0,0,2,1,0,0,0,0,1,0,1,0,0,1,0,·\
1,0,0,0,1,1,2,0,0,0,1,0,0,0,0,1,0,·\
1,0,3,0,1,1,1,3,0,0,2,0,0,0,0,2,1,"

469


790  PGGL_4_4_SUBGROUP_5A=--PGGL_4.4:
791  \\
792  793  PGGL_4_4_SUBGROUP_5A_NORMALIZER=--PGGL_4.4:
794  \\
795  796  \\
797  \\
798  \\
799  \\
800  \\
801  \\
802  PGGL_4_4_SUBGROUP_5B=--PGGL_4.4:
803  \\
804  \\
805  \\
806  \\
807  \\
808  \\
809  \\
810  \\
811  \\
812  \\
813  \\
814  \\
815  PGGL_4_4_SUBGROUP_2Cx2_0=--PGGL_4.4:
816  \\
817  \\
818  \\
819  \\
820  \\
821  \\
822  \\
823  \\
824  \\
825  \\
826  \\
827  \\
828  \\
829  \\
830  \\
831  \\
832  \\
833  \\
834
elementary-abelian-subgroups-of-order-4-with-3-elements-of-class-2C:

nice-generators, from Michael Epstein:
# subgroup-of-order-31-for-the-construction-of-regular-packings-in-PG_3_5:

PGL_4_5_SUBGROUP_31=-PGL_4_5::
- subgroup_by_generators:"31".31-1::
  "2,0,0,0,0,0,0,0,1,0,0,0,1,0,3,0,4"::
PGL_4_5_SUBGROUP_31_NORMALIZER=-PGL_4_5::
- subgroup_by_generators:"normalizer_31"."372".4-1::
  "1,0,0,0,0,4,0,0,0,4,0,0,0,0,4,"::
  "1,0,0,0,0,3,0,0,0,3,0,0,0,0,3,"::
  "1,0,0,0,0,4,0,0,0,2,1,0,3,2,4,"::
  "1,0,0,0,0,0,1,0,0,0,1,0,1,1,3,"::

#372:
"1,0,0,0,0,4,0,0,0,4,0,0,0,0,4,"::
"1,0,0,0,0,3,0,0,0,3,0,0,0,0,3,"::
"1,0,0,0,0,4,0,0,0,2,1,0,3,2,4,"::
"1,0,0,0,0,0,1,0,0,0,1,0,1,1,3,"::

#Exterior-square-roots:

#elt-of-order-3:
the-exterior-square-root-of-f-is-X= #[1:0:0:0]
[0:1:0:0]
[0:0:2:2]
[0:0:4:2]

#elt-of-order-31:
the-exterior-square-root-of-g-is-Z= #[1:0:0:0]
[0:3:4:3]
[0:3:3:4]
[0:3:2:3]

#Michael
HAMMING_CODE_CODEWORDS="0,·67,·37,·102,·22,·85,·51,·112,·15,·76,·42,·105,·25,·90,·60,·127"

SIMPLEX_CODE_GENERATOR="\n1,0,1,0,1,0,1,\n0,1,1,0,0,1,1,\n0,0,0,1,1,1,1"

HAMMING_CODE_GENERATOR="\n1,0,0,0,0,1,1,\n0,1,0,0,1,0,1,\n0,0,1,0,1,1,0,\n0,0,1,1,1,1"

HAMMING_CODE_ROWS_IN_BINARY_RANKS="67,37,22,15"

SIMPLEX_CODE_GENMA_CYCLIC="\n1,0,0,1,1,0,\n0,1,0,0,1,1,\n0,0,1,1,1,0,\n0,0,0,1,1,1"

CODE_GV_N15_K6="\n111111111100000\n11110000010000\n11001100001000\n11010101000001\n10101011000001\n10110100100001"

CODE_GV_N15_K6_CHECK="\n10000000011111\n01000000011110\n00100000011101\n00010000011010\n00001000010101\n00000100010110\n00000010010101\n00000001010110\n00000000110001"
REED_MULLER_3_1_CODEWORDS="0,255,170,85,204,51,102,\
153,240,15,90,165,60,195,150,105"

REED_MULLER_3_1_BASIS_IN_BINARY="255,170,204,240"

REED_MULLER_4_1_COLUMNS_OF_PARTITY_CHECK="1,3,5,7,9,11,13,\
15,17,19,21,23,25,27,29,31"

#-nearest codeword:"8,16,32,24,40,48,56,1,2,4,3,5,6,7,9,18,36,27,45,54,63,33,42,2
0,11,53,62,31"

RM_6_GENERATOR_1="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,\
22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,\
46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63"

RM_6_GENERATOR_2="1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,\
41,43,45,47,49,51,53,55,57,59,61,63"

RM_6_GENERATOR_3="2,3,6,7,18,19,22,23,10,11,14,15,26,27,30,31,34,35,38,\
39,42,43,46,47,50,51,54,55,58,59,62,63"

RM_6_GENERATOR_4="4,6,12,14,36,38,52,54,5,7,13,15,37,39,53,55,20,22,28,\
30,44,46,60,62,21,23,29,31,45,47,61,63"

RM_6_GENERATOR_5="8,9,12,13,24,25,28,29,10,11,14,15,26,27,30,31,40,41,\
44,45,56,57,60,61,42,43,46,47,58,59,62,63"

RM_6_GENERATOR_6="16,18,24,26,48,50,56,58,17,19,25,27,49,51,57,59,20,22,\
28,30,52,54,60,62,21,23,29,31,53,55,61,63"

RM_6_GENERATOR_7="32,34,48,50,33,35,49,51,42,43,46,47,52,54,40,42,\
56,58,41,43,57,59,44,46,60,62,45,47,61,63"

AG_2_3_BLOCKS="0,13,22,27,35,41,47,53,55,59,71,76"

LARGE_SET_AG_2_3_NEIGHBOR_SET="129,130,133,134,136,139,141,142,153,154,156,160,16
5,166,178,179,183,184,185,190,192,194,197,203,204,206,218,221,222,225,227,231,248
,315,319,320,323,325,341,342,343,344,345,349,368,371,375,378,381,383,392,393,397,4
02,403,405,416,419,421,422,425,426,429,430,440,443,447,449,453,454,464,467,468,47
3,474,479,490,493,494,497,500,503,513,517,518,520,523,527,536,539,541,542,544,547
,548,551,563,566,567,571,572,573,585,589,590,593,595,596,600,601,603,611,614,615,
625,629,631,635,637,638,657,659,661,667,668,671,681,683,686,689,691,693,705,706,7
09,710,712,715,717,718,720,723,724,729,733,735,747,748,750,752,754,757,777,780,78
1,784,790,791,802,804,807,808,811,814,824,827,828,831,832,835,837,838"

TEST_SYSTEM="\n0,1,0,1,0,0,\"
0,0,1,0,1,0,
1,0,1,0,0,0,
0,1,0,1,0,1,
1,0,0,0,0,1,
1,0,1,0,0,0,
0,1,0,1,1
TEST
RHS="1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"

PP4=-d1·-q1·-d2·-q2·-K·-search_control·-W·-end·-problem_label·PP4
PP4_GROUP1=-subgroup."1,1,1,1,."21·-group_label."cyclic21"
PP4_MASK1=\n
PP4=-d1·-q1·-d2·-q2·-K·-search_control·-W·-end·-problem_label·PP4

DELANDTSHEER_DOYEN_PROBLEM_COLBOURN_COLBOURN_7_13=-d1·-q1·-d2·-q2·-K·-search_control·-W·-end·-problem_label·DD_CC_7_13
DELANDTSHEER_DOYEN_PROBLEM_COLBOURN_COLBOURN_7_13_GROUP1=-subgroup."1,1,1,1,."91·-group_label."cyclic91"

DELANDTSHEER_DOYEN_PROBLEM_COLBOURN_COLBOURN_7_13_MASK1=\n
DELANDTSHEER_DOYEN_PROBLEM_27_53=-d1·-q1·-d2·-q2·-K·-search_control·-W·-end·
DELANDTSHEER_DOYEN_PROBLEM_27_53_GROUP1=-subgroup."1,1,1,0,.1,3,1,0,.1,9,1,0,.1,0,.1,1,.1,2,0,.4,.4,.1".18603·-group_label."group1"

#mask1:
#XX.
#X.X+

DELANDTSHEER_DOYEN_PROBLEM_27_53_MASK1=\n
DELANDTSHEER_DOYEN_PROBLEM_3_7=-d1·-q1·-d2·-q2·-K·-DDx·-DDy·-search_control·-W·-end·

475
Consider the binary code with generator matrix:

\[
\begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

Triangle graph:

\[
(0,1,1,0,1,0,1,1,0)
\]

PG.3.5 Packing 0 with AG03 = 444, 5001, 12957, 18194, 23485, 26817, 34667, 38299, 41249, 47472, 50450, 56601, 62638, 68986, 71833, 75369, 80805, 87025, 92577, 95676, 104509, 109718, 114948, 116333, 124391, 127498, 133240, 137711, 144777, 148059, 150175
# q=17:
# 3.is.p.e.mod.17.
# so.we.pick f=3.
# then,.2f^2=18=1
# 4f.=12
# X^4.-Y^4.-Z^4.+2f^2-Y^2Z^2.+4fx^2yz.

edge_curve_q17_equation="1,16,16,0,0,0,0,0,0,0,0,1,12,0,0"
edge_curve_q17_as_points="4,7,16,19,20,23,32,35,-89,-100,244,251"

edge_curve_q17_equation="1,16,16,0,0,0,0,0,0,0,0,1,12,0,0"
edge_curve_q17_as_points="4,7,16,19,20,23,32,35,-89,-100,244,251"

\n0,0,"$(edge_curve_q17_equation)"\n"
$(edge_curve_q17_as_points)\",\",-1\n\nend"
# Section 2.2: Orbiter Session

```bash
test: $(ORBITER_PATH)orbiter.out -v -define S-set -here "2,3,5,7,11,13" -end
```

# Section 2.3: Makefiles and Shell Scripts

# Section 2.4: Objects and Activities

```bash
test: $(ORBITER_PATH)orbiter.out -v 2
```

```bash
test: $(ORBITER_PATH)orbiter.out -v 2
```

```bash
test: $(ORBITER_PATH)orbiter.out -v 2
```
object_F_2:
 $(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_3\_\text{define}\_F\_\text{finite}\_\text{field}\_q\_2\_\text{end} 

object_PG_3_2:
 $(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_2\$
 $\text{define}\_F\_\text{finite}\_\text{field}\_q\_2\_\text{end}$
 $\text{define}\_P\_\text{projective}\_\text{space}\_3\_F\_\text{end}$

text ex:

(vector.ex):

create_BLT_5_1:
 $(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_2\$
 $\text{define}\_F\_\text{finite}\_\text{field}\_q\_5\_\text{end}$
 $\text{define}\_O\_\text{orthogonal}\_\text{space}\_0\_5\_F\_\text{end}$
 $\text{with}\_O\_\text{do}\_\text{orthogonal}\_\text{space}\_\text{activity}$
 $\text{create}\_\text{BLT}\_\text{set}\_\text{catalogue}\_1\_\text{end}$

create_surface_4_0:
 $(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_3$
 $\text{define}\_F\_\text{finite}\_\text{field}\_q\_4\_\text{end}$
 $\text{define}\_P\_\text{projective}\_\text{space}\_3\_F\_\text{end}$
 $\text{with}\_P\_\text{do}$
 $\text{projective}\_\text{space}\_\text{activity}$
 $\text{define}\_\text{surface}\_S4\_0\_q\_4\_\text{catalogue}\_0\_\text{end}$

end
-with S4_0-do
- cubic_surface_activity
-report
-end

### Section 2.6: Set-Builder

**SECTION_SET_BUILDER:**

set_of_primes:

$(ORBITER_PATH) orbiter.out -v 2

-do

-define S-set -here "2, 3, 5, 7, 11, 13" -end

-print_symbols

-set_interval:

$(ORBITER_PATH) orbiter.out -v 2 -define S-set -loop 0-64 1 -end

-print_symbols

-set builder_examples:

$(ORBITER_PATH) orbiter.out -v 2 -long_code 64 7

-set_builder-loop 0-64 1 -end

-set_builder-loop 0-32 1 -affine_function 2-1 -end

-set_builder-loop 0-16 1

-affine_function 4-2 -clone_with_affine_function 4-3 -end

-set_builder-set_builder-set_builder

-loop 0-4 1 -affine_function 1-4

-clone_with_affine_function 1-12 -end

-clone_with_affine_function 1-16

-end-clone_with_affine_function 1-32 -end

-set_builder-set_builder-loop 0-8 1 -affine_function 1-8

-clone_with_affine_function 1-24 -end

-clone_with_affine_function 1-32 -end

-set_builder-loop 0-16 1 -affine_function 1-16

-clone_with_affine_function 1-48 -end

-set_builder-loop 0-32 1 -affine_function 1-32 -end

-set_builder-loop 0-32 1 -affine_function 1-32 -end

-set_builder-loop 0-32 1 -affine_function 1-32 -end

-set_builder-loop 0-32 1 -affine_function 1-32 -end

-section: 2.7: Vector-Builder

**SECTION_VECTOR_BUILDER:**
vector_example1:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q5\-end\)
(define\-v\-vector\-field\-F\-dense:"0,1,2,3,4"\-end\)
-print\_symbols
\end{verbatim}

vector_example2:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q5\-end\)
(define\-v\-vector\-field\-F\-format\-2\-dense:"0,1,2,3,4,0"\-end\)
-print\_symbols
\end{verbatim}

vector_example_sparse:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q5\-end\)
(define\-v\-vector\-field\-F\-format\-4\-sparse:20:"1,0,1,19"\-end\)
-print\_symbols
\end{verbatim}

vector_example_repeat:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q5\-end\)
(define\-v\-vector\-repeat:"0,1,2,3"\-11\-end\)
-print\_symbols
\end{verbatim}

vector_example_all_one_11:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q5\-end\)
(define\-v\-vector\-repeat:1\-11\-end\)
-print\_symbols
\end{verbatim}

matrix_example1:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q2\-end\)
(define\-v\-vector\-field\-F\-format\-4\)
-dense$(HAMMING\_CODE\_GENERATOR)\-end\)
-print\_symbols
\end{verbatim}

matrix_example_co_1:

$(ORBITER\_PATH)\text{orbiter.out}\cdot-v.2:\$

\begin{verbatim}
(define\-F\-finite\_field\-q2\-end\)
(define\-v\-vector\-field\-F\-format\-22\)
\end{verbatim}

481
Section 2.8: Input Streams

SECTION_INPUT_STREAMS:

Section 3: Basic Algebra

SECTION_BASIC_NUMBER_THEORY:

order_of_2_mod_n:

$\text{order of q mod n:}$

$\text{order of q mod n: q^2}$

PR7:

PR11:

PR13:

PR17:

PR19:
PR23:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-smallest\_primitive\_root}\!\cdot\!23$

PR29:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-smallest\_primitive\_root}\!\cdot\!29$

PR31:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-smallest\_primitive\_root}\!\cdot\!31$

PR37:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-smallest\_primitive\_root}\!\cdot\!37$

PR\_100:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-smallest\_primitive\_root\_interval}\!\cdot\!2\!\cdot\!100$

Eulerfunction\_150:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-eulerfunction\_interval}\!\cdot\!1\!\cdot\!150$

PR\_1000:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-smallest\_primitive\_root\_interval}\!\cdot\!2\!\cdot\!1000$

Eulerfunction\_10000:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!1\!\cdot\text{-eulerfunction\_interval}\!\cdot\!10000\!\cdot\!100$

power\_function\_2\_mod\_11:
$(ORBITER\_PATH)orbiter.out-\!v\!\cdot\!5\!\cdot\text{-power\_function\_mod\_n}\!\cdot\!2\!\cdot\!11$

power\_function\_k2\_n11.csv

483
#randomized algo:

PR2:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -primitive_root 915839
```

```bash
# a-primitive-root-modulo-915839-is-43085
```

PM2:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -power_mod 43085 49842 915839
```

```bash
# the power of 43085 to the 49842 mod 915839 is 487320
```

DL2:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -discrete_log 487320 43085 915839
```

```bash
# The discrete log is 49842 since 487320 = 43085^49842 mod 915839, time: 0:22
```

IM_723:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -inverse_mod 723 4060
```

IM_3_19:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -inverse_mod 3 19
```

IM:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -inverse_mod 1865025205 2147483647
```

IM_gcd:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -extended_gcd 1865025205 2147483647
```

PM3a:

```bash
$ (ORBITER_PATH)orbiter.out -v 5 -power_mod 16807 1073741823 2147483647
```

sqrt_mod:

```bash
$ (ORBITER_PATH)orbiter.out -v 2 -square_root_mod 33 41
```

sqrt_5_mod_11:

```bash
$ (ORBITER_PATH)orbiter.out -v 2 -square_root_mod 5 11
```
\sqrt{5 \mod 19}:

\texttt{\textbackslash \texttt{draw_mod_8:}}
\texttt{$\texttt{\$(ORBITER\_PATH)orbiter.out-v.2-draw_options-embedded-end:\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_mod_n-n-8-file-mod_8-end}}$
\texttt{\textbackslash \texttt{pdflatex-mod_8_draw.tex}}
\texttt{\textbackslash \texttt{open-mod_8_draw.pdf}}

\texttt{\textbackslash \texttt{draw_mod_13:}}
\texttt{$\texttt{\$(ORBITER\_PATH)orbiter.out-v.2-\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_options-embedded-end:\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_mod_n-n-13-file-mod_13-power_cycle-2-end}}$
\texttt{\textbackslash \texttt{pdflatex-mod_13_draw.tex}}
\texttt{\textbackslash \texttt{open-mod_13_draw.pdf}}

\texttt{\textbackslash \texttt{draw_mod_3:}}
\texttt{$\texttt{\$(ORBITER\_PATH)orbiter.out-v.2-\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_options-embedded-nodes_empty-end:\}}
\texttt{\textbackslash \texttt{pdflatex-mod_3_draw.tex}}
\texttt{\textbackslash \texttt{open-mod_3_draw.pdf}}

\texttt{\textbackslash \texttt{draw_mod_3_c:}}
\texttt{$\texttt{\$(ORBITER\_PATH)orbiter.out-v.2-\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_options-embedded-nodes_empty-end:\}}
\texttt{\textbackslash \texttt{pdflatex-mod_3_c_draw.tex}}
\texttt{\textbackslash \texttt{open-mod_3_c_draw.pdf}}

\texttt{\textbackslash \texttt{draw_mod_4:}}
\texttt{$\texttt{\$(ORBITER\_PATH)orbiter.out-v.2-\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_options-embedded-nodes_empty-end:\}}
\texttt{\textbackslash \texttt{pdflatex-mod_4_draw.tex}}
\texttt{\textbackslash \texttt{open-mod_4_draw.pdf}}

\texttt{\textbackslash \texttt{draw_mod_6:}}
\texttt{$\texttt{\$(ORBITER\_PATH)orbiter.out-v.2-\}}
\texttt{$\texttt{\textbackslash \texttt{-draw_options-embedded-nodes_empty-end:\}}
\texttt{\textbackslash \texttt{pdflatex-mod_6_draw.tex}}
draw_mod_7:

```latex
\$\text{\$(ORBITER\_PATH)orbiter.out-\text{-v.2}\$
\$\text{-draw\_options:\text{-embedded\text{-nodes\_empty\text{-end}}\text{-draw\_mod\_n\text{-n.7\text{-file\_mod\_7\text{-end}}}}\text{-pdflatex\_mod\_7\_draw\_tex}}\text{-open\_mod\_7\_draw\_pdf}}$
```

draw_mod_15:

```latex
\$\text{\$(ORBITER\_PATH)orbiter.out-\text{-v.2}\$
\$\text{-draw\_options:\text{-embedded\text{-end}}\text{-draw\_mod\_n\text{-n.15\text{-file\_mod\_15\text{-end}}}}\text{-pdflatex\_mod\_15\_draw\_tex}}\text{-open\_mod\_15\_draw\_pdf}}$
```

draw_mod_127:

```latex
\$\text{\$(ORBITER\_PATH)orbiter.out-\text{-v.2}\$
\$\text{-draw\_options:\text{-scale\_0.8\text{-embedded\text{-end}}\text{-draw\_mod\_n\text{-n.127\text{-file\_mod\_127\text{-power\_cycle\_3\text{-end}}}}\text{-pdflatex\_mod\_127\_draw\_tex}}\text{-open\_mod\_127\_draw\_pdf}}$
```

sqrt_mod_20_31:

```latex
\$\text{\$(ORBITER\_PATH)orbiter.out-\text{-v.2\text{-square\_root\_mod\_20\_31}}}$
```

SECTION_PRIME_FIELDS:

```
# Section 3.2: Prime Fields

#pdflatex
```
F_3:

```bash
$ (ORBITER_PATH) orbiter.out \-v.3 \n(define F \-finite_field \-q.3 \-end \n(with F \-do \-finite_field_activity \-cheat_sheet_GF \-end

#pdflatex GF_3.tex

open GF_3.pdf
```

F_5:

```bash
$ (ORBITER_PATH) orbiter.out \-v.3 \n(define F \-finite_field \-q.5 \-end \n(with F \-do \-finite_field_activity \-cheat_sheet_GF \-end

pdflatex GF_5.tex

open GF_5.pdf
```

F_7:

```bash
$ (ORBITER_PATH) orbiter.out \-v.3 \n(define F \-finite_field \-q.7 \-end \n(with F \-do \-finite_field_activity \-cheat_sheet_GF \-end

pdflatex GF_7.tex

open GF_7.pdf
```

F_13:

```bash
$ (ORBITER_PATH) orbiter.out \-v.3 \n(define F \-finite_field \-q.13 \-end \n(with F \-do \-finite_field_activity \-cheat_sheet_GF \-end

pdflatex GF_13.tex

open GF_13.pdf
```

F_17:

```bash
$ (ORBITER_PATH) orbiter.out \-v.3 \n(define F \-finite_field \-q.17 \-end \n(with F \-do \-finite_field_activity \-cheat_sheet_GF \-end

pdflatex GF_17.tex

open GF_17.pdf
```
1600 F_19:
1601 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1602 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}19\text{-end}\backslash$
1603 ▶ ▶ ▶ \text{-with}\text{-do-}\text{\_finite}\text{\_field}\text{-activity-}\text{cheat}\text{-sheet}\text{GF}\text{-end}$
1604 ▶ pdflatex\text{-}GF\text{-}19.tex
1605 ▶ open\text{-}GF\text{-}19.pdf
1606
1607 F_31:
1608 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1609 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}31\text{-end}\backslash$
1610 ▶ ▶ ▶ \text{-with}\text{-do-}\text{\_finite}\text{\_field}\text{-activity-}\text{cheat}\text{-sheet}\text{GF}\text{-end}$
1611
1612 F_127:
1613 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1614 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}127\text{-end}\backslash$
1615 ▶ ▶ ▶ \text{-with}\text{-do-}\text{\_finite}\text{\_field}\text{-activity-}\text{cheat}\text{-sheet}\text{GF}\text{-end}$
1616
1617 F_11\text{\_product\_of\_all\_nonzero\_elements:}
1618 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1619 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}11\text{-end}\backslash$
1620 ▶ ▶ ▶ \text{-define}\text{\_S\_vector\_field}\text{\_F}\text{-\_loop}\text{1}\text{1}\text{1}\text{1}\text{-end}\backslash$
1621 ▶ ▶ ▶ \text{-with}\text{-do-}\text{\_finite}\text{\_field}\text{-activity-}\text{\text{\_product}\_of}\text{\_S\_end}$
1622
1623 F_7\text{_vandermonde:}
1624 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1625 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}7\text{-end}\backslash$
1626 ▶ ▶ ▶ \text{-with}\text{-do-}\text{\_finite}\text{\_field}\text{-activity\_Vandermonde\_matrix}\backslash$
1627 ▶ ▶ ▶ ▶ ▶ \text{-end}$
1628
1629 F_101\text{\_wo:}
1630 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1631 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}101\text{-without}\text{\_tables}\text{-end}\backslash$
1632 ▶ ▶ ▶ \text{-with}\text{-do-}\text{\_finite}\text{\_field}\text{-activity-}\text{cheat}\text{-sheet}\text{GF}\text{-end}$
1633 ▶ pdflatex\text{-}GF\text{-}101.tex
1634 ▶ open\text{-}GF\text{-}101.pdf
1635
1636 F_1021\text{\_wo:}
1637 ▶ $(\text{ORBITER\_PATH})\text{or} \text{b} \text{e} \text{r} \text{t} \text{e} \text{r} \text{.} \text{o} \text{u} \text{t} \text{-} \text{v} \text{.} \text{3} \backslash \n$
1638 ▶ ▶ ▶ \text{-define}\text{-finite}\text{\_field}\text{-q}1021\text{-without}\text{\_tables}\text{-end}\backslash$
1639 ▶ ▶ ▶ ▶ ▶ ▶ \text{-end}$
### Section 3.3: Extension Fields

**SECTION_EXTENSION_FIELDS:**

- **F₄:**
  - $(\text{ORBITER\_PATH})orbiter.out -v 3$
  - `-define $F$ Finite\_Field -q 4 -end`
  - `-with $F$ -do Finite\_Field\_Activity - cheat\_sheet.GF -end`
  - `-draw\_matrix\_input\_csv\_file GF\_q4\_addition\_table.csv`
  - `-draw\_matrix\_input\_csv\_file GF\_q4\_multiplication\_table.csv`
  - `-box width 40 -bit depth 24 -partition 3\_4 4 -end`

- **F₄ tables:**
  - $(\text{ORBITER\_PATH})orbiter.out -v 3$
  - `-define $F$ Finite\_Field -q 4 -end`
  - `-with $F$ -do Finite\_Field\_Activity - cheat\_sheet.GF -end`
  - `-draw\_matrix\_input\_csv\_file GF\_q4\_addition\_table.csv`
  - `-draw\_matrix\_input\_csv\_file GF\_q4\_multiplication\_table.csv`
  - `-box width 40 -bit depth 24 -partition 3\_3 3 -end`
  - `#pdflatex GF\_4.tex`
  - `#open GF\_4.pdf`

- **trace 4:**
  - $(\text{ORBITER\_PATH})orbiter.out -v 3$
  - `-define $F$ Finite\_Field -q 4 -end`
  - `-with $F$ -do Finite\_Field\_Activity - trace -end`
  - `$\text{ORBITER\_PATH}$ orbiter.out -v 3 -reformat $F$ q4\_trace.csv $F$ q4\_trace\_2x2.csv 2`
  - `$\text{ORBITER\_PATH}$ orbiter.out -v 2 -draw\_matrix\_input\_csv\_file $F$ q4\_trace\_2x2.csv`
  - `-box width 40 -bit depth 24 -partition 4\_2 2 -end`
  - `$\text{ORBITER\_PATH}$ orbiter.out -v 3 -define $F$ Finite\_Field -q 4 -end`
  - `-with $F$ -do Finite\_Field\_Activity`
  - `-algebraic\_normal\_form $F$ q4\_trace.csv 2`
  - `-end`

- **trace 4.WH.transform:**
  - $(\text{ORBITER\_PATH})orbiter.out -v 3$
  - `-define $F$ Finite\_Field -q 4 -end`
  - `-with $F$ -do Finite\_Field\_Activity`
  - `-Walsh\_Hadamard\_transform $F$ q4\_trace.csv 2`
  - `-end`
\begin{verbatim}
F_8:
  \$\{\text{ORBITER\_PATH}\}\text{orbiter.out\_v.3}\$
  \$\text{-define}F\_\text{-finite\_field\_q8\_end}\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity\_cheat\_sheet}\_GF\_end$  
  pdflatex\_GF\_8\_tex
  open\_GF\_8\_pdf

# compute 2*3 and 3*3 in F_8:

F_8\_arithmetic:
  \$\{\text{ORBITER\_PATH}\}\text{orbiter.out\_v.3}\$
  \$\text{-define}F\_\text{-finite\_field\_q8\_end}\$
  \$\text{-define\_e1\_formula}\_\"test\"\_\"test\"\_\"a*b\"\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity}\$
  \$\text{-evaluate\_e1}\_\"a=2,b=3\"\_\text{-end}\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity}\$
  \$\text{-evaluate\_e1}\_\"a=3,b=3\"\_\text{-end}\$

F_16_7\_power_5:
  \$\{\text{ORBITER\_PATH}\}\text{orbiter.out\_v.3}\$
  \$\text{-define}F\_\text{-finite\_field\_q16\_end}\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity}\$
  \$\text{-parse\_and\_evaluate}\_\"test\"\_\"(a*a)*(a*a)*a\"\_\"a=7\"\_\text{-end}\$
  \$\text{dot\_Tpng\_test.gv\_test.png}\$
  \# the answer is 11.

F_8\_near\_bent_5:
  \$\{\text{ORBITER\_PATH}\}\text{orbiter.out\_v.3}\$
  \$\text{-define}F\_\text{-finite\_field\_q8\_end}\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity}\$
  \$\text{-identity\_function}\_F8\_csv\$
  \$\text{-end}\$
  \$\{\text{ORBITER\_PATH}\}\text{orbiter.out\_v.3}\$
  \$\text{-define}F\_\text{-finite\_field\_q8\_end}\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity}\$
  \$\text{-apply\_power\_function}\_F8\_csv\_5\$
  \$\text{-end}\$
  \$\{\text{ORBITER\_PATH}\}\text{orbiter.out\_v.3}\$
  \$\text{-define}F\_\text{-finite\_field\_q8\_end}\$
  \$\text{-with}F\_\text{-do}\_\text{-finite\_field\_activity}\$
\end{verbatim}
}
1740 ▶ ▶ ▶ -apply_trace_function F8_power_5.csv
1741 ▶ ▶ -end
1742 ▶ $(ORBITER_PATH)orbiter.out-v.3-
1743 ▶ ▶ -define F-finite_field-q.2-end-
1744 ▶ ▶ -with F-do-finite_field_activity-
1745 ▶ ▶ -Walsh_Hadamard_transform-
1746 ▶ ▶ ▶ F8_power_5_trace.csv-4-
1747 ▶ ▶ -end
1748 ▶ $(ORBITER_PATH)orbiter.out-v.3-reformat-
1749 ▶ ▶ F8_power_5_trace.csv F8_power_5_trace_8x1.csv-1
1750 ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
1751 ▶ ▶ -input_csv_file F8_power_5_trace_8x1.csv-
1752 ▶ ▶ ▶ -box_width 40-bit_depth 24-
1753 ▶ ▶ ▶ -partition 4 8 1-
1754 ▶ ▶ -end
1755 ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
1756 ▶ ▶ -input_csv_file Walsh_01_3.csv-
1757 ▶ ▶ -box_width 40-bit_depth 24-partition 4 8 8-end
1758 ▶ $(ORBITER_PATH)orbiter.out-v.3-
1759 ▶ ▶ -reformat F8_power_5_trace_transformed.csv-
1760 ▶ ▶ F8_power_5_trace_transformed_8x1.csv-1
1761 ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
1762 ▶ ▶ -input_csv_file F8_power_5_trace_transformed_8x1.csv-
1763 ▶ ▶ -box_width 40-bit_depth 24-partition 4 8 1-end
1764
1765 bent_4:
1766 ▶ $(ORBITER_PATH)orbiter.out-v.3-bent-4
1767
1768 bent_4a:
1769 ▶ $(ORBITER_PATH)orbiter.out-v.3-
1770 ▶ ▶ -reformat-bent_function_n4.csv-
1771 ▶ ▶ bent_function_n4_16x1.csv-1
1772 ▶ $(ORBITER_PATH)orbiter.out-v.3-
1773 ▶ ▶ -define F-finite_field-q.2-end-
1774 ▶ ▶ -with F-do-finite_field_activity-
1775 ▶ ▶ -Walsh_Hadamard_transform-bent_function_n4.csv-4-end
1776 ▶ $(ORBITER_PATH)orbiter.out-v.3-
1777 ▶ ▶ -reformat-bent_function_n4_transformed.csv-
1778 ▶ ▶ bent_function_n4_transformed_16x1.csv-1
1779 ▶ #$$(ORBITER_PATH)orbiter.out-v.2-draw_matrix-input_csv_file-bent_function_n4_transformed_16x1.csv-box_width 40-bit_depth 24-partition 4 16 1-end
1780
1781 #0:0/448:(0,0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,1,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,
1782 #X0*X3*X4^2+X1*X2*X4^2

491
F_9:
\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$
\$\text{-define\text{-}finite\_field\_q.9\text{-}end}\$
\$\text{-with\text{-}finite\_field\_activity\text{-}cheat\_sheet\_GF\text{-}end}\$
\$\text{(ORBITER\_PATH)orbiter.out-v.2\text{-}draw\_matrix}\$
\$\text{-input\_csv\_file\_GF\_q9\_addition\_table.csv}\$
\$\text{-box\_width\_40\_bit\_depth\_24\_partition\_3\_"3,6\"\_"3,6\"\_end}\$
\$\text{(ORBITER\_PATH)orbiter.out-v.2\text{-}draw\_matrix}\$
\$\text{-input\_csv\_file\_GF\_q9\_multiplication\_table.csv}\$
\$\text{-box\_width\_40\_bit\_depth\_24\_partition\_3\_"2,6\"\_"2,6\"\_end}\$
\$\text{(ORBITER\_PATH)orbiter.out-v.2\text{-}draw\_matrix}\$
\$\text{-input\_csv\_file\_GF\_q9\_addition\_table\_reordered.csv}\$
\$\text{-box\_width\_40\_bit\_depth\_24\_partition\_3\_9\_9\_end}\$
\$\text{(ORBITER\_PATH)orbiter.out-v.2\text{-}draw\_matrix}\$
\$\text{-input\_csv\_file\_GF\_q9\_multiplication\_table\_reordered.csv}\$
\$\text{-box\_width\_40\_bit\_depth\_24\_partition\_3\_8\_8\_end}\$
\$\text{pdflatex\_GF\_9.tex}\$
\$\text{open\_GF\_9.pdf}\$

trace_9:
\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$
\$\text{-define\text{-}finite\_field\_q.9\text{-}end}\$
\$\text{-with\text{-}finite\_field\_activity\text{-}trace\text{-}end}\$
\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$
\$\text{-reformat\_F\_q9\_trace.csv\_F\_q9\_trace\_3x3.csv\_3}\$
\$\text{(ORBITER\_PATH)orbiter.out-v.2\text{-}draw\_matrix}\$
\$\text{-input\_csv\_file\_F\_q9\_trace\_3x3.csv}\$
\$\text{-box\_width\_40\_bit\_depth\_24\_partition\_4\_3\_3\_end}\$

F_16:
\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$
\$\text{-define\text{-}finite\_field\_q.16\text{-}end}\$
\$\text{-with\text{-}finite\_field\_activity\text{-}cheat\_sheet\_GF\text{-}end}\$
\$\text{pdflatex\_GF\_16.tex}\$
F_16_tables:

- $(ORBITER_PATH)orbiter.out-v.3:\
  -define:F--finite_field--q.16--end:\
  -with:F--do--finite_field_activity--cheat_sheet_GF--end

- $(ORBITER_PATH)orbiter.out--v.2--draw_matrix:\
  -input_csv_file:GF_q16_addition_table.csv:\
  -box_width:40--bit_depth:24--partition:3:16:16--end

- $(ORBITER_PATH)orbiter.out--v.2--draw_matrix:\
  -input_csv_file:GF_q16_multiplication_table.csv:\
  -box_width:40--bit_depth:24--partition:3:16:16--end

- $(ORBITER_PATH)orbiter.out--v.2--draw_matrix:\
  -input_csv_file:GF_q16_addition_table_reordered.csv:\
  -box_width:40--bit_depth:24--partition:3:16:16--end

- $(ORBITER_PATH)orbiter.out--v.2--draw_matrix:\
  -input_csv_file:GF_q16_multiplication_table_reordered.csv:\
  -box_width:40--bit_depth:24--partition:3:15:15--end

trace_16:

- $(ORBITER_PATH)orbiter.out-v.3:\
  -define:F--finite_field--q.16--end:\
  -with:F--do--finite_field_activity--trace--end

- $(ORBITER_PATH)orbiter.out-v.3:\
  -reformat_F_q16_trace.csv:F_q16_trace_4x4.csv:4

- $(ORBITER_PATH)orbiter.out--v.2--draw_matrix:\
  -input_csv_file:F_q16_trace_4x4.csv:\
  -box_width:40--bit_depth:24--partition:4:4:4--end

F_16_bent_wrong:

- $(ORBITER_PATH)orbiter.out-v.3:\
  -define:F--finite_field--q.16--end:\
  -with:F--do--finite_field_activity:\

- $(ORBITER_PATH)orbiter.out-v.3:\
  -identity_function:F16.csv--end

- $(ORBITER_PATH)orbiter.out-v.3:\
  -define:F--finite_field--q.16--end:\
  -with:F--do--finite_field_activity:\

- $(ORBITER_PATH)orbiter.out-v.3:\
  -apply_power_function:F16.csv:9--end

- $(ORBITER_PATH)orbiter.out-v.3:\
  -define:F--finite_field--q.16--end:\
  -with:F--do--finite_field_activity:\

- $(ORBITER_PATH)orbiter.out-v.3:\
  -apply_trace_function:F16_power_9.csv--end

- $(ORBITER_PATH)orbiter.out-v.3:\
  -define:F--finite_field--q.2--end:\
  -with:F--do--finite_field_activity:\
-Walsh_Hadamard_transform-F16_power_9_trace.csv-4-end

$\text{(ORBITER PATH)}$ orbiter.out-v:3:\n
-reformat-F16_power_9_trace.csv-F16_power_9_trace_16x1.csv-1\n
$\text{(ORBITER PATH)}$ orbiter.out-v:2.-draw_matrix\n
-input_csv_file-F16_power_9_trace_16x1.csv\n
$\text{(ORBITER PATH)}$ orbiter.out-v:2.-draw_matrix\n
-input_csv_file-Walsh_01_4.csv\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:16:1.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end

-define-F-finite_field-q-16.-end\n
-loop:L-0:16:1:\n
-with:F-do-\n
-finite_field_activity-\n
-field_reduction-"F16_over_F4\_%L":4:1:1."%L":-end-\n
-end_loop\n
-loop:L-0:16:1:\n
-draw_matrix-input_csv_file-F16_over_F4\_%L.csv-\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end\n
-end_loop

#the_polynomial-31.is_notPrimitive:\n
F_16.poly31:\n
$\text{(ORBITER PATH)}$ orbiter.out-v:3:\n
-define-F-finite_field-q-16.-override_polynomial-31.-end\n
-with:F-do-finite_field_activity-\n
-cheat_sheet_GF-\n
F_16_over_F4_field_reduction:

$\text{(ORBITER PATH)}$ orbiter.out-v:2:\n
-define-F-finite_field-q-16.-end-\n
-loop:L-0:16:1-\n
-with:F-do-\n
-finite_field_activity-\n
-field_reduction."F16_over_F4\_%L":4:1:1."%L":-end-\n
-end_loop-\n
-loop:L-0:16:1-\n
-draw_matrix-input_csv_file-F16_over_F4\_%L.csv-\n
-box_width-40.-bit_depth-24.-partition-4:2:2.-end-\n
-end_loop-\n
-end_loop
1924
1925
1926  F_25:
1927  ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.3\$
1928  ▶ ▶ -define F\_finite\_field\_q\_25\_end\$
1929  ▶ ▶ -with F\_do\_finite\_field\_activity\$
1930  ▶ ▶ ▶ -cheat_sheet_GF\_end
1931  ▶ ▶ -input Csv_file\_GF\_q25\_addition\_table.csv\$
1932  ▶ ▶ -box_width\_40\_bit_depth\_24\_partition\_3\_25\_25\_end
1933  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.2\_draw\_matrix\$
1934  ▶ ▶ -input Csv_file\_GF\_q25\_multiplication\_table.csv\$
1935  ▶ ▶ -box_width\_40\_bit_depth\_24\_partition\_3\_24\_24\_end
1936  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.2\_draw\_matrix\$
1937  ▶ ▶ -box_width\_40\_bit_depth\_24\_partition\_3\_25\_25\_end
1938  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.2\_draw\_matrix\$
1939  ▶ ▶ -input Csv_file\_GF\_q25\_addition\_table\_reordered.csv\$
1940  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.2\_draw\_matrix\$
1941  ▶ ▶ -box_width\_40\_bit_depth\_24\_partition\_3\_24\_24\_end
1942  ▶ ▶ trace_25:
1943  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.3\$
1944  ▶ ▶ -define F\_finite\_field\_q\_25\_end\$
1945  ▶ ▶ -with F\_do\_finite\_field\_activity\_trace\_end
1946  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.3\$
1947  ▶ ▶ -reformat F\_q25\_trace.csv\_F\_q25\_trace\_5x5.csv\_5
1948  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.2\_draw\_matrix\$
1949  ▶ ▶ -input Csv_file\_GF\_q25\_addition\_table.csv\$
1950  ▶ ▶ -box_width\_40\_bit_depth\_24\_partition\_3\_25\_25\_end
1951  ▶ ▶ F_32:
1952  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.3\$
1953  ▶ ▶ -define F\_finite\_field\_q\_32\_end\$
1954  ▶ ▶ -with F\_do\_finite\_field\_activity\$
1955  ▶ ▶ -cheat_sheet_GF\_end
1956  ▶ ▶ pdf\_latex\_GF\_32\_tex
1957  ▶ ▶ open\_GF\_32\_pdf
1958  ▶ F_49:
1959  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.3\$
1960  ▶ ▶ -define F\_finite\_field\_q\_49\_end\$
1961  ▶ ▶ -with F\_do\_finite\_field\_activity\_cheat_sheet_GF\_end
1962  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter\_out}\cdot v.2\_draw\_matrix\$
1963
1964
1965
1966
1967
1968
1969
1970
trace_49:

```
trace_64_poly_2:

trace_81:

F_125:
2065 F_256:
2066 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3\:\$
2067 \> \> -\text{define-F\,-\text{-\,finite\_\,field\,-\,q}\,256\,-\text{-\,end}}\:
2068 \> \> -\text{with-F\,-\,do\,-\,finite\_\,field\,activity\,-\,\text{-\,cheat\_\,sheet\_\,GF\,-\,end}}\:
2069 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,-\text{-\,draw\_\,matrix}\,\$
2070 \> \> -\text{\text{-\,input\_\,csv\_\,file\,GF\,\,q}\,256\,\,addition\,table\,\,csv}\,\$
2071 \> \> -\text{\,box\,\,width\,40\,-\,bit\,\,depth\,24\,\,-\,partition\,3\,\,256\,\,256\,-\,end}\,$
2072 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,-\text{-\,draw\,\,matrix}\,$
2073 \> \> -\text{\,input\,\,csv\,\,file\,\,GF\,\,q}\,256\,\,multiplication\,table\,\,csv\,$
2074 \> \> -\text{\,box\,\,width\,40\,-\,bit\,\,depth\,24\,\,-\,partition\,3\,\,255\,\,255\,-\,end}\,$
2075 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,-\text{-\,draw\,\,matrix}\,$
2076 \> \> -\text{\,input\,\,csv\,\,file\,\,GF\,\,q}\,256\,\,addition\,table\,\,reordered\,csv\,$
2077 \> \> -\text{\,box\,\,width\,40\,-\,bit\,\,depth\,24\,\,-\,partition\,3\,\,256\,\,256\,-\,end}\,$
2078 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,-\text{-\,draw\,\,matrix}\,$
2079 \> \> -\text{\,input\,\,csv\,\,file\,\,GF\,\,q}\,256\,\,multiplication\,table\,\,reordered\,csv\,$
2080 \> \> -\text{\,box\,\,width\,40\,-\,bit\,\,depth\,24\,\,-\,partition\,3\,\,255\,\,255\,-\,end}\,$
2081 \> \> pdf\,\,latex-GF\,\,256\,\,tex\,$
2082 \> \> open-GF\,\,256\,\,pdf\,$
2083
2084
2085 trace_256:
2086 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3\,$
2087 \> \> -\text{define-F\,-\,finite\,field\,-\,q\,256\,-\,end}\,$
2088 \> \> -\text{with-F\,-\,do\,-\,finite\,field\,activity\,-\,trace\,-\,end}\,$
2089 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3\,$
2090 \> \> -\text{reformat-F\,\,q256\,\,trace\,\,csv\,\,F\,\,q256\,\,trace\,\,16x16\,\,csv\,\,16}\,$
2091 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,-\text{-\,draw\,\,matrix}\,$
2092 \> \> -\text{\,input\,\,csv\,\,file\,\,F\,\,q256\,\,trace\,\,16x16\,\,csv\,\,\,}\,$
2093 \> \> -\text{\,box\,\,width\,40\,-\,bit\,\,depth\,24}\,$
2094 \> \> -\text{\,partition\,\,4\,\,"1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"}\,$
2095 \> \> "1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"\,-\text{-\,end}\,$
2096
2097
2098
2099 F_289:
2100 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3\,$
2101 \> \> -\text{define-F\,-\,finite\,field\,-\,q\,289\,-\,end}\,$
2102 \> \> -\text{with-F\,-\,do\,-\,finite\,field\,activity\,-\,\text{-\,cheat\,\,sheet\,\,GF\,-\,end}}\,$
2103 \> pdf\,\,latex-GF\,\,289\,\,tex\,$
2104 \> \> open-GF\,\,289\,\,pdf\,$
2105
2106
2107 F_512:
2108 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3\,$
2109 \> \> -\text{define-F\,-\,finite\,field\,-\,q\,512\,-\,end}\,$
2110
2111

498
2112 # User-time: 2/100 seconds
2113
2114 F_1024:
2115  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2116  ▶ ▶ -define F: finite_field -q 1024 -end
2117 # User-time: 10/100 seconds
2118
2119 F_2048:
2120  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2121  ▶ ▶ -define F: finite_field -q 2048 -end
2122
2123 F_4096:
2124  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2125  ▶ ▶ -define F: finite_field -q 4096 -end
2126
2127 # User-time: 0:2
2128
2129 F_8192:
2130  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2131  ▶ ▶ -define F: finite_field -q 8192 -end
2132
2133 # User-time: 0:8
2134
2135 F_16384:
2136  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2137  ▶ ▶ -define F: finite_field -q 16384 -end
2138
2139 # User-time: 0:37
2140
2141 F_32768:
2142  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2143  ▶ ▶ -define F: finite_field -q 32768 -end
2144
2145
2146 F_65536:
2147  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2148  ▶ ▶ -define F: finite_field -q 65536 -end
2149
2150 F_131072:
2151  ▶ $(ORBITER_PATH)orbiter.out -v 3 \n2152  ▶ ▶ -define F: finite_field -q 131072 -end
2153
2159  F_262144:
2160  |> $(ORBITER_PATH)orbiter.out-v.3:\n2161  |  |> -define F=finite_field-q262144-end
2162
2163  F_524288:
2164  |> $(ORBITER_PATH)orbiter.out-v.3:\n2165  |  |> -define F=finite_field-q524288-end
2166
2167  F_1048576:
2168  |> $(ORBITER_PATH)orbiter.out-v.3:\n2169  |  |> -define F=finite_field-q1048576-end
2170
2171  F_2097152:
2172  |> $(ORBITER_PATH)orbiter.out-v.3:\n2173  |  |> -define F=finite_field-q2097152-end
2174
2175  F_4194304:
2176  |> $(ORBITER_PATH)orbiter.out-v.3:\n2177  |  |> -define F=finite_field-q4194304-end
2178
2179  F_8388608:
2180  |> $(ORBITER_PATH)orbiter.out-v.3:\n2181  |  |> -define F=finite_field-q8388608-end
2182
2183  #stuck in finite field::find primitive element
2184
2185
2186
2187  #############################################################################
2188  #Section 3.4: Linear Algebra over Finite Fields
2189
2190
2191  SECTION_LINEAR_ALGEBRA:
2192
2193
2194  RREF:
2195  |> $(ORBITER_PATH)orbiter.out-v.2:\n2196  |  |> -define F=finite_field-q2-end:\n2197  |  |  |> -define v=vector-field F=format 2\n2198  |  |  |  |> -dense="1,1,1,0,1,1,0,0,1":\n2199  |  |  |  |  |> -end:\n2200  |  |  |> -with F-do-finite_field_activity:\n2201  |  |  |> -RREF v=normalize_from_the_right:\n2202  |  |  |> -end
2203
2204
2205  RREF_V7:
nullspace:

```bash
nullspace:
```

eigenstuff:

```bash
eigenstuff:
```

classes_GL_3_2:

```bash
classes_GL_3_2:
```

classes_GL_4_2:

```bash
classes_GL_4_2:
```

---

501
2253 2254   #.252·classes
2255 2256 2257 2258  RREF.demo_4_4_q5:
2259      $(\text{ORBITER\ PATH})orbiter.out\ -v\ 2:\$
2260      $(\text{define\ -finite\ field\ -q\ 5\ -end})$
2261      $(\text{-with\ -do})$
2262      $(\text{-finite\ field\ activity\ -RREF.demo_4_4\ -end})$
2263      pdflatex\ RREF.example_q5_4_4.tex
2264      \#open\ RREF.example_q5_4_4.pdf
2265      gs\ -sDEVICE=png16\ -dFIXEDMEDIA\$
2266      $(\text{-dDEVICEWIDTHPOINTS=500\ -dDEVICEHEIGHTPOINTS=450})$
2267      $(\text{-r240\ -oRREF.example_q5_4_4.page%02d.png})$
2268      $(\text{RREF.example_q5_4_4.pdf})$
2269 2270  #\text{-dDEVICEWIDTHPOINTS=100\ -dDEVICEHEIGHTPOINTS=100}$
2271 2272 2273  RREF.demo_4_6_q7:
2274      $(\text{ORBITER\ PATH})orbiter.out\ -v\ 2:\$
2275      $(\text{define\ -finite\ field\ -q\ 7\ -end})$
2276      $(\text{-with\ -do})$
2277      $(\text{-finite\ field\ activity\ -RREF.random_matrix_4_6\ -end})$
2278      pdflatex\ RREF.example_q7_4_6.tex
2279      gs\ -sDEVICE=png16\ -dFIXEDMEDIA\$
2280      $(\text{-dDEVICEWIDTHPOINTS=500\ -dDEVICEHEIGHTPOINTS=450})$
2281      $(\text{-r240\ -oRREF.example_q7_4_6.page%02d.png})$
2282      $(\text{RREF.example_q7_4_6.pdf})$
2283      open\ RREF.example_q7_4_6.pdf
2284 2285 2286  RREF.demo_4_8_q8:
2287      $(\text{ORBITER\ PATH})orbiter.out\ -v\ 2:\$
2288      $(\text{define\ -finite\ field\ -q\ 8\ -end})$
2289      $(\text{-with\ -do})$
2290      $(\text{-finite\ field\ activity\ -RREF.random_matrix_4_8\ -end})$
2291      pdflatex\ RREF.example_q8_4_8.tex
2292      \#open\ RREF.example_q8_4_8.pdf
2293      gs\ -sDEVICE=png16\ -dFIXEDMEDIA\$
2294      $(\text{-dDEVICEWIDTHPOINTS=500\ -dDEVICEHEIGHTPOINTS=450})$
2295      $(\text{-r240\ -oRREF.example_q8_4_8.page%02d.png})$
2296      $(\text{RREF.example_q8_4_8.pdf})$
2297 2298 2299

502
SECTION ADVANCED TOPICS IN FINITE FIELDS:

normal basis 2,3:

F8 over F2 field reduction:
\$\newcommand{\field_reduction}{\text{field_reduction}}\$

\begin{verbatim}
2347 \> \> \> \> \> -\field_reduction\"F8_red\_%L\".2:1.1\"%L\".\\
2348 \> \> \> \> -\end\\
2349 \> \> \> -\end_loop\$

2350 \>$\text{(ORBITER PATH)}\text{orbiter.out}\ >\ -v.2\ >\ -loop\ L.0.8.1\$

2351 \> \> \> -\draw_matrix -\input_csv_file\F8_red\_%L.csv\$

2352 \> \> \> -box_width.40 -bit_depth.24 -partition.4.3.3 -end\$

2353 \> \> -\end_loop\$

2354 \> #pdflatex\field_reduction\_Q8_q2.5.7.tex

2355

2356

2357

2358

2359 \F_{64\over F8}\text{field_reduction}:\$

2360 \> $(\text{ORBITER PATH})\text{orbiter.out}\ >\ -v.2\$

2361 \> > \defineF\finite_field\ -q.64\ -end\$

2362 \> > \define_elts\vector\field\ -l.0.64.1\ -end\$

2363 \> > \withF\do\$

2364 \> > \finite_field_activity\ -\field_reduction\"F64\_over\_F8\".8.8.8\$

2365 \> > \> \elts\ -end\$

2366 \> $(\text{ORBITER PATH})\text{orbiter.out}\ >\ -v.2\ -\draw_matrix\$

2367 \> > \> \input_csv_file\F64\_over\_F8.csv\$

2368 \> > \> \box_width.40 -bit_depth.24\$

2369 \> > \> \partition.4\"2,2,2,2,2,2,2,2\".2,2,2,2,2,2,2,2\". -end\$

2370 \> open\F64\_over\_F8\_draw.bmp

2371 \> #pdflatex\field_reduction\_Q64_q8.8.8.tex

2372 \> #open\field_reduction\_Q64_q8.8.8.pdf

2373

2374

2375 \F_{64\over F4}\text{field_reduction}:\$

2376 \> $(\text{ORBITER PATH})\text{orbiter.out}\ >\ -v.2\$

2377 \> > \defineF\finite_field\ -q.64\ -end\$

2378 \> > \define_elts\vector\field\ -l.0.64.1\ -end\$

2379 \> > \withF\do\$

2380 \> > \finite_field_activity\$

2381 \> > > \field_reduction\"F64\_over\_F4\".4.8.8\elts\ -end\$

2382 \> $(\text{ORBITER PATH})\text{orbiter.out}\ >\ -v.2\ -\draw_matrix\$

2383 \> > \> \input_csv_file\F64\_over\_F4.csv\$

2384 \> > \> \box_width.40 -bit_depth.24\$

2385 \> > \> \partition.4\"3,3,3,3,3,3,3,3\".3,3,3,3,3,3,3,3\". -end\$

2386 \> open\F64\_over\_F4\_draw.bmp

2387 \> #pdflatex\field_reduction\_Q64_q4.8.8.tex

2388 \> #open\field_reduction\_Q64_q4.8.8.pdf

2389

2390

2391 \F_{64\over F2}\text{field_reduction}:\$

2392 \> $(\text{ORBITER PATH})\text{orbiter.out}\ >\ -v.2\$

2393 \> > \defineF\finite_field\ -q.64\ -end\$

504
\begin{verbatim}
2394 ▶ ▶ -define-elts-vector-field-F-loop-0.64.1-end\end{verbatim}
2395 ▶ ▶ -with-F-do\end{verbatim}
2396 ▶ ▶ -finite_field_activity\end{verbatim}
2397 ▶ ▶ ▶ -field_reduction-"F64_over_F2".2-8.8-elts-end
2398 ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\end{verbatim}
2399 ▶ ▶ -input_csv_file-F64_over_F2.csv\end{verbatim}
2400 ▶ ▶ -box_width-40-bit_depth-24\end{verbatim}
2401 ▶ ▶ -partition-"6,6,6,6,6,6,6"."6,6,6,6,6,6,6"-end
2402 ▶ open-F64_over_F2_draw.bmp
2403 ▶ #pdflatex-field_reduction_Q64_q2_8_8.tex
2404 ▶ #open-field_reduction_Q64_q2_8_8.pdf
2405
2406
2407
2408
2409
2410 F_8_Nth_roots_21:
2411 ▶ $(ORBITER_PATH)orbiter.out-v.3\end{verbatim}
2412 ▶ ▶ -define-F-finite_field-q-8-override_polynomial-11-end\end{verbatim}
2413 ▶ ▶ -with-F-do-finite_field_activity-nth_roots_21-end
2414 ▶ pdflatex-Nth_roots_q8_n21.tex
2415 ▶ open-Nth_roots_q8_n21.pdf
2416
2417
2418
2419
2420 F_8_vandermonde:
2421 ▶ $(ORBITER_PATH)orbiter.out-v.3\end{verbatim}
2422 ▶ ▶ -define-F-finite_field-q-8-end\end{verbatim}
2423 ▶ ▶ -with-F-do-finite_field_activity\end{verbatim}
2424 ▶ ▶ ▶ -Vandermonde_matrix\end{verbatim}
2425 ▶ ▶ -end
2426
2427
2428
2429 F_1024_vandermonde:
2430 ▶ $(ORBITER_PATH)orbiter.out-v.3\end{verbatim}
2431 ▶ ▶ -define-F-finite_field-q-1024-end\end{verbatim}
2432 ▶ ▶ -with-F-do-finite_field_activity\end{verbatim}
2433 ▶ ▶ ▶ -Vandermonde_matrix\end{verbatim}
2434 ▶ ▶ -end
2435 ▶ rm-Vandermonde_1024.csv
2436 ▶ rm-Vandermonde_inv_1024.csv
2437
2438 #User.time:-0:46
2439
2440

505
NTT_k4_q17.cpp:

```
$ (ORBITER_PATH) orbiter.out -v 3 \n  -define F -finite_field q 17 -end \n  -with F -do -finite_field_activity -NTT 4 17 -end
```

F_17_NTT_compile: NTT_k4_q17.cpp

```
$ (MY_CPP) -NTT_k4_q17.cpp $(CPPFLAGS) \n  $(LIB) $(LFLAGS) -o NTT_k4_q17.out
```

# ToDo:

FGDTP:

```
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt F_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt Fv_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt AAv_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt G_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt D_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt T_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt Tv_k4.csv 20 8
```

no:

```
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt P_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt G_r_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt Dr_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt Tr_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt Pr_k3.csv 20 8
```

```
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt G_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt D_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt T_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt P_k3.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt G_k2.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt D_k2.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt T_k2.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt P_k2.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix ntt_the_P_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix_nnt_L_k4.csv 20 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix_fourier_q17 N16.csv 40 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix_fourier_q17 N8.csv 40 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix_fourier_q17 N4.csv 40 8
$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix_fourier_q17 N2.csv 40 8
```

```
Polyomial_ring:

```bash
$(ORBITER_PATH) orbiter.out -v 3 \n
-define F -finite_field -q 4 -end \n
-define R -polynomial_ring -field F \n
-number_of_variables 4 \n
-homogeneous_of_degree 3 \n
-variables "x0,x1,x2,x3" "x_0,x_1,x_2,x_3" \n
-end \n```

PG_3_2_easy:

```bash
$(ORBITER_PATH) orbiter.out -v 3 \n
-define F -finite_field -q 2 -end \n
-define P -projective_space 3 F -end \n```

PG_1_16:

```bash
$(ORBITER_PATH) orbiter.out \n
-define F -finite_field -q 16 -end \n
-define P -projective_space 1 F -end \n
-with P -do -projective_space_activity \n
-cheat_sheet \n
-end \n
pdflatex PG_1_16.tex \n```
PG_2.4:
\$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2\$
\$\text{define}\, F\cdot\text{-finite\_field}\cdot-q.4\cdot-\text{end}\$
\$\text{define}\, P\cdot\text{-projective\_space}\cdot2\cdot F\cdot-\text{end}\$
\$\text{with}\, P\cdot\text{-do}\cdot\text{-projective\_space\_activity}\$
\$\text{cheat\_sheet}\$
\$\text{end}\$
pdflatex\,PG_2.4.tex
open-PG_2.4.pdf

PG_2.13:
\$(\text{ORBITER\_PATH})\text{orbiter.out}\$
\$\text{define}\, F\cdot\text{-finite\_field}\cdot-q.13\cdot-\text{end}\$
\$\text{define}\, P\cdot\text{-projective\_space}\cdot2\cdot F\cdot-\text{end}\$
\$\text{with}\, P\cdot\text{-do}\cdot\text{-projective\_space\_activity}\$
\$\text{cheat\_sheet}\$
\$\text{end}\$
pdflatex\,PG_2.13.tex
open-PG_2.13.pdf

PG_2.64:
\$(\text{ORBITER\_PATH})\text{orbiter.out}\$
\$\text{define}\, F\cdot\text{-finite\_field}\cdot-q.64\cdot-\text{end}\$
\$\text{define}\, P\cdot\text{-projective\_space}\cdot2\cdot F\cdot-\text{end}\$
\$\text{with}\, P\cdot\text{-do}\cdot\text{-projective\_space\_activity}\$
\$\text{cheat\_sheet}\$
\$\text{end}\$
pdflatex\,PG_2.64.tex
open-PG_2.64.pdf

PG_3.2:
\$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.0\$
\$\text{define}\, F\cdot\text{-finite\_field}\cdot-q.2\cdot-\text{end}\$
\$\text{define}\, P\cdot\text{-projective\_space}\cdot3\cdot F\cdot-\text{end}\$
\$\text{with}\, P\cdot\text{-do}\cdot\text{-projective\_space\_activity}\$
\$\text{cheat\_sheet}\$

\text{pdflatex}\,PG_3.2.tex
open-PG_3.2.pdf
2582 ▷ ▷ -end
2583 ▷ pdflatex PG_3_2.tex.
2584 ▷ open PG_3_2.pdf
2585
2586
2587 PG_3_4:
2588 ▷ $(ORBITER_PATH)orbiter.out-v.10\n2589 ▷ ▷ -define F-finite_field-q 4 -end\n2590 ▷ ▷ -define P-projective_space 3-F-end\n2591 ▷ ▷ -with P-do-projective_space_activity\n2592 ▷ ▷ ▷ -cheat_sheet\n2593 ▷ ▷ -end
2594 ▷ pdflatex PG_3_4.tex.
2595 ▷ open PG_3_4.pdf
2596
2597 PG_3_5:
2598 ▷ $(ORBITER_PATH)orbiter.out\n2599 ▷ ▷ -define F-finite_field-q 5 -end\n2600 ▷ ▷ -define P-projective_space 3-F-end\n2601 ▷ ▷ -with P-do-projective_space_activity\n2602 ▷ ▷ ▷ -cheat_sheet\n2603 ▷ ▷ -end
2604 ▷ pdflatex PG_3_5.tex.
2605 ▷ open PG_3_5.pdf
2606
2607
2608 PG_3_7:
2609 ▷ $(ORBITER_PATH)orbiter.out\n2610 ▷ ▷ -define F-finite_field-q 7 -end\n2611 ▷ ▷ -define P-projective_space 3-F-end\n2612 ▷ ▷ -with P-do-projective_space_activity\n2613 ▷ ▷ ▷ -cheat_sheet\n2614 ▷ ▷ -end
2615 ▷ pdflatex PG_3_7.tex.
2616 ▷ open PG_3_7.pdf
2617
2618
2619
2620
2621 PG_3_8:
2622 ▷ $(ORBITER_PATH)orbiter.out\n2623 ▷ ▷ -define F-finite_field-q 8 -end\n2624 ▷ ▷ -define P-projective_space 3-F-end\n2625 ▷ ▷ -with P-do-projective_space_activity\n2626 ▷ ▷ ▷ -cheat_sheet\n2627 ▷ ▷ -end
2628 ▷ pdflatex PG_3_8.tex.
$(\text{ORBITER PATH})$orbiter.out
\[\begin{align*}
\text{define F - finite field - q \text{\texttt{16}} - end} \\
\text{define P - projective space - 3 F - end} \\
\text{-with P - do - projective space activity - end} \\
\text{-cheat sheet - end} \\
pdflatex PG_3_{16}.tex \\
open-PG_3_{16}.pdf
\end{align*}\]

$(\text{ORBITER PATH})$orbiter.out
\[\begin{align*}
\text{define F - finite field - q \text{\texttt{25}} - end} \\
\text{define P - projective space - 3 F - end} \\
\text{-with P - do - projective space activity - end} \\
\text{-cheat sheet - end} \\
pdflatex PG_3_{25}.tex \\
open-PG_3_{25}.pdf
\end{align*}\]

$(\text{ORBITER PATH})$orbiter.out
\[\begin{align*}
\text{define F - finite field - q \text{\texttt{3}} - end} \\
\text{define P - projective space - 4 F - end} \\
\text{-with P - do - projective space activity - end} \\
\text{-cheat sheet - end} \\
pdflatex PG_4_{3}.tex \\
open-PG_4_{3}.pdf
\end{align*}\]
Section 4.2: Indexing Points

SECTION_INDEXING_POINTS:

PG_2.4_rank_point:

$\text{ORBITER\_PATH}\text{orbiter.out}\text{-v.2}\$

\$\text{define-F\_finite_field}\text{-q.4\_end}\$

\$\text{with-F\_do\_finite_field\_activity}\$

\$\text{-rank_point\_in\_PG_2\"3,3,1\"\_end}\$

# geometry_global::do_rank_point_in_PG::coeff::\(3,3,1\)\_has\_rank:20

elliptic_curve_b1_c3_q11.txt:

$\text{ORBITER\_PATH}\text{orbiter.out}\text{-v.2}\$

\$\text{define-F\_finite_field}\text{-q.11\_end}\$

\$\text{define-P\_projective_space2\_F\_end}\$

\$\text{define-EC\_geometric_object}\text{P}\$

\$\text{-elliptic_curve1.3}\$

\$\text{-end}\$

\$\text{-with-EC\_do\_combinatorial\_object\_activity\_save}\$

\$\text{-end}\$

PG_2.2_incidence_matrix:

$\text{ORBITER\_PATH}\text{orbiter.out}\text{-v.2}\$

\$\text{define-F\_finite_field}\text{-q.2\_end}\$

\$\text{define-P\_projective_space2\_F\_end}\$

\$\text{with-P\_do\_projective_space\_activity}\$

\$\text{-export_point\_line\_incidence\_matrix}\$

\$\text{-end}\$

\$\text{ORBITER\_PATH}\text{orbiter.out}\text{-v.2}\$

\$\text{define-all_one\_vector\_repeat-1.7\_end}\$

\$\text{-draw\_matrix}\$

\$\text{-input\_csv\_file}\text{PG_n2_q2_incidence_matrix.csv}\$

\$\text{-box\_width:20\_bit\_depth:8}\$

\$\text{-partition-3}\$
PG_2.4.incidence_matrix:
$\text{(ORBITER PATH)}$orbiter.out -v 2 \n-define F -finite_field -q 4 -end \n-define P -projective_space -2 F -end \n-with P -do -projective_space_activity \n-export point line incidence_matrix \n-end
$\text{(ORBITER PATH)}$orbiter.out -v 2
-define all one -vector -repeat 1 21 -end
-draw_matrix
-define all one -vector -repeat 1 73 -end
-draw_matrix
-input_csv_file $\text{PG}_2.4 \text{ incidence_matrix.csv}$
-box_width 20 -bit_depth 8
-partition 3
-end
$\text{(ORBITER PATH)}$orbiter.out -v 2
-open $\text{PG}_2.4 \text{ incidence_matrix_draw.bmp}$

PG_2.8.incidence_matrix:
$\text{(ORBITER PATH)}$orbiter.out -v 2 \n-define F -finite_field -q 8 -end \n-define P -projective_space -2 F -end \n-with P -do -projective_space_activity \n-export point line incidence_matrix \n-end
$\text{(ORBITER PATH)}$orbiter.out -v 2
-define all one -vector -repeat 1 73 -end
-draw_matrix
-input_csv_file $\text{PG}_2.8 \text{ incidence_matrix.csv}$
-box_width 20 -bit_depth 8
-partition 3
-end
$\text{(ORBITER PATH)}$orbiter.out -v 2
-open $\text{PG}_2.8 \text{ incidence_matrix_draw.bmp}$
SECTION FINITE DESARGUESIAN PROJECTIVE PLANES:

PG_2_16:

$\texttt{\$(ORBITER\_PATH)orbiter.out\ -draw_options-xin:20000-yin:20000\ -radius:200-line_width:0.3-nodes_empty-end\ -define:F-finite_field-q:16-end\ -define:P-projective_space-2-F-end\ -with:P-do-projective_space_activity\ -cheat_sheet-end}$

pdflatex PG_2_16.tex
open PG_2_16.pdf

PG_2_4_with_decomposition:

$\texttt{\$(ORBITER\_PATH)orbiter.out-v:2\ -define:F-finite_field-q:4-end\ -define:P-projective_space-2-F-end\ -with:P-do-projective_space_activity\ -cheat_sheet_for_decomposition_by_element_PG-end}$

pdflatex PG_2_4_singer.tex
open PG_2_4_singer.pdf

PG_2_4.incma_cyclic:

$\texttt{\$(ORBITER\_PATH)orbiter.out-v:4\ -list_arguments-end}$
Section 4.4: The Grassmannian

SECTION GRASSMANNIAN:

GR_3_2.2:

-define F-finite_field-q-2-end

-define R-vector-repeat-1.21-end

-define C-vector-repeat-1.21-end

-draw_matrix

-input_csv_file-PG_2.4_singer_incma_cyclic.csv

-box_width-40-bit_depth-24

-partition-3:R:C

-end

-open-PG_2.4_singer_incma_cyclic_draw.bmp

PG_2.4_incma_singer_sub_3:

-define F-finite_field-q-2-end

-input_csv_file-PG_2.4_singer_incma_subgroup_index_3.csv

-box_width-40-bit_depth-24

-partition-3:R:C

-end

-open-PG_2.4_singer_incma_subgroup_index_3_draw.bmp

PG_2.4_incma_singer_sub_7:

-define F-finite_field-q-2-end

-input_csv_file-PG_2.4_singer_incma_subgroup_index_7.csv

-box_width-40-bit_depth-24


-end

-open-PG_2.4_singer_incma_subgroup_index_7_draw.bmp
rank_lines: $(\text{ORBITER\_PATH})\text{orbiter.out}-v.2$

-define v1 -vector -format 3 \n-dense:1,0,2,2,0,1,1,2,1,0,2,0,0,1,1,2,1,0,2,2,0,1,2,1 \n-end \n-define v2 -vector -format 3 \n-dense:1,0,0,0,0,1,0,0,1,0,0,0,0,1,0,1,0,0,0,2,1 \n-end \n-define F -finite_field -q 3 \n-end \n-define P -projective_space -3 \n-end \n-with P -do \n-projective_space_activity \n-rank_lines_in PG v1 \n-end \n-with P -do \n-projective_space_activity \n-rank_lines_in PG v2 \n-end

#Section 4.5: Algebraic Sets

SECTION ALGEBRAIC SETS:

EC_11.txt:

-define F -finite_field -q 11 \n-end \n-define R -polynomial_ring -field F \n-number_of_variables 3 \n-homogeneous_of_degree 3 \n-end \n-define P -projective_space -2 \n-end \n-define EC -geometric_object P
Hirschfeld surface q4.txt:

```
2911 # define F - finite field - q^4 - end
2912 # define R - polynomial ring - field F - end
2913 # define P - projective space - 3 F - end
2914 # define H4 - geometric object P - end
2915 # with H4 - do - combinatorial object activity - save - end
```

Hirschfeld surface q16.txt:

```
2941 # define F - finite field - q^16 - end
2942 # define R - polynomial ring - field F - end
2943 # define P - projective space - 3 F - end
2944 # define H16 - geometric object P - end
2945 # with H16 - do - combinatorial object activity - save - end
```

# creates Hirschfeld surface q4.txt

# creates Hirschfeld surface q16.txt
#the-coefficient-vector-is-given-as-a-list-of-pairs.
#165=binomial(11,3)

Endrass_F7.txt:

```
▶ $(ORBITER_PATH)orbiter.out -v 2 \n▶ define F -finite_field -q 7 -end \n▶ define R -polynomial_ring -field F \n▶ number_of_variables 4 \n▶ homogeneous_of_degree 8 \n▶ -end \n▶ define eqn -vector -field F -sparse 165 \n▶ $(ENDRASS_SPARSE) -end \n▶ define P -projective_space 3 F -end \n▶ number_of_variables 4 \n▶ homogeneous_of_degree 8 \n▶ -end \n▶ define P -projective_space 3 F -end \n▶ number_of_variables 4 \n▶ homogeneous_of_degree 8 \n▶ -end
```

we created a set of 33 points, called Endrass_F7.txt

```
octic_prepare:  
▶ $(ORBITER_PATH)orbiter.out -v 4 \n▶ define A -vector -format 1 -dense "1,1,1,1" -end \n▶ define D -diophant \n▶ label octic_monomials \n▶ coefficient_matrix A \n▶ -RHS "8,8,1" \n▶ -x_min_global 0 -x_max_global 8 \n▶ -end \n▶ -with D -do \n▶ diophant_activity -solve mckay \n▶ -end
```

#Found 165 solutions with 210 backtrack steps

#165=binomial(11,3)
Section 4.6: The Klein-Quadric and Pluecker-coordinates

SECTION_KLEIN_QUADRIC_AND_PLUECKER_COORDINATES:

GR_4.2.2:
$\text{(ORBITER PATH)}\text{orbiter.out}\text{-v.2}$
$\\text{-define}\ F\text{-finite_field}\text{-q.2\-end}$
$\\text{-with}\ F\text{-do}\text{-finite_field_activity}$
$\\text{-cheat_sheet}\text{Gr.4.2\-end}$
$\text{pdflatex}\text{Gr.4.2.2.tex}$
$\text{open}\text{Gr.4.2.2.pdf}$

SECTION_ORTHOGONAL_SPACES:

Op_4.2:
$\text{(ORBITER PATH)}\text{orbiter.out}\text{-v.2}$
$\\text{-define}\ F\text{-finite_field}\text{-q.2\-end}$
$\\text{-define}\ \text{0-orthogonal_space}\text{1.4.2\-without_group\-end}$
$\\text{-with}\ \text{0\-do\-orthogonal_space_activity}$
$\\text{-cheat_sheet}\text{orthogonal\-end}$
$\text{pdflatex}\text{0.1.4.2_report.tex}$
$\text{open}\text{0.1.4.2_report.pdf}$

0.5.2_incidence_matrix.csv:
$\text{(ORBITER PATH)}\text{orbiter.out}\text{-v.2}$
$\\text{-define}\ F\text{-finite_field}\text{-q.2\-end}$
$\\text{-define}\ \text{0-orthogonal_space}\text{0.5.2\-without_group\-end}$
$\\text{-with}\ \text{0\-do\-orthogonal_space_activity}$
$\\text{-export}\text{point}\text{line}\text{incidence}\text{matrix}$
$\\text{-end}$
$\\text{(ORBITER PATH)}\text{orbiter.out}\text{-v.2}$
\begin{verbatim}
  3052  \$define-all_one_r-vector-repeat-1:15-end\
  3053  \$define-all_one_c-vector-repeat-1:15-end\
  3054  \$draw_matrix\
  3055  \$input_csv_file:0_5_2_incidence_matrix.csv\
  3056  \$box_width:20\$bit_depth:8\
  3057  \$partition:2\
  3058  \$all_one_r\$all_one_c\
  3059  \$end
  3060  \$open:0_5_2_incidence_matrix_draw.bmp
  3061
  3062
  3063
  3064  Op_6_2:
  3065  $(ORBITER_PATH)orbiter.out\$v:2\
  3066  \$define-F\$finite_field\$q:2\$end\
  3067  \$define-0\$orthogonal_space\$1:6:F\$without_group\$end\
  3068  \$with\$do\$orthogonal_space_activity\
  3069  \$end
  3070  \$pdflatex:0_1_6_2_report.tex\
  3071  \$open:0_1_6_2_report.pdf
  3072
  3073
  3074
  3075  Op_6_2_incidence_matrix.csv:
  3076  $(ORBITER_PATH)orbiter.out\$v:2\
  3077  \$define-F\$finite_field\$q:2\$end\
  3078  \$define-0\$orthogonal_space\$1:6:F\$without_group\$end\
  3079  \$with\$do\$orthogonal_space_activity\
  3080  \$end
  3081  \$$(ORBITER_PATH)orbiter.out\$v:2\
  3082  \$define-all_one_r-vector-repeat-1:35\$end\
  3083  \$define-all_one_c-vector-repeat-1:105\$end\
  3084  \$draw_matrix\
  3085  \$input_csv_file:0p_6_2_incidence_matrix.csv\
  3086  \$box_width:20\$bit_depth:8\
  3087  \$partition:2\
  3088  \$all_one_r\$all_one_c\
  3089  \$end
  3090  \$open:0p_6_2_incidence_matrix_draw.bmp
  3091
  3092
  3093
  3094  Op_6_2_with_group:
  3095  $(ORBITER_PATH)orbiter.out\$v:2\
  3096  \$define-F\$finite_field\$q:2\$end\
  3097  \$define-0\$orthogonal_space\$1:6:F\$end\
  3098  \$with\$do\$orthogonal_space_activity\
\end{verbatim}
#problem:
#error-message:
#stabilizer_chain_base_data::allocate_base_data-degree.is.too.large

Op_8_2:
\$(ORBITER_PATH)orbiter.out-v.2:\
define F=finite_field-q.2-end\ndefine 0-orthogonal_space:1.8:F:-without_group-end\nwith 0-do-orthogonal_space.activity:\n\-cheat_sheet_orthogonal-end
\$pdflatex-0.1.6.2_report.tex\n\open-0.1.6.2_report.pdf

Op_6_64:
\$(ORBITER_PATH)orbiter.out-v.2:\n\define F=finite_field-q.64-end\n\define 0-orthogonal_space:1.6:F:-without_group-end\n\with 0-do-orthogonal_space.activity:\n\-cheat_sheet_orthogonal-end
\pdflatex-0.1.6.4_report.tex\n\open-0.1.6.4_report.pdf

Op_6_64_line_rank:
\$(ORBITER_PATH)orbiter.out-v.4:\n\define F=finite_field-q.64-end\n\define 0-orthogonal_space:1.6:F:-end\n\with 0-do-orthogonal_space.activity:\n\-unrank_line_through_two_points-15447347:15225451:\n\end

#use-option:-without_group-to-skip_the_group..This.will.work:

Op_6_64_line_rank:
#this will create a basic report without the group:

Op_6.64_report:

 Op_6.64_report:

 Op_6.8.2:

 Op_6.8.2:

 #Section 4.8: Hermitian varieties

 SECTION_HERMITIAN_VARIETIES:

 H_2.4:

 H_2.4:

 H_2.9:

 H_2.9:
\begin{verbatim}
3193 \pdflatex \H_{2.9}.tex
3194 \open \H_{2.9}.pdf
3195
3196 #28 points: 6, 11, 9, 7, 14, 19, 17, 15, 80, 75, 78, 74, 35, 30, 33, 29, 80, 75, 78, 74, 35, 30, 33, 29, 62, 57, 60, 56, 26, 21, 24, 3, 22, 82, 64, 46.
3198
3199 \H_{3.4}:
3200 \$(\text{ORBITER}\text{PATH})\text{or}b\text{iter}\text{.}\text{out} -\text{v}2\
3201 \quad -\text{define}\text{-}\text{F}\text{-}\text{finite}\text{\_}\text{field}\text{-}\text{q}4\text{-}\text{end}\
3202 \quad -\text{with}\text{-}\text{F}\text{-}\text{do}\text{-}\text{finite}\text{\_}\text{field}\text{\_}\text{activity}\
3203 \quad -\text{cheat}\text{-}\text{sheet}\text{\_}\text{hermitian}\text{-}\text{3}\text{-}\text{end}
3204 \pdflatex \H_{3.4}.tex
3205 \open \H_{3.4}.pdf
3206
3207 #\H_{3.4}:=\text{the Hirschfeld surface}
3208
3209 \end{verbatim}

\section*{SECTION PROJECTIVE SPACE ADVANCED TOPICS:}

\begin{verbatim}
3220 \fix_structure_2A:
3221 \$(\text{ORBITER}\text{PATH})\text{or}b\text{iter}\text{.}\text{out} -\text{v}2\
3222 \quad -\text{define}\text{-}\text{F}\text{-}\text{finite}\text{\_}\text{field}\text{-}\text{q}4\text{-}\text{end}\
3223 \quad -\text{define}\text{-}\text{P}\text{-}\text{projective}\text{\_}\text{space}\text{-}\text{3}\text{-}\text{F}\text{-}\text{end}\
3224 \quad -\text{with}\text{-}\text{P}\text{-}\text{do}\
3225 \quad -\text{projective}\text{\_}\text{space}\text{\_}\text{activity}\
3226 \quad -\text{cheat}\text{\_}\text{sheet}\text{\_}\text{for}\text{\_}\text{decomposition}\text{\_}\text{by}\text{\_}\text{element}\text{\_}\text{PG}\text{\_}1\
3227 \quad -\text{fix}\text{-}\text{structure}_2A\
3228 \quad -\text{end}\
3229 \pdflatex \fix_structure_2A.tex
3230 \open \fix_structure_2A.pdf
3231
3232 \fix_structure_2B:
3233 \$(\text{ORBITER}\text{PATH})\text{or}b\text{iter}\text{.}\text{out} -\text{v}2\
3234 \quad -\text{define}\text{-}\text{F}\text{-}\text{finite}\text{\_}\text{field}\text{-}\text{q}4\text{-}\text{end}\
3235 \quad -\text{define}\text{-}\text{P}\text{-}\text{projective}\text{\_}\text{space}\text{-}\text{3}\text{-}\text{F}\text{-}\text{end}\
3236 \quad -\text{with}\text{-}\text{P}\text{-}\text{do}\
3237 \pdflatex \fix_structure_2B.tex
3238 \open \fix_structure_2B.pdf
3239
3240 \end{verbatim}

522
trans:
3245 $\left(\text{ORBITER PATH}\right)\text{orbiter.out-\text{-v-2}}$
3246 $\text{-define-F-finite\_field-q-4-\text{-end}}$
3247 $\text{-define-P-projective\_space-3-F-\text{-end}}$
3248 $\text{-with-P-do-\text{-end}}$
3249 $\text{-projective\_space\_activity-\text{-end}}$
3250 $\text{-cheat\_sheet\_for\_decomposition\_by\_element\_PG-1}$
3251 $\text{"1,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0"}$
3252 $\text{-fix\_structure-2C}$
3253 $\text{-end}$
3254 $\text{pdflatex\_fix\_structure-2C.tex}$
3255 $\text{open\_fix\_structure-2C.pdf}$
3256 $\text{del_Pezzo_F13ab\_report}$
3257 $\text{-define-F-finite\_field-q-13-\text{-end}}$
3258 $\text{-define-P-projective\_space-3-F-\text{-end}}$
3259 $\text{-define-f3-formula}$
3260 $\text{"del_Pezzo_F13a\"{\"del_Pezzo\F13a\"x,y,z\"}}$
3261 $\text{\"x*x*x*y*y*y+z*z+z+8*x*x*y+y+8*x*x*z+z+8*y*y*z+z\"}$
3262 $\text{-define-f4-formula}$
3263 $\text{\"del_Pezzo_F13b\"{\"del_Pezzo\F13b\"x,y,z\"}}$
3264 $\text{\"x*x*x+y*y*y+z*z+z-x*x*y*y\"}$
\define-del_Pezzo13-collection="f3,f4"\n\-with-P\-do\n\-projective_space_activity\n\-analyze_del_Pezzo_surface-del_Pezzo13."".\n\-end
\pdf_latex-del_Pezzo_F13b_report.tex
\open-del_Pezzo_F13b_report.pdf

\del_Pezzo_F13a_points.txt:
\$(\textbf{ORBITER\_PATH})\texttt{orbiter.out\-v.3}\\
\-define-F\-finite_field\-q\-13\-end\n\-define-P\-projective_space\-3\-F\-end\n\-define-f1\-formula\"del_Pezzo_F9\".\n\-define-f2\-formula\"del_Pezzo_F11\".\n\-define-f3\-formula\"del_Pezzo_F13a\".\n\-define-f4\-formula\"del_Pezzo_F13b\".\n\-with-P\-do\n\-projective_space_activity\n\-analyze_del_Pezzo_surface-del_Pezzo13."".\n\-end
\pdf_latex-del_Pezzo_F13a_report.tex
\pdf_latex-del_Pezzo_F13b_report.tex
\open-del_Pezzo_F13a_report.pdf
\open-del_Pezzo_F13b_report.pdf

\#\texttt{dot\-\textit{Tpng}\-del_Pezzo_F13a.gv\>del_Pezzo_F13a.png}
\#\texttt{open\-del_Pezzo_F13a.png}
\#\texttt{writes\-del_Pezzo_F13a_points.txt}

\del_Pezzo_169:
\$(\textbf{ORBITER\_PATH})\texttt{orbiter.out\-v.3}
define \( F \) - finite field - \( q \) - end \( \backslash \)
define \( P \) - projective space - 3 - \( F \) - end \( \backslash \)
define \( f_3 \) - formula - "del Pezzo_F169a" \( \backslash \)
define \( f_4 \) - formula - "del Pezzo_F169b" \( \backslash \)
define \( O \) - geometric object \( P \) \( \backslash \)
define \( O \) - elliptic quadric ovoid \( \backslash \)
define \( O \) - with \( O \) - do - combinatorial object activity - save \( \backslash \)
define \( O \) - end \( \backslash \)
define \( O \) - ovoid ST \( \backslash \)
define \( O \) - end \( \backslash \)

SECTION GEOMETRIC OBJECTS:

elliptic quadric ovoid q8:

ovoid ST q8:
Edge_curve_17:

Edge_curve_17_line_type:

Edge_q17:
Edge_curve_q23_line_type:
$\$(ORBITER\_PATH)\$\texttt{\textbackslash orbiter.out-v.2}\$
$\$-define-F\_finite\_field-q.23\_end\$
$\$-define-P\_projective\_space-2-F\_end\$
$\$-define-E\_geometric\_object-P\$
$\$-set-\$(\texttt{EDGE\_CURVE\_Q23\_AS\_POINTS})\$
$\$-end\$
$\$-with-E\_do\$
$\$-combinatorial\_object\_activity\$
$\$-save\$
$\$-end\$
$\$-with-E\_do\$
$\$-combinatorial\_object\_activity\$
$\$-line\_type\$
$\$-end\$
$\$-print\_symbols

SECTION\_PERMUTATION\_GROUPS:
C13:

```
$\text{(ORBITER\_PATH)}\text{orbit}.\text{out} -v.5\$

```

```
$\text{-define-gens-vector-dense$(GEN\_C13)-end}\$
```

```
$\text{-define-G-permutation\_group}\$
```

```
$\text{-bsgs-C13-C_{13}-13:13:0:1}\$
```

```
$\text{-define gens}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-export\_orbiter}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-export\_group\_table}\$
```

```
$\text{-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-report}\$
```

```
$\text{-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-save\_elements\_csv\"C13\_elts.csv\"}\$
```

```
$\text{-end}\$
```

```
pdflatex-C13\_report.tex
```

```
open-C13\_report.pdf
```

Symmetric_3:

```
$\text{(ORBITER\_PATH)}\text{orbit}.\text{out} -v.10\$
```

```
$\text{-define-G-permutation\_group-symmetric\_group-3-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-export\_orbiter}\$
```

```
$\text{-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-export\_group\_table}\$
```

```
$\text{-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
$\text{-print\_elements\_tex}\$
```

```
$\text{-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group\_theoretic\_activity}\$
```

```
528
```
Symmetric_3:

```bash
$(ORBITER_PATH)orbiter.out --v.10
```

Symmetric_4:

```bash
$(ORBITER_PATH)orbiter.out --v.10
```

Symmetric_5:

```bash
$(ORBITER_PATH)orbiter.out --v.10
```

Symmetric_6:

```bash
$(ORBITER_PATH)orbiter.out --v.10
```

Symmetric_7:

```bash
$(ORBITER_PATH)orbiter.out --v.10
```
```bash
#pdflatex Perm4_elements_tree.tex
#open Perm4_elements_tree.pdf

ToDo:
```
Symmetric_4_stab:

$\text{(ORBITER PATH)orbiter.out\,-v\,-20\,}$

$\text{define-G\,-permutation\,group\,-symmetric\,group\,-4\,-end\,}$

$\text{define-G\,-do\,}$

$\text{-group\,theoretic\,activity\,}$

$\text{orbits\,on\,points\,}$

$\text{stabilizer\,of\,orbit\,rep\,0\,}$

$\text{end}$

$\text{(ORBITER PATH)orbiter.out\,-v\,-2\,}$

$\text{define-gens\,-vector\,-file\,Perm4\,stab\,orb\,0\,gens\,csv\,-end\,}$

$\text{define-G\,-permutation\,group\,}$

$\text{bsgs\,Perm4\,stab\,orb\,0\,"Sym3"\,4\,6\,"0,1,2"\,2\,gens\,-end\,}$

$\text{define-Gr\,-modified\,group\,-from\,G\,}$

$\text{report\,}$

$\text{end}$

PGL_4_2_export:

$\text{(ORBITER PATH)orbiter.out\,-v\,-2\,}$

$\text{define-F\,-finite\,field\,-q\,2\,-end\,}$

$\text{define-G\,-linear\,group\,-PGL\,4\,-F\,-end\,}$

$\text{define-G\,-do\,}$

$\text{-group\,theoretic\,activity\,}$

$\text{-report\,}$

$\text{end}$

$\text{pdflatex\,PGL_4_2_report.tex}$

$\text{open\,PGL_4_2_report.pdf}$

C13.as_subgroup:

$\text{(ORBITER PATH)orbiter.out\,-v\,-10\,}$

$\text{define-G\,-permutation\,group\,-symmetric\,group\,-13\,}$

$\text{define-G\,-do\,}$

$\text{-group\,theoretic\,activity\,}$

$\text{-export\,orbiter\,}$
3662 ▶ ▶ -end\n3663 ▶ ▶ -with G-do\n3664 ▶ ▶ -group_theoretic_activity\n3665 ▶ ▶ ▶ -report\n3666 ▶ ▶ -end\n3667 ▶ ▶ -with G-do\n3668 ▶ ▶ -group_theoretic_activity\n3669 ▶ ▶ ▶ -save_elements_csv"C13_elts.csv"\n3670 ▶ ▶ -end
3671 ▶ #pdflatex-Perm13_Subgroup_C13_13_report.tex
3672 ▶ #open-Perm13_Subgroup_C13_13_report.pdf
3673
3674
3675
3676 #Section 5.2: Linear Groups
3677
3678
3679
3680 SECTION_LINEAR_GROUPS:
3681
3682
3683 PGL_3_2:
3684 ▶ $(ORBITER_PATH)orbiter.out-v.2\n3685 ▶ ▶ -define F-finite_field-q 2-end\n3686 ▶ ▶ -define G-linear_group-PGL 3-F-end\n3687 ▶ ▶ -with G-do\n3688 ▶ ▶ -group_theoretic_activity\n3689 ▶ ▶ ▶ -report\n3690 ▶ ▶ -end
3691 ▶ pdflatex-PGL_3_2_report.tex
3692 ▶ open-PGL_3_2_report.pdf
3693
3694 PGL_4_2:
3695 ▶ $(ORBITER_PATH)orbiter.out-v.2\n3696 ▶ ▶ -define F-finite_field-q 2-end\n3697 ▶ ▶ -define G-linear_group-PGL 4-F-end\n3698 ▶ ▶ -with G-do\n3699 ▶ ▶ -group_theoretic_activity\n3700 ▶ ▶ ▶ -report\n3701 ▶ ▶ -end
3702 ▶ pdflatex-PGL_4_2_report.tex
3703 ▶ open-PGL_4_2_report.pdf
3704
3705
3706 L_5_3:
3707 ▶ $(ORBITER_PATH)orbiter.out-v.2\n3708 ▶ ▶ -define F-finite_field-q 3-end\n532
#PSL(5,3):
Order 237783237120 = 121*120*117*108*81*16

#PSL(4,5):
Order 7254000000

PGGL_3.4:
Order 237783237120 = 121*120*117*108*81*16

#PSL(3,4):
Order 7254000000
PGGL_3:8:
3760 \>$\$(ORBITER_PATH)orbiter.out$-v.5$
3761 \>$\define\G\lineargroup\PGGL_3:8\end$
3762
3763
3764 PGGL_3:8_report:
3765 \>$\$(ORBITER_PATH)orbiter.out$-v.3$
3766 \>$\define\G\lineargroup\PGGL_3:8\end$
3767 \>$\with\G\end$
3768 \>$\grouptheoreticactivity$
3769 \>$\report$
3770 \>$\end$
3771 \>$\pdflatex\PGGL_3:8_report.tex$
3772 \>$\open\PGGL_3:8_report.pdf$
3773
3774
3775 AGL_1:27:
3776 \>$\$(ORBITER_PATH)orbiter.out$-v.2$
3777 \>$\define\F\finitefield$-q.27\end$
3778 \>$\define\G\lineargroup\AGL_1:F\end$
3779 \>$\with\G\end$
3780 \>$\grouptheoreticactivity$
3781 \>$\report$
3782 \>$\end$
3783 \>$\pdflatex\AGL_1:27_report.tex$
3784 \>$\open\AGL_1:27_report.pdf$
3785
3786 \>\$\order\$720$
3787
3788
3789 SP_4:2:
3790 \>$\$(ORBITER_PATH)orbiter.out$-v.2$
3791 \>$\define\F\finitefield$-q.2\end$
3792 \>$\define\G\lineargroup\GL_4:F$
3793 \>$\symplecticgroup$
3794 \>$\end$
3795 \>$\with\G\end$
3796 \>$\grouptheoreticactivity$
3797 \>$\report$
3798 \>$\end$
3799 \>$\pdflatex\GL_4:2_Sp_4:2_report.tex$
3800 \>$\open\GL_4:2_Sp_4:2_report.pdf$
3801
3802 \>$\order\$720
PSP_4_4:

```bash
$(ORBITER_PATH)orbiter.out-v.2:\n```

```bash
define-F-finite_field-q.4-end\n```

```bash
define-G-linear_group-PGL.4-F:\n```

```bash
define-symplectic_group-\n```

```bash
with-G-do-\n```

```bash
group_theoretic_activity-\n```

```bash
report-\n```

```bash
end\n```

```bash
pdflatex PGL.4_4_Sp.4_4_report.tex
```

```bash
open-PGL.4_4_Sp.4_4_report.pdf
```

```bash
#order-979200
```

PGO_5_2:

```bash
$(ORBITER_PATH)orbiter.out-v.2:\n```

```bash
define-F-finite_field-q.2-end\n```

```bash
define-G-linear_group-PGO.5-F.-end\n```

```bash
with-G-do-\n```

```bash
group_theoretic_activity-\n```

```bash
report-\n```

```bash
end\n```

```bash
pdflatex PGO.5_2_report.tex
```

```bash
open-PGO.5_2_report.pdf
```

PGGO_5_4:

```bash
$(ORBITER_PATH)orbiter.out-v.2:\n```

```bash
define-F4-finite_field-q.4-end\n```

```bash
define-G-linear_group-PGGO.5-F.4.-end\n```

```bash
with-G-do-\n```

```bash
group_theoretic_activity-\n```

```bash
report-\n```

```bash
end\n```

```bash
pdflatex PGGO.5_4_report.tex
```

```bash
open-PGGO.5_4_report.pdf
```

PGOp_6_2:

```bash
$(ORBITER_PATH)orbiter.out-v.2:\n```

```bash
define-F-finite_field-q.2-end\n```

```bash
define-G-linear_group-PGOp.6-F.-end\n```

535
The following two groups are isomorphic:

**PSP\(_{6,2}\):**

```plaintext
$(\text{ORBITER\_PATH})\text{orbiter.out}-v.2$
```

**PGO\(_{7,2}\):**

```plaintext
$(\text{ORBITER\_PATH})\text{orbiter.out}-v.2$
```

---

536
Section 5.3: Subgroups

J1:
\$\text{G}\text{-linear group -PGL}7\cdot11\text{-Janko1}\text{-end}\$
\$\text{G}\text{-do}\$
\$\text{group-theoretic activity}\$
\$\text{-report}\$
\$\text{-end}\$
\$\text{pdflatex PGL}7\cdot11\text{Subgroup Janko1 report.tex}\$
\$\text{open-PGL}7\cdot11\text{Subgroup Janko1 report.pdf}\$

PGL\_3\cdot11\_singer:
\$\text{G}\text{-linear group -PGL}3\cdot11\text{-singer\_19}\text{-end}\$
\$\text{G}\text{-do}\$
\$\text{group-theoretic activity}\$
\$\text{-report}\$
\$\text{-end}\$
\$\text{pdflatex PGL}3\cdot11\text{Singer}3\cdot11\_19\text{report.tex}\$
\$\text{open-PGL}3\cdot11\text{Singer}3\cdot11\_19\text{report.pdf}\$

PGL\_3\cdot11\_singer\_and\_frobenius:
\$\text{G}\text{-linear group -PGL}3\cdot11\text{-singer\_and\_frobenius\_19}\text{-end}\$
\$\text{G}\text{-do}\$
\$\text{group-theoretic activity}\$
\$\text{-report}\$
\$\text{-end}\$
\$\text{pdflatex PGL}3\cdot11\text{Singer\_and\_Frob3}3\cdot11\_19\text{report.tex}\$
\$\text{open-PGL}3\cdot11\text{Singer\_and\_Frob3}3\cdot11\_19\text{report.pdf}\$

PG\_2\cdot4\_order\_21:
\$\text{G}\text{-linear group -PGL}3\cdot4\text{-end}\$
\$\text{G}\text{-do}\$
\$\text{group-theoretic activity}\$
quaternion:

```latex
\texttt{-search_element_of_order:21}\backslash
\texttt{-end}
```

cube group:

```latex
\texttt{-define\_G\_linear\_group\_SL:2:3}\backslash
\texttt{-subgroup\_by\_generators:\textquote{quaternion\textasciitilde8\textasciitilde3}\backslash
\texttt{-\textbackslash
\texttt{-with\_G\_do}\backslash
\texttt{-group\_theoretic\_activity}\backslash
\texttt{-print\_elements\_tex}\backslash
\texttt{-group\_table}\backslash
\texttt{-report}\backslash
\texttt{-end}\backslash
```

tetra group:
3991  
3992  
3993  
3994  
3995  
3996  
3997  
3998  
3999  
4000  
4001  
4002  
4003  
4004  
4005  
4006  
4007  
4008  
4009  
4010  
4011  
4012  
4013  
4014  
4015  
4016  
4017  
4018  
4019  
4020  
4021  
4022  
4023  
4024  
4025  
4026  
4027  
4028  
4029  
4030  
4031  
4032  
4033  
4034  
4035  
4036  
4037  

Hesse_group:

#Hesse_group:

Weyl_E8:

539
GROUP $E_8$)

-define G-linear_group-GL-8-3-

subgroup_by_generators-

"Weyl_E8".696729600.2-

$(GENERATORS_WEYL_GROUP_E8)-

-end-

-with G-do-

-end-

define -G-linear_group-PGL-3-F-

subgroup_by_generators."U_3_3".6048.2-

"1,6,4,-5,0,6,8,5,1,-

"6,2,1,7,8,4,-0,6,6"-

-end-

-with G-do-

-end-

pdflatex-GL-8-3-Subgroup_Weyl_E8_696729600_report.tex

open-GL-8-3-Subgroup_Weyl_E8_696729600_report.pdf


SECTION_LINEAR_GROUPS_ADVANCED_TOPICS:

U_3_3:

$(ORBITER_PATH)orbiter.out-v.3-

-define F-finite_field-q-9-override_polynomial."17"-end-

-define G-linear_group-PGL-3-F-

subgroup_by_generators."U_3_3".6048.2-

"1,6,4,-5,0,6,8,5,1,-

"6,2,1,7,8,4,-0,6,6"-

-end-

-with G-do-

-end-

pdflatex-PGL-3_9_Subgroup_U_3_3_6048_report.tex

open-PGL-3_9_Subgroup_U_3_3_6048_report.pdf

open-PGL-3_9_Subgroup_U_3_3_6048_report.pdf

540
PGL_2_3:

$(\text{ORBITER\_PATH})\text{oriter.out}\ -v\ 3:  
\text{define}\ -G\ -linear\_group\ -PGL\ 2\ 3\ -end\  
\text{with}\ -G\ -do\  
-\text{group\_theoretic\_activity}\  
\text{report}\  
\text{-group\_table}\  
\text{-end}\  
pdflatex PGL_2_3\_group\_table\_order\_24.txt  
pdflatex PGL_2_3\_report.txt  
open PGL_2_3\_group\_table\_order\_24.pdf  
open PGL_2_3\_report.pdf

#Co3:\text{-from}\-\text{-Conway}\text{-et}\text{-al.},\text{-1985}\text{-(ATLAS)}  
#order:=495766656000  
#Co3:\text{-from}\-\text{-the}\text{-paper}\text{-by}\text{-Suleiman}\text{-and}\text{-Wilson}\text{-1997}

Co3:

$(\text{ORBITER\_PATH})\text{oriter.out}\ -v\ 6:  
\text{define}\ -F\ -finite\_field\ -q\ 2\ -end\  
\text{define}\ -g1\ -vector\ -field\ F\ -format\ 22\ -compact\ $(\text{CONWAY}\_GEN1)\ -end\  
\text{define}\ -g2\ -vector\ -field\ F\ -format\ 22\ -compact\ $(\text{CONWAY}\_GEN2)\ -end\  
\text{define}\ -gens\ -vector\ -concatenate\ g1\ -concatenate\ g2\ -end\  
\text{-subgroup\_by\_generators} \"Co3" \"495766656000\ "2\_gens\  
\text{-end}\  
\text{-with}\ -G\ -do\  
\text{-group\_theoretic\_activity}\  
\text{-report}\  
\text{-end}\  
pdflatex PGL_22_2\_Subgroup\_Co3\_495766656000\_report.txt  
open PGL_22_2\_Subgroup\_Co3\_495766656000\_report.pdf

#needs\ a\ lot\ of\ memory\ to\ run!

Ree_27:

$(\text{ORBITER\_PATH})\text{oriter.out}\ -v\ 6:  
\text{define}\ -F\ -finite\_field\ -q\ 27\ -override\_polynomial\ "34"\ -end\  
\text{define}\ -g1\ -vector\ -field\ F\ -format\ 7\ -dense\ $(\text{Ree}\_\text{gen1})\ -end\  
\text{define}\ -g2\ -vector\ -field\ F\ -format\ 7\ -dense\ $(\text{Ree}\_\text{gen2})\ -end\  
\text{define}\ -gens\ -vector\ -concatenate\ g1\ -concatenate\ g2\ -end\  
\text{-define}\ -G\ -linear\_group\ -PGL\ 7\ -F\
subgroup_by_generators "Ree_27" . "10073444472" . gens \n\n- with G - do \n- group_theoretic_activity \n- report \n- end

# needs a lot of memory to run!

Section 5.5: Induced Actions

SECTION_INDUCED_ACTIONS:

T3_on_tensors:

T3r1:
T4_on_tensors:

$\texttt{(ORBITER\_PATH)}\texttt{orbiter.out}\ -v\cdot 4$

-define-G-

-linear_group-GL_dq_wr_Sym_n\cdot 2\cdot 2\cdot 4$

-on_tensors-end-

-with-G-do-

-group_theoretic_activity-

-report-

-end

pdflatex\texttt{GL_2_2\_wreath_Sym4\_report.tex}

open-G\texttt{L_2_2\_wreath_Sym4\_report.pdf}

T4r1:

$\texttt{(ORBITER\_PATH)}\texttt{orbiter.out}\ -v\cdot 4$

-define-G-

-linear_group-GL_dq_wr_Sym_n\cdot 2\cdot 2\cdot 4$

-on_tensors-end-

-with-G-do-

-group_theoretic_activity-

-report-

-end

pdflatex\texttt{GL_2_2\_wreath_Sym4\_report.tex}

open-G\texttt{L_2_2\_wreath_Sym4\_report.pdf}

PGGL_2_8_on_conic:

$\texttt{(ORBITER\_PATH)}\texttt{orbiter.out}\ -v\cdot 4$

-define-G-

-linear_group-PGGL_2_8-PGL2OnConic-end-

-with-G-do-

-group_theoretic_activity-

-report-

-end

pdflatex\texttt{PGGL_2_8_OnConic_2_8\_report.tex}

open-PG\texttt{GL_2_8\_OnConic_2_8\_report.pdf}

SURFACE_q13_STAB="1,0,0,0,0,12,0,0,0,12,0,0,0,0,1,

1,0,0,0,0,12,0,0,0,1,0,0,0,0,12,0,0,0,0,1,

1,0,0,0,0,12,0,0,0,12,0,0,0,0,0,1,

0,1,0,0,1,0,0,0,0,1,0,0,0,0,1"

surface_q13_stab_on_tritangents_orbits:
$(\text{ORBITER}\ 	ext{PATH})/\text{orbiter.out}$ -v 30:

-define F -finite_field -q 13 -end
-define P -projective_space 3 F -end

-define surface S -q 13

-arc_lifting "0,1,2,3,43,113" -end
-define TriP -set -file

family Eckardt q13 a2 b1 tritangent planes.csv

-end

-define G -linear_group -PGL 4 13
-subgroup_by_generators "SURF_STAB"

24 4 $(\text{SURFACE}_q13\ 	ext{STAB})

-end

-define G on planes -modified_group -from G
-on_k_subspaces 3

-end

-define G -linear_group -PGL 4 2 -wedge_detached

-end

-define G -linear_group -PGL 4 2 -wedge_detached -end

-end

-end

-end

-end

-define G on planes -restricted_action TriP

-end

-end

-end

-end

-end

-end

-end

-end

-end

-end

-end

-end

-group_theoretic_activity

-report

-end

-end

-end

-end

-end

-end

-end

-end

-define G on planes -restricted_action TriP

-end

-end

-group_theoretic_activity

-orbits_on_points

-stabilizer

-end


PGL 4 2 wd:

$(\text{ORBITER}\ 	ext{PATH})/\text{orbiter.out}$ -v 12:

-define G -linear_group -PGL 4 2 -wedge_detached -end

-end

-end

-group_theoretic_activity


544
Section 5.6: Group-Theoretic Activities

SECTION

GROUP

THEORETIC

ACTIVITIES:

PGL_3_2_elements:

$\text{(ORBITER\ PATH)}\ orbiter.out\ -v\ 5\ \$

-define G -linear_group -PGL_3_2 -end

-with G -do

-group_theoretic_activity

-save_elements_csv"PGL_3_2_elements.csv"

-end

Sym_3_elements:

$\text{(ORBITER\ PATH)}\ orbiter.out\ -v\ 3\ \$

-define G -permutation_group -symmetric_group 3 -end

-with G -do

-with G -do

-group_theoretic_activity

-print_elements_tex

-end

$\text{(ORBITER\ PATH)}\ orbiter.out\ -v\ 20\ \$

-draw_options

-nodes
Cycle_13_power:
$(ORBITER_PATH)orbiter.out-v.5\$
-define-G-permutation_group-symmetric_group.13-end\$
-group_theoretic_activity\$
-consecutive_powers:"1,2,3,4,5,6,7,8,9,10,11,12,0".13\$
-end
pdflatex-Perm13_all_powers.tex
open-Perm13_all_powers.pdf

Cycle_12_power:
$(ORBITER_PATH)orbiter.out-v.5\$
-define-G-permutation_group-symmetric_group.12-end\$
-with-G-do\$
-group_theoretic_activity\$
-consecutive_powers:"1,2,3,4,5,6,7,8,9,10,11,0".12\$
-end
pdflatex-Perm12_all_powers.tex
open-Perm12_all_powers.pdf

PGL_3.4_singer:
$(ORBITER_PATH)orbiter.out-v.5\$
-define-G-linear_group-PGL.3.4-end\$
-with-G-do\$
-group_theoretic_activity\$
-find_singer_cycle\$
-end

GL_2.8_multiply:
$(ORBITER_PATH)orbiter.out-v.5\$
-define-G-linear_group-GL.2.8-end\$
GL_2.7.multiply:
$(\text{ORBITER\_PATH})$orbiter.out-v.5\`
-define-G-linear_group-GL.2.7-end\`
-\text{group\_theoretic\_activity}\`
-multiply."0,1,2,3"."4,5,6,7"\`
-end
-pdflatex-GL_2.8_mult.tex
-open-GL_2.8_mult.pdf

GL_2.7.inv:
$(\text{ORBITER\_PATH})$orbiter.out-v.5\`
-define-G-linear_group-GL.2.7-end\`
-\text{group\_theoretic\_activity}\`
-inverse."0,1,2,3"\`
-end
-pdflatex-GL_2.7_inv.tex
-open-GL_2.7_inv.pdf

GL_2.7.power:
$(\text{ORBITER\_PATH})$orbiter.out-v.5\`
-define-G-linear_group-GL.2.7-end\`
-\text{group\_theoretic\_activity}\`
-raise_to_the_power."0,1,2,3".2\`
-end
-pdflatex-GL_2.7_power.tex
-open-GL_2.7_power.pdf

PGL_3.2.classes:
$(\text{ORBITER\_PATH})$orbiter.out-v.3\`
-define-G-linear_group-PGL.3.2-end\`
-\text{group\_theoretic\_activity}\`
-\text{-classes\_based\_on\_normal\_form}\`
-end
normal_forms_PGL_4_4:

PGL_4_4_2A_rank:

PGL_4_4_2A_unrank:

#ToDo:

cent_2A:
PGL(4,5) rank:

```
$\text{(ORBITER_PATH)}\text{orbiter.out-v.6}\$
```

```
$\text{-define-G-linear_group-PGL(4,5)-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group_theoretic_activity}\$
```

```
$\text{-element_rank}:0,0,0,1,2,3,0,1,0,3,4,4,0,1,2,1\$
```

```
$\text{-end}\$
```

PGL(4,5) unrank:

```
$\text{(ORBITER_PATH)}\text{orbiter.out-v.6}\$
```

```
$\text{-define-G-linear_group-PGL(4,5)-end}\$
```

```
$\text{-with-G-do}\$
```

```
$\text{-group_theoretic_activity}\$
```

```
$\text{-element_unrank}:701459351\$
```

```
$\text{-end}\$
```

#eigen_3A:

```
$\text{(ORBITER_PATH)}\text{orbiter.out-v.6-finite_field_activity-q.5-eigenstuff-4}\$
```

```
$0,1,0,2,0,1,2,1,4,2,3,1,2,0,4,3\$
```

```
$\text{-end}\$
```

#eigen_3B:

```
$\text{(ORBITER_PATH)}\text{orbiter.out-v.6-finite_field_activity-q.5-eigenstuff-4}\$
```

```
$0,0,1,2,3,0,1,0,3,4,4,0,1,2,1\$
```

```
$\text{-end}\$
```

#element-of-order-31-in-PGL(4,5):

```
in-data[]:=-\{1,0,0,0,0,1,0,0,0,0,1,0,1,0,1,3,0\};
```

549
normal_forms_PGL_4_5:

\$\text{(ORBITER\_PATH)orbiter.out-v.7}\$

\$\text{-define-G-linear_group-PGL.4-5-end}\$

\$\text{-with-G-do}\$

\$\text{-group_theoretic_activity}\$

\$\text{-classes_based_on_normal_form}\$

\$\text{-end}\$

\$\text{pdflatex-PGL.4-5_classes_normal_form.tex}\$

\$\text{open-PGL.4-5_classes_normal_form.pdf}\$

\$\text{pdflatex}\$

\$\text{PGL.4-5_classes_normal_form.tex}\$

\$\text{open}\$

\$\text{PGL.4-5_classes_normal_form.pdf}\$

\$\text{open}\$

\$\text{PGL.4-5_classes_normal_form.csv}\$

\\

\$\text{SECTION\_GROUP\_THEORETIC\_ACTIVITIES\_BASED\_ON\_MAGMA:}\$

\$\text{PGGL.2-4_classes:}\$

\$\text{\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$}\$

\$\text{-define-G}\$

\$\text{-linear_group-PGGL.2-4}\$

\$\text{-end}\$

\$\text{-with-G-do}\$

\$\text{-group_theoretic_activity}\$

\$\text{-classes}\$

\$\text{-end}\$

\$\text{\$\text{(MAGMA\_PATH)magma-PGGL.2-4_classes.magma}\$}\$

\$\text{\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$}\$

\$\text{-define-G}\$

\$\text{-linear_group-PGGL.2-4}\$

\$\text{-end}\$

\$\text{-with-G-do}\$

\$\text{-group_theoretic_activity}\$

\$\text{-classes}\$

\$\text{-end}\$

\$\text{pdflatex-PGGL.2-4_classes.out.tex}\$

\$\text{open-PGGL.2-4_classes_out.pdf}\$

\$\text{open-PGGL.2-4_classes_out.csv}\$

\$\text{PGGL.2-4_cent.2A:}\$

\$\text{\$\text{(ORBITER\_PATH)orbiter.out-v.3}\$}\$
```
4552 ▷ ▷ -define G \n4553 ▷ ▷ -linear_group -PGGL-2-4 -end \n4554 ▷ ▷ -with G -do \n4555 ▷ ▷ -group_theoretic_activity \n4556 ▷ ▷ ▷ -centralizer_of_element "2A" "1,0,0,1,1" \n4557 ▷ ▷ ▷ -report \n4558 ▷ ▷ ▷ -end \n4559 ▷ ▷ $(MAGMA_PATH) magma -element_2A -centralizer.magma \n4560 ▷ ▷ $(ORBITER_PATH) orbiter.out -v 6 \n4561 ▷ ▷ ▷ -define G \n4562 ▷ ▷ ▷ -linear_group -PGGL-2-4 -end \n4563 ▷ ▷ ▷ -with G -do \n4564 ▷ ▷ ▷ -group_theoretic_activity \n4565 ▷ ▷ ▷ ▷ -centralizer_of_element "2A" "1,0,0,1,1" \n4566 ▷ ▷ ▷ ▷ -report \n4567 ▷ ▷ ▷ ▷ -end \n4568 ▷ ▷ pdflatex PGGL-2-4_elt -2A -centralizer.tex \n4569 ▷ ▷ open PGGL-2-4_elt -2A -centralizer.pdf 
```

```
4577 PGGL-3-4_classes: 
4578 ▷ ▷ $(ORBITER_PATH) orbiter.out -v 3 \n4579 ▷ ▷ ▷ -define G \n4580 ▷ ▷ ▷ -linear_group -PGGL-3-4 \n4581 ▷ ▷ ▷ -end \n4582 ▷ ▷ ▷ -with G -do \n4583 ▷ ▷ ▷ -group_theoretic_activity \n4584 ▷ ▷ ▷ ▷ -classes \n4585 ▷ ▷ ▷ ▷ -end \n4586 ▷ ▷ pdflatex PGGL-3-4_classes_out.tex \n4587 ▷ ▷ open PGGL-3-4_classes_out.pdf 
```

```
4590 #1,3,3,1,3,2,3,0,3,0, 
4591 # is an element of order 21 
4592 
4593 # - subgroup by generators "singer"."21" 1 
4594 # - "0,1,0,0,0,1,2,1,1" 
4596 
4597 
4598 
```
classes_PGGL_4:4:
   \$(\text{ORBITER PATH})\text{or}b\text{iter.out}\ -v\ -3\ \$
   \(\triangleright\triangleright\ -\text{m}a\text{gma}\_\text{path}\$(\text{MAGMA PATH})\ \$
   \(\triangleright\triangleright\ -\text{d}\i\f\text{ine}\_\text{G}\$
   \(\triangleright\triangleright\ -\text{l}\i\n\er\_\text{g}\_\text{r}\o\up\g\up\u\p\e\ -\text{PGGL}\_4\_4\ -\text{e}nd\$
   \(\triangleright\triangleright\ -\text{w}ith\_\text{G}\-\text{do}\$
   \(\triangleright\triangleright\ -\text{g}r\ou\p\_\text{th}e\o\ri\c\_\text{a}c\t\_\text{i}c\_\text{v}i\t\$
   \(\triangleright\triangleright\ \triangleright\ -\text{c}l\a\ss\e\s\$
   \(\triangleright\triangleright\ -\text{e}nd\$

\#\text{g}r\ou\p\_\text{order}\_1\text{974067200}:=.2^\text{13}\cdot3^4\cdot5^2\cdot7\cdot17$

\#\text{the}\_\text{-f}i\d\_\text{s}u\b\g\o\p\_\text{c}u\m\text{and}\text{i}\text{s}\text{too}\_\text{s}p\e\c\i\z\e\i\n\i\z\e\i\n$

subgroups_PGGL_4:5:
   \$(\text{ORBITER PATH})\text{or}b\text{iter.out}\ -v\ -6\ \$
   \(\triangleright\triangleright\ -\text{d}\i\f\text{ine}\_\text{G}\$
   \(\triangleright\triangleright\ -\text{l}\i\n\er\_\text{g}\_\text{r}\o\up\g\up\u\p\e\ -\text{PGL}\_4\_5\ -\text{e}nd\$
   \(\triangleright\triangleright\ -\text{w}ith\_\text{G}\-\text{d}o\$
   \(\triangleright\triangleright\ -\text{g}r\ou\p\_\text{th}e\o\ri\c\_\text{a}c\t\_\text{i}c\_\text{v}i\t\$
   \(\triangleright\triangleright\ \triangleright\ -\text{f}i\d\_\text{s}u\b\g\o\p\_\text{c}u\m\_3\$
   \(\triangleright\triangleright\ -\text{e}nd\$
   \(\triangleright\ pd\text{f}la\text{t}e\xop -\text{PGL}_4\_5\_\text{r}e\p\o\r\text{t}.t\e\x$
   \(\triangleright\ o\pen\_\text{PGL}_4\_5\_\text{r}e\p\o\r\text{t}.p\d\f$

\$\$

classes_PGGL_4:5:
   \$(\text{ORBITER PATH})\text{or}b\text{iter.out}\ -v\ -6\ \$
   \(\triangleright\triangleright\ -\text{d}\i\f\text{ine}\_\text{G}\$
   \(\triangleright\triangleright\ -\text{l}\i\n\er\_\text{g}\_\text{r}\o\up\g\up\u\p\e\ -\text{PGL}\_4\_5\ -\text{e}nd\$
   \(\triangleright\triangleright\ -\text{w}ith\_\text{G}\-\text{d}o\$
   \(\triangleright\triangleright\ -\text{g}r\ou\p\_\text{th}e\o\ri\c\_\text{a}c\t\_\text{i}c\_\text{v}i\t\$
   \(\triangleright\triangleright\ \triangleright\ -\text{c}l\a\ss\e\s\$
   \(\triangleright\triangleright\ -\text{e}nd\$
   \(\triangleright\ pd\text{f}la\text{t}e\xop -\text{PGL}_4\_5\_\text{c}l\a\ss\e\s_{\text{out}}.t\e\x$
   \(\triangleright\ o\pen\_\text{PGL}_4\_5\_\text{c}l\a\ss\e\s_{\text{out}}.p\d\f$

\#163\_\text{c}l\a\ss\e\s$

\#\text{t}w\o\_\text{c}l\a\ss\e\s\_\text{o}f\_\text{e}l\e\n\m\e\l\e\s\_\text{o}f\_\text{order}:3
#Order of element = 3 
Class size = 310000 
Centralizer order = 93600 
Normalizer order = 187200 

# of order 3 and with 0 fixed points.

#0,1,0,2,0,1,4,2,3,1,2,0,4,3,

#Class size = 10075000 
Centralizer order = 2880 
Normalizer order = 5760 

# of order 3 and with 6 fixed points.

#0,0,0,1,2,3,0,1,0,3,4,4,0,1,2,1,

\texttt{PGL}\_4\_5\_3B\_class\_again: 
\texttt{$(ORBITER\_PATH) orbiter.out -v 6 -define G\ \} 
\texttt{linear\_group -PGL\_4\_5 -end \} 
\texttt{with G -do \} 
\texttt{group\_theoretic\_activity \} 
\texttt{> conjucacy\_class\_of \} 
\texttt{> "0,0,0,1,2,3,0,1,0,3,4,4,0,1,2,1" \} 
\texttt{end \} 
\texttt{> \} 

\texttt{search\_primitive\_poly\_q5\_deg3: 
\texttt{$(ORBITER\_PATH) orbiter.out -v 6 \} 
\texttt{-search\_for\_primitive\_polynomial\_in\_range 5:5:3:3 \} 
\texttt{OK, we found an irreducible and primitive polynomial \} 
\texttt{X^{3}+X^{2}+2 \} 

\texttt{GL}\_3\_5\_singer\_power: 
\texttt{$(ORBITER\_PATH) orbiter.out -v 6 \} 
\texttt{-linear\_group -GL\_3\_5 -end \} 
\texttt{with G -do \} 
\texttt{group\_theoretic\_activity \} 
\texttt{> raise\_to\_the\_power \} 
\texttt{> "0,1,0,-0,0,1,3,0,4" 31 \} 
\texttt{end \} 
\texttt{pdf:\_latex -GL\_3\_5\_power.tex \} 
\texttt{open -GL\_3\_5\_power.pdf \} 

\texttt{PGL}\_4\_5\_norm\_31: 
\texttt{$(ORBITER\_PATH) orbiter.out -v 6 \} 
\texttt{-linear\_group -PGL\_4\_5 -end \} 
\texttt{with G -do \} 
\texttt{group\_theoretic\_activity \} 
\texttt{> normalizer\_of\_cyclic\_subgroup "31" \} 
\texttt{> "2,0,0,0,-0,0,1,0,0,0,1,0,3,0,4" \} 
\texttt{end \} 
\texttt{pdf:\_latex -normalizer\_of\_31\_in\_PGL\_4\_5.tex \} 
\texttt{open -normalizer\_of\_31\_in\_PGL\_4\_5.pdf \}
Normalizer of $\mathbb{Z}_2^2$ in $\text{PGL}_2(9)$:

\begin{verbatim}
Normalizer_of_Z22_in_PGL_2_9:
$\text{ORBITER_PATH}$/orbiter.out -v 4
-define G -linear_group -PGL 2 9
-define G -subgroup_by_generators Z22 4 2
"2,0,0,1,-0,1,1,0" -end
-define G -with G -do
-define G -group_theoretic_activity
-define G -normalizer
-define G -end

$\text{ORBITER_PATH}$/orbiter.out -v 4
-define G -linear_group -GL d q wr Sym n 2 2 3
-define G -on_rank_one_tensors -end
-define G -with G -do
-define G -group_theoretic_activity
\end{verbatim}
\#write:GL_2.2_wreath_Sym3_res27_0.layered_graph

2C\_orbit\_under\_PGGL\_4.4\_elements\_coded.csv:
\#write:GL_2.2_wreath_Sym3_res27_0.layered_graph

PGGL\_4.4\_subgroups\_of\_type\_2C\_2C:\2C\_orbit\_under\_PGGL\_4.4\_elements\_coded.csv

1:33 on Mac

User\_time:2:59 on Mac
The distribution of orbit lengths is:

(1, 2, 15, 20, 24 \cdot 3, 30, 40, 240, 256, 480, 512, 960 \cdot 2, 1280, 1920 \cdot 2, 2560 \cdot 4, 3840, 5120, 6144 \cdot 3, 7680, 10240)

#group_theoretic_activity: do orbits on group elements under conjugation after Classes.compute_all_point_orbits

#User-time: 0:57

orbits on conics q13:

$\text{\texttt{\$(ORBITER\_PATH)orbiter.out -v 4\)}}$

$\text{\texttt{-define G -linear_group -PGL 3 13 -end\)}}$

$\text{\texttt{-with G -do\)}}$

$\text{\texttt{-group_theoretic_activity\)}}$

$\text{\texttt{-orbits on polynomials 2\)}}$

$\text{\texttt{-end\)}}$

$\text{\texttt{pdflatex poly_orbits_d2_n2_q13.tex\)}}$

$\text{\texttt{open-poly_orbits_d2_n2_q13.pdf\)}}$

orbits cubic curves q2:

$\text{\texttt{\$(ORBITER\_PATH)orbiter.out -v 4\)}}$

$\text{\texttt{-define G -linear_group -PGL 3 2 -end\)}}$

$\text{\texttt{-with G -do\)}}$

$\text{\texttt{-group_theoretic_activity\)}}$

$\text{\texttt{-orbits on polynomials 3\)}}$

$\text{\texttt{-end\)}}$

$\text{\texttt{pdflatex poly_orbits_d3_n3_q2.tex\)}}$
orbits_cubic_curves_q2_with_draw_tree:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
-draw_options=-yout 500000 -radius 15 -nodes_empty
```

```bash
-line_width 0.5 -y_stretch 0.25 -embedded -end
```

```bash
-define G -linear_group -PGL 3 2 -end
```

```bash
-with G -do
```

```bash
-group_theoretic_activity
```

```bash
-orbits_on_polynomials 3
```

```bash
-orbits_on_polynomials_draw_tree 6
```

```bash
-end
```

poly_orbits_d3_n3_q2.csv:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
-draw_options=-yout 500000 -radius 15 -nodes_empty
```

```bash
-line_width 0.5 -y_stretch 0.25 -embedded -end
```

```bash
-define G -linear_group -PGL 4 2 -end
```

```bash
-with G -do
```

```bash
-group_theoretic_activity
```

```bash
-orbits_on_polynomials 3
```

```bash
-orbits_on_polynomials_draw_tree 6
```

```bash
-end
```

poly_orbits_d3_n3_q2_get_ranks:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
-draw_options=-yout 500000 -radius 15 -nodes_empty
```

```bash
-line_width 0.5 -y_stretch 0.25 -embedded -end
```

```bash
-define G -linear_group -PGL 4 2 -end
```

```bash
-with G -do
```

```bash
-group_theoretic_activity
```

```bash
-orbits_on_polynomials 3
```

```bash
-orbits_on_polynomials_draw_tree 6
```

```bash
-end
```

T4_orbits:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
-define G
```

```bash
-linear_group -GL d q wr Sym n 2 2 4
```

```bash
-on_tensors -end
```

```bash
-with G -do
```

```bash
-group_theoretic_activity
```

```bash
-report
```

```bash
-end
```

```bash
pdflatex GL 2 2 wreath Sym4 res 65535 orbits.tex
```

```bash
open GL 2 2 wreath Sym4 res 65535 orbits.pdf
```

```bash
#pdflatex GL 2 2 wreath Sym4 report.tex
```
4878 \#open-GL_2_2_wreath_Sym4_report.pdf
4879
4880
4881
4882 T4r1_orbits:
4883 \$\{ORBITER_PATH\}orbiter.out-v.4\$
4884 \$\{ORBITER_PATH\}\$-define-G-linear_group-GL_d_q wr_Sym_n 2 2 4\$
4885 \$\{ORBITER_PATH\}\$-on_rank_one_tensors-end\$
4886 \$\{ORBITER_PATH\}\$-with-G-do\$
4887 \$\{ORBITER_PATH\}\$-group_theoretic_activity\$
4888 \$\{ORBITER_PATH\}\$-orbits_on_points\$
4889 \$\{ORBITER_PATH\}\$-export_trees\$
4890 \$\{ORBITER_PATH\}\$-report-end\$
4891 \$\{ORBITER_PATH\}\$-end\$
4892 \$\{ORBITER_PATH\}\$-pdflatex-GL_2_2_wreath_Sym4_orbits_report.tex
4893 \$\{ORBITER_PATH\}\$-open-GL_2_2_wreath_Sym4_orbits_report.pdf
4894
4895
4896 T4r1_orbits_draw:
4897 \$\{ORBITER_PATH\}orbiter.out-v.3\$
4898 \$\{ORBITER_PATH\}\$-draw_layered_graph\$
4899 \$\{ORBITER_PATH\}\$-GL_2_2_wreath_Sym4_res81_0.layered_graph\$
4900 \$\{ORBITER_PATH\}\$-radius 400-spanning_tree-embedded-end\$
4901 \$\{ORBITER_PATH\}\$-line_width 1.1-x_stretch 2.5-scale 0.15-end\$
4902 \$\{ORBITER_PATH\}\$-end\$
4903 \$\{ORBITER_PATH\}\$-pdflatex-GL_2_2_wreath_Sym3_report.tex
4904 \$\{ORBITER_PATH\}\$-open-GL_2_2_wreath_Sym3_report.pdf
4905 \$\{ORBITER_PATH\}\$-pdflatex-GL_2_2_wreath_Sym4_res81_0.draw.tex
4906 \$\{ORBITER_PATH\}\$-open-GL_2_2_wreath_Sym4_res81_0.draw.pdf
4907
4908
4909 T4r1_orbits_4:
4910 \$\{ORBITER_PATH\}orbiter.out-v.4\$
4911 \$\{ORBITER_PATH\}\$-orbiter_path-\$\{ORBITER_PATH\}\$
4912 \$\{ORBITER_PATH\}\$-define-G-linear_group-GL_d_q wr_Sym_n 2 2 4\$
4913 \$\{ORBITER_PATH\}\$-on_rank_one_tensors-end\$
4914 \$\{ORBITER_PATH\}\$-with-G-do\$
4915 \$\{ORBITER_PATH\}\$-group_theoretic_activity\$
4916 \$\{ORBITER_PATH\}\$-poset_classification_control-problem_label-T4r1-W-end\$
4917 \$\{ORBITER_PATH\}\$-bit_depth 4-draw_options-end-draw_poset-report-end-end\$
4918 \$\{ORBITER_PATH\}\$-end\$
4919 \$\{ORBITER_PATH\}\$-orbits_on_subsets-4-end\$
4920 \$\{ORBITER_PATH\}\$-report-end\$
4921 \$\{ORBITER_PATH\}\$-end\$
4922 \$\{ORBITER_PATH\}\$-pdflatex-T4r1_poset.tex
4923 \$\{ORBITER_PATH\}\$-open-T4r1_poset.pdf
4924 \$\{ORBITER_PATH\}\$-pdflatex-GL_2_2_wreath_Sym4_report.tex

558
PGGL_2.8_on_conic_orbits:
$$(\text{ORBITER \ PATH})\text{orbiter.out}\ -v\cdot4$$
-define-G-
-linear_group-PGGL_2.8-PGL2OnConic-end-
-with-G-do-
group_theoretic_activity-
orbits_on_points-
-report-
end
pdflatex PGGL_2.8_OnConic_2.8_orbits_report.tex
open-PGGL_2.8_OnConic_2.8_orbits_report.pdf

#example from the Fining manual, page 107:

PGGL_7.8_orbits:
$$(\text{ORBITER \ PATH})\text{orbiter.out}\ -v\cdot4$$
-define-G-
-linear_group-PGGL_7.8-end-
-with-G-do-
group_theoretic_activity-
-report-
orbits_on_points-
end

#1.min.31.sec.on Mac

SECTION_POSET_CLASSIFICATION:

poset_of_4subsets:
$$(\text{ORBITER \ PATH})\text{orbiter.out}\ -v\cdot3$$
-orbiter_path$$(\text{ORBITER \ PATH)}\cdot$
-linear_group-PGL_2.3-identity_group-end-
-with-G-do-
group_theoretic_activity-
-report_classification_control-
\[V_{3,2}_{\text{trivial}}:\]
\[\text{V}_3 \text{ trivial:} \]
\[\text{pdflatex poset}_5 \text{poset}_l_1 \text{_5}_{\text{draw}}.\text{tex}\]
\[\text{open-poset}_5 \text{poset}_l_1 \text{_5}_{\text{draw}}.\text{pdf}\]
\[\text{V}_4 \text{ trivial:} \]
\[\text{pdflatex poset}_5 \text{poset}_l_1 \text{_5}_{\text{draw}}.\text{tex}\]
\[\text{open-poset}_5 \text{poset}_l_1 \text{_5}_{\text{draw}}.\text{pdf}\]
SECTION_6.3: Orbits on Subsets:

PG_2_2_subsets:

\$\text{(ORBITER\_PATH)orbiter.out\$-v.3}\$

- \text{orbiter_path}\$\text{(ORBITER\_PATH)\$}

- \text{define}\text{F}\$\text{-finite_field}\$-q.2\text{-end}\$

- \text{define}\text{G}\$\text{-linear_group}\$\text{-PGL}\_3\text{-F}\$-\text{-end}\$

- \text{with}\text{G}\$\text{-do}\$

- \text{group}\text{theoretic}\text{activity}\$

- \text{poset}\text{classification}\text{control}\$

- \text{problem}\text{label}\text{PGL}\_3\_2$

- \text{depth}\text{7}\$

- \text{radius}\text{200}\$\text{-embedded}\$

- \text{end}\$

- \text{report}\text{-end}\$

- \text{draw}\text{poset}\$

- \text{end}\$

- \text{orbits}\text{on}\text{subsets}\text{7}\$

- \text{report}\$

- \text{end}\$

\text{pdflatex}\text{PGL}\_3\_2\text{-poset}\_lvl\_7\text{.tex}\$

\text{open}\text{PGL}\_3\_2\text{-poset}\_lvl\_7\text{.pdf}\$
5119 #PG(3,2) has 2^3+2^2+2^1+1 = 15 points:
5120 #PG(3,3) has 3^3+3^2+3^1+1 = 27+9+3+1 = 40 points.
5123
5124 PG_3_2_subsets_again:
5125 \$(\text{ORBITER\ PATH})\text{orbiter.out}-v.3:\$
5126 \quad -\text{orbiter.path}\$(\text{ORBITER\ PATH})\$
5127 \quad -\text{define}\text{-}\text{F}\text{-}\text{finite_field}\text{-}q.2\text{-}end\$
5128 \quad -\text{define}\text{-}\text{G}\text{-}\text{linear_group}\text{-}\text{PGL.4}\text{-}F\text{-}end\$
5129 \quad -\text{with}\text{-}\text{G}\text{-}do\$
5130 \quad -\text{group}\text{-theoretic}\text{activity}\$
5131 \quad -\text{poset}\text{-}\text{classification}\text{}\text{control}\$
5132 \quad -\text{-}\text{problem}\text{-}\text{label}\text{-}\text{PGL.4}\text{-}2\$
5133 \quad -\text{-}\text{depth}\text{-}15\$
5134 \quad -\text{-}\text{draw}\text{-}\text{options}\$
5135 \quad -\text{-}\text{radius}\text{-}200\text{-}embedded\$
5136 \quad -\text{-}\text{end}\$
5137 \quad -\text{-}\text{report}\text{-}end\$
5138 \quad -\text{-}\text{draw}\text{-}\text{poset}\$
5139 \quad -\text{-}\text{end}\$
5140 \quad -\text{-}\text{orbits}\text{-}\text{on}\text{-}\text{subsets}\text{-}15\$
5141 \quad -\text{-}\text{report}\$
5142 \quad -\text{-}\text{end}\$
5143 \quad pdflatex-PGL_4_2_poset_lvl_15.tex
5144 \quad open-PGL_4_2_poset_lvl_15.pdf
5145 \quad pdflatex-PGL_4_2_poset.tex
5146 \quad pdflatex-PGL_4_2_poset.tex
5147 \quad open-PGL_4_2_poset.pdf
5148 \quad #pdflatex-PGL_4_2_poset_detailed_lvl_15.tex
5149 \quad #open-PGL_4_2_poset_detailed_lvl_15.pdf
5150
5153 PG_3_2_subsets:
5154 \$(\text{ORBITER\ PATH})\text{orbiter.out}-v.3:\$
5155 \quad -\text{orbiter.path}\$(\text{ORBITER\ PATH})\$
5156 \quad -\text{define}\text{-}\text{F}\text{-}\text{finite_field}\text{-}q.2\text{-}end\$
5157 \quad -\text{define}\text{-}\text{G}\text{-}\text{linear_group}\text{-}\text{PGL.4}\text{-}F\text{-}end\$
5158 \quad -\text{with}\text{-}\text{G}\text{-}do\$
5159 \quad -\text{group}\text{-theoretic}\text{activity}\$

563
PGL\textsubscript{3,2} singer:

```
\$\texttt{ORBITER\ PATH}\texttt{orbiter.out}\texttt{-v.3}\$
```

\texttt{-define G-linear\ group-PGL\textsubscript{3,2}-singer\ 1-end}\n
\texttt{-with G-do}\n
\texttt{-group\ theoretic\ activity}\n
\texttt{-poset\ classification\ control}\n
\texttt{-problem\ label-PGL\textsubscript{3,2}\ singer\ 1-W-depth\ 7}\n
\texttt{-draw\ poset}\n
\texttt{-report\ -end}\n
\texttt{-orbits\ on\ subsets\ 7}\n
```
pdflatex PGL\textsubscript{3,2} singer\_1_poset.tex
open PGL\textsubscript{3,2} singer\_1 poset.pdf
```

PGL\textsubscript{3,2} on\_lines:

```
\$\texttt{ORBITER\ PATH}\texttt{orbiter.out}\texttt{-v.3}\$
```

\texttt{-define G-linear\ group-PGL\textsubscript{3,2}-end}\n
\texttt{-define G on\_lines-modified\ group-from G}\n
\texttt{-on k\ subspaces\ 2}\n
\texttt{-end}\n
\texttt{-with G on\_lines-do}\n
\texttt{-group\ theoretic\ activity}\n
\texttt{-poset\ classification\ control}\n
\texttt{-problem\ label-PGL\textsubscript{3,2} lines-W-depth\ 7}\n
\texttt{-draw\ poset}\n
\texttt{-report\ -end}\n
\texttt{-end}\n
\texttt{-orbits\ on\ subsets\ 7}\n
```
pdflatex PGL\textsubscript{3,2} lines\_1_poset.tex
open PGL\textsubscript{3,2} lines\_1 poset.pdf
```

564
PGL_2.5_on_subsets:

- report
- end

- pdflatex-PGL_3.2_lines_poset.tex
- open-PGL_3.2_lines_poset.pdf

PGL_2.7_on_subsets:

- report
- end

- pdflatex-PGL_3.2_lines_poset.tex
- open-PGL_3.2_lines_poset.pdf

PGGL_2.8_on_subsets:

- report
- end

- pdflatex-PGL_3.2_lines_poset.tex
- open-PGL_3.2_lines_poset.pdf
5254 \> \> -with\:G\:-do\:\\ 
5255 \> \> -group_theoretic_activity\:\\ 
5256 \> \> \> -poset_classification_control\:\\ 
5257 \> \> \> \> -problem_label\:PGGL_2.8\:W\:-depth\:9\:\\ 
5258 \> \> \> \> -draw_poset\:\\ 
5259 \> \> \> \> -draw_options\:-radius\:200\:-end\:\\ 
5260 \> \> \> \> -report\:-end\:\\ 
5261 \> \> \> -end\:\\ 
5262 \> \> \> -orbits_on_subsets\:9\:\\ 
5263 \> \> \> -report\:\\ 
5264 \> \> \> -end 
5265 \> \> \> pdflatex\:PGGL_2.8_poset.tex 
5266 \> \> \> open-PGGL_2.8_poset.pdf 
5267 
5268  
5269 \> \> PGGL_2.9_on_subsets:\n5270 \> \> \> $(\text{ORBITER\:PATH})\text{orbiter.out\:-v\:10}\:\\ 
5271 \> \> \> \> -orbiter_path\:$\:\text{ORBITER\:PATH}$\:\\ 
5272 \> \> \> \> -define\:G\:-linear_group\:-PGGL_2.9\:-end\:\\ 
5273 \> \> \> \> -with\:G\:-do\:\\ 
5274 \> \> \> \> -group_theoretic_activity\:\\ 
5275 \> \> \> \> \> -poset_classification_control\:\\ 
5276 \> \> \> \> \> \> -problem_label\:PGGL_2.9\:W\:-depth\:10\:\\ 
5277 \> \> \> \> \> \> -draw_poset\:\\ 
5278 \> \> \> \> \> \> -draw_options\:-radius\:200\:-end\:\\ 
5279 \> \> \> \> \> \> -report\:-end\:\\ 
5280 \> \> \> \> \> \> -end\:\\ 
5281 \> \> \> \> -orbits_on_subsets\:10\:\\ 
5282 \> \> \> \> -report\:\\ 
5283 \> \> \> \> -end 
5284 \> \> \> pdflatex\:PGGL_2.9_poset.tex 
5285 \> \> \> open-PGGL_2.9_poset.pdf 
5286 
5287  
5288 \> \> PGL_2.11_on_subsets:\n5289 \> \> \> $(\text{ORBITER\:PATH})\text{orbiter.out\:-v\:10}\:\\ 
5290 \> \> \> \> -orbiter_path\:$\:\text{ORBITER\:PATH}$\:\\ 
5291 \> \> \> \> -define\:G\:-linear_group\:-PGL_2.11\:-end\:\\ 
5292 \> \> \> \> -with\:G\:-do\:\\ 
5293 \> \> \> \> -group_theoretic_activity\:\\ 
5294 \> \> \> \> \> -poset_classification_control\:\\ 
5295 \> \> \> \> \> \> -problem_label\:PGL_2.11\:W\:-depth\:12\:\\ 
5296 \> \> \> \> \> \> -draw_poset\:\\ 
5297 \> \> \> \> \> \> -draw_options\:-radius\:200\:-end\:\\ 
5298 \> \> \> \> \> \> -report\:-end\:\\ 
5299 \> \> \> \> \> \> -end\:\\ 
5300 \> \> \> \> \> \> -orbits_on_subsets\:12\:\

566
5301  \> \> \> -report\ \ \\
5302  \> \> -end\ \\
5303  \> pdflatex PGL_2_11_poset.tex\ \\
5304  \> open PGL_2_11_poset.pdf\ \\
5305\ \\
5306\ \\
5307\ \\
5308  PGGL_2_16_on_subsets:\ \\
5309  \> $(ORBITER\_PATH)\texttt{orbiter.out-v.3}\ \ \\
5310  \> \> -orbiter_path$(ORBITER\_PATH)\ \ \\
5311  \> \> -define-G-linear_group-PGGL_2_16-end\ \\
5312  \> \> -with-G-do\ \\
5313  \> \> -group_theoretic_activity\ \\
5314  \> \> \> -poset_classification_control\ \\
5315  \> \> \> \> -problem_label-PGGL_2_16-W-depth_10\ \\
5316  \> \> \> \> -draw_poset\ \\
5317  \> \> \> \> -report-end\ \\
5318  \> \> \> -end\ \\
5319  \> \> \> -orbits_on_subsets_10\ \\
5320  \> \> \> -report\ \\
5321  \> \> -end\ \\
5322  \> pdflatex PGL_2_16_poset.tex\ \\
5323  \> open PGL_2_16_poset.pdf\ \\
5324\ \\
5325\ \\
5326  PGGL_2_32_on_subsets:\ \\
5327  \> $(ORBITER\_PATH)\texttt{orbiter.out-v.3}\ \ \\
5328  \> \> -orbiter_path$(ORBITER\_PATH)\ \ \\
5329  \> \> -define-G-linear_group-PGGL_2_32-end\ \\
5330  \> \> -with-G-do\ \\
5331  \> \> -group_theoretic_activity\ \\
5332  \> \> \> -poset_classification_control\ \\
5333  \> \> \> \> -problem_label-PGGL_2_32-W-depth_8\ \\
5334  \> \> \> \> -draw_poset\ \\
5335  \> \> \> \> -report-end\ \\
5336  \> \> \> -end\ \\
5337  \> \> \> -orbits_on_subsets_8\ \\
5338  \> \> \> -report\ \\
5339  \> \> -end\ \\
5340  \> pdflatex PGL_2_32_poset.tex\ \\
5341  \> open PGL_2_32_poset.pdf\ \\
5342\ \\
5343\ \\
5344  PG_3_4_subsets:\ \\
5345  \> $(ORBITER\_PATH)\texttt{orbiter.out-v.3}\ \ \\
5346  \> \> -orbiter_path$(ORBITER\_PATH)\ \ \\
5347  \> \> -define-G-linear_group-PGGL_4_4-end\ \\

567
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>5348</td>
<td>\echo -with:\G:do:\</td>
</tr>
<tr>
<td>5349</td>
<td>\echo -group_theoretic_activity:\</td>
</tr>
<tr>
<td>5350</td>
<td>\echo \poset_classification_control:\</td>
</tr>
<tr>
<td>5351</td>
<td>\echo \problem_label:PGGL_4:4\</td>
</tr>
<tr>
<td>5352</td>
<td>\echo \depth:5\</td>
</tr>
<tr>
<td>5353</td>
<td>\echo \draw_poset:\</td>
</tr>
<tr>
<td>5354</td>
<td>\echo \draw_options:\</td>
</tr>
<tr>
<td>5355</td>
<td>\echo \radius:200\</td>
</tr>
<tr>
<td>5356</td>
<td>\echo \end\</td>
</tr>
<tr>
<td>5357</td>
<td>\echo \report:--end\</td>
</tr>
<tr>
<td>5358</td>
<td>\echo \end\</td>
</tr>
<tr>
<td>5359</td>
<td>\echo \orbits_on_subsets:5\</td>
</tr>
<tr>
<td>5360</td>
<td>\echo \report\</td>
</tr>
<tr>
<td>5361</td>
<td>\echo \end</td>
</tr>
<tr>
<td>5362</td>
<td>pdflatex PGGL_4:4_poset.tex</td>
</tr>
<tr>
<td>5363</td>
<td>open-PGGL_4:4_poset.pdf</td>
</tr>
<tr>
<td>5364</td>
<td></td>
</tr>
<tr>
<td>5365</td>
<td></td>
</tr>
<tr>
<td>5366</td>
<td>PGGL_2:9_orbits:</td>
</tr>
<tr>
<td>5367</td>
<td>$(\ORBITER_PATH)orbiter.out--v:3\</td>
</tr>
<tr>
<td>5368</td>
<td>\echo \orbiter_path:$(\ORBITER_PATH)\</td>
</tr>
<tr>
<td>5369</td>
<td>\echo \define:\G:linear_group:PGGL_2:9:--end\</td>
</tr>
<tr>
<td>5370</td>
<td>\echo \with:\G:do:\</td>
</tr>
<tr>
<td>5371</td>
<td>\echo \group_theoretic_activity:\</td>
</tr>
<tr>
<td>5372</td>
<td>\echo \poset_classification_control:\</td>
</tr>
<tr>
<td>5373</td>
<td>\echo \problem_label:PGGL_2:9:--W:--depth:5\</td>
</tr>
<tr>
<td>5374</td>
<td>\echo \report:--end\</td>
</tr>
<tr>
<td>5375</td>
<td>\echo \draw_poset:\</td>
</tr>
<tr>
<td>5376</td>
<td>\echo \draw_options:--radius:200:--end\</td>
</tr>
<tr>
<td>5377</td>
<td>\echo \end\</td>
</tr>
<tr>
<td>5378</td>
<td>\echo \orbits_on_subsets:5\</td>
</tr>
<tr>
<td>5379</td>
<td>\echo \report:</td>
</tr>
<tr>
<td>5380</td>
<td>\echo \end</td>
</tr>
<tr>
<td>5381</td>
<td>pdflatex PGGL_2:9_poset.tex</td>
</tr>
<tr>
<td>5382</td>
<td>open-PGGL_2:9_poset.pdf</td>
</tr>
<tr>
<td>5383</td>
<td></td>
</tr>
<tr>
<td>5384</td>
<td></td>
</tr>
<tr>
<td>5385</td>
<td></td>
</tr>
<tr>
<td>5386</td>
<td></td>
</tr>
<tr>
<td>5387</td>
<td>PGO_5:2_on_subsets:</td>
</tr>
<tr>
<td>5388</td>
<td>$(\ORBITER_PATH)orbiter.out--v:3\</td>
</tr>
<tr>
<td>5389</td>
<td>\echo \orbiter_path:$(\ORBITER_PATH)\</td>
</tr>
<tr>
<td>5390</td>
<td>\echo \define:\F:finite_field:q:2:--end\</td>
</tr>
<tr>
<td>5391</td>
<td>\echo \define:\G:linear_group:PGO_5:F:--end\</td>
</tr>
<tr>
<td>5392</td>
<td>\echo \with:\G:do:\</td>
</tr>
<tr>
<td>5393</td>
<td>\echo \group_theoretic_activity:\</td>
</tr>
<tr>
<td>5394</td>
<td>\echo \poset_classification_control:\</td>
</tr>
</tbody>
</table>
Section 6.4: Orbits on Subspaces:

subspaces.Op_4_2:

$(ORBITER_PATH)orbiter.out-v.5\
-orbiter_path:$(ORBITER_PATH)\
-definition-G-linear_group-PGL.4:2-orthogonal.1-end\n
-node_label_is_element\n
-draw_poset-draw_options-radius:200-end\n
-depth:4\n
-report-end\n
-endsubset\n
-endsubspaces

PGL.4:2 on subspaces:

$(ORBITER_PATH)orbiter.out-v.5\
-orbiter_path:$(ORBITER_PATH)\
-definition-G-linear_group-PGL.4:2-end\n
-with-G-do\n
-group_theoretic_activity
5442 \triangleright \triangleright \triangleright -\text{poset\_classification\_control}\backslash
5443 \triangleright \triangleright \triangleright -\text{problem\_label}PGL\_4\_2-P\_W-depth\_4\backslash
5444 \triangleright \triangleright \triangleright -\text{report}\backslash
5445 \triangleright \triangleright -\text{end}\backslash
5446 \triangleright \triangleright -\text{orbits\_on\_subspaces}\_4\backslash
5447 \triangleright \triangleright -\text{report}\backslash
5448 \triangleright -\text{end}\backslash
5449 -pdflatex\_PGL\_4\_2\_report.tex
5450 -open\_PGL\_4\_2\_report.pdf
5451 -pdflatex\_PGL\_4\_2\_poset.tex
5452 -open\_PGL\_4\_2\_poset.pdf
5453
5454
5455 \triangleright
5456 \text{PGL\_4\_2\_singer\_on\_subspaces}:\backslash
5457 \triangleright \$(\text{ORBITER\_PATH})\text{orbiter\_out}\_v\_5\backslash
5458 \triangleright -\text{orbiter\_path}$(\text{ORBITER\_PATH})\backslash
5459 \triangleright -\text{define}G-\text{linear\_group}PGL\_4\_2\_singer\_1\_\text{end}\backslash
5460 \triangleright -\text{with}G\_\text{do}\backslash
5461 \triangleright -\text{group\_theoretic\_activity}\backslash
5462 \triangleright \triangleright -\text{poset\_classification\_control}\backslash
5463 \triangleright \triangleright \triangleright -\text{node\_label\_is\_element}\backslash
5464 \triangleright \triangleright \triangleright -\text{draw\_poset}\backslash
5465 \triangleright \triangleright \triangleright -\text{draw\_options}\_\text{end}\backslash
5466 \triangleright \triangleright \triangleright -\text{problem\_label}PGL\_4\_2\_singer\_\_W\_\text{depth}\_4\backslash
5467 \triangleright \triangleright \triangleright -\text{report}\_\text{end}\backslash
5468 \triangleright \triangleright -\text{end}\backslash
5469 \triangleright \triangleright -\text{orbits\_on\_subspaces}\_4\backslash
5470 \triangleright \triangleright -\text{report}\backslash
5471 \triangleright -\text{end}\backslash
5472 -pdflatex\_PGL\_4\_2\_Singer\_4\_2\_1\_poset.tex
5473 -open\_PGL\_4\_2\_Singer\_4\_2\_1\_poset.pdf
5474
5475
5476
5477 \text{PGL\_8\_2\_singer\_on\_subspaces}:\backslash
5478 \triangleright \$(\text{ORBITER\_PATH})\text{orbiter\_out}\_v\_5\backslash
5479 \triangleright -\text{orbiter\_path}$(\text{ORBITER\_PATH})\backslash
5480 \triangleright -\text{define}G-\text{linear\_group}PGL\_8\_2\_singer\_1\_\text{end}\backslash
5481 \triangleright -\text{with}G\_\text{do}\backslash
5482 \triangleright -\text{group\_theoretic\_activity}\backslash
5483 \triangleright \triangleright -\text{poset\_classification\_control}\backslash
5484 \triangleright \triangleright \triangleright -\text{node\_label\_is\_element}\backslash
5485 \triangleright \triangleright \triangleright -\text{draw\_poset}\backslash
5486 \triangleright \triangleright \triangleright -\text{draw\_options}\_\text{radius}\_150\_\text{end}\backslash
5487 \triangleright \triangleright \triangleright -\text{problem\_label}PGL\_8\_2\_singer\_\_W\_\text{depth}\_8\backslash
5488 \triangleright \triangleright \triangleright -\_\text{report}\_\text{end}\backslash
Orbits on subspaces:

- orbits_on_subspaces-8-
- report-
- end

pdflatex PGL_8_2_Singer_8_2_1_poset.tex
open PGL_8_2_Singer_8_2_1_poset.pdf

# May 7, 2020: 16:sec on Mac
# 1643 orbits in total

Op_6_2_orbits_on_subspaces:

$(ORBITER\_PATH)orbiter.out\ -v\ -5$
-orbiter_path$(ORBITER\_PATH)$
-define G-linear_group-PGL_6_2-orthogonal_1-end
-with G-do
-group_theoretic_activity

- node_label_is_element
- draw_poset
-draw_options-radius 200-end
-problem_label Op_6_2-W
-depth 6-report-end
-end
-orbits_on_subspaces-6-
-report-
-end

pdflatex PGL_6_2_Orthogonal plus_6_2_report.tex
open PGL_6_2_Orthogonal plus_6_2_report.pdf

Op_6_3_orbits_on_subspaces:

$(ORBITER\_PATH)orbiter.out\ -v\ -5$
-orbiter_path$(ORBITER\_PATH)$
-define G-linear_group-PGL_6_3-orthogonal_1-end
-with G-do
-group_theoretic_activity

- node_label_is_element
- draw_poset
-draw_options-radius 200-end
-problem_label Op_6_3-W
-depth 6-report-end
-end
-orbits_on_subspaces-6-
Op_6_11_orbits_on_subspaces:

Op_8_2_orbits_on_subspaces:

# June 3, 2020 on Mac: 12 sec
Section 6.5: Arcs and Caps in Projective Spaces

SECTION ARCS AND CAPS IN PROJECTIVE SPACES:

PGL_3.27:

```bash
$ (ORBITER_PATH) orbiter.out --v 5
```

```bash
define G
```

```bash
define linear group -PGL_3.27 -end
```

```bash
with G do
```

```bash
define F finite field -q 2 -end
```
AGGL_2.27:

$ $(\text{ORBITER\_PATH}) \text{orbiter.out}\cdot -v.5: \$
\text{define-G:}
\text{linear\_group}\cdot \text{AGGL}\cdot 2.27: \text{end:}
\text{with-G:do:}
\text{group\_theoretic\_activity:}
\text{report:}
\text{end}

pdflatex \text{AGGL}_2.27\_report.tex
open \text{AGGL}_2.27\_report.pdf

hyperoval_4\_classify:

$ $(\text{ORBITER\_PATH}) \text{orbiter.out}\cdot -v.4: \$
\text{define-F:finite\_field}\cdot q.4: \text{end:}
\text{define-P:projective\_space}\cdot 2.F: \text{end:}
\text{with-P:do:}
\text{projective\_space\_activity:}
\text{classify\_arcs:}
\text{poset\_classification\_control:}
\text{problem\_label-hyperoval}\cdot q.4: \text{end:}
\text{w-depth}\cdot 6: \text{end:}
\text{target\_size}\cdot 6: \text{end:}
\text{d}\cdot 2: \text{end}

pdflatex hyperoval\_q4\_poset.tex
open hyperoval\_q4\_poset.pdf

hyperoval_8\_classify:

$ $(\text{ORBITER\_PATH}) \text{orbiter.out}\cdot -v.4: \$
\text{orbiter\_path}\cdot $(\text{ORBITER\_PATH})\$
\text{define-F:finite\_field}\cdot q.8: \text{end:}
\text{define-P:projective\_space}\cdot 2.F: \text{end:}
\text{with-P:do:}
\text{projective\_space\_activity:}
\text{classify\_arcs:}
\begin{verbatim}
5677 ▶ ▶ ▶ ▶ -poset_classification_control:\n5678 ▶ ▶ ▶ ▶ -problem_label-hyperoval_q8:\n5679 ▶ ▶ ▶ ▶ -W-depth:10:\n5680 ▶ ▶ ▶ ▶ -report:-end:\n5681 ▶ ▶ ▶ ▶ -draw_poset:\n5682 ▶ ▶ ▶ ▶ -draw_options:\n5683 ▶ ▶ ▶ ▶ -radius:200:\n5684 ▶ ▶ ▶ ▶ -end:\n5685 ▶ ▶ ▶ -end\n5686 ▶ ▶ ▶ -target_size:10:\n5687 ▶ ▶ ▶ -d:2:\n5688 ▶ ▶ ▶ -end\n5689 ▶ ▶ -end
5690 ▶ pdflatex hyperoval_q8_poset.tex
5691 ▶ open hyperoval_q8_poset.pdf
5692
5693
5694
5695
5696 frame_stabilizer_PGGL:
5697 ▶ $(ORBITER_PATH)orbiter.out-v.4\n5698 ▶ ▶ -define-G\n5699 ▶ ▶ -linear_group-PGL.3.8\n5700 ▶ ▶ -depth:4\n5701 ▶ ▶ -group_theoretic_activity\n5702 ▶ ▶ ▶ -poset_classification_control\n5703 ▶ ▶ ▶ -problem_label-frame_q8-W-depth:4\n5704 ▶ ▶ ▶ -draw_options\n5705 ▶ ▶ ▶ -report\n5706 ▶ ▶ ▶ -end\n5707 ▶ ▶ ▶ -classify_arcs\n5708 ▶ ▶ ▶ -target_size:4\n5709 ▶ ▶ ▶ -q:8\n5710 ▶ ▶ ▶ -n:3\n5711 ▶ ▶ ▶ -d:2\n5712 ▶ ▶ ▶ -end\n5713 ▶ ▶ -end
5714 ▶ pdflatex frame_q8_poset.tex
5715 ▶ open frame_q8_poset.pdf
5716
5717 frame_stabilizer_PGL:
5718 ▶ $(ORBITER_PATH)orbiter.out-v.4\n5719 ▶ ▶ -define-G\n5720 ▶ ▶ -linear_group-PGL.3.8\n5721 ▶ ▶ -with-G-do\n5722 ▶ ▶ -group_theoretic_activity\n5723 ▶ ▶ ▶ -poset_classification_control\n\end{verbatim}

575
hyperoval_16_classify:
$(ORBITER_PATH)orbiter.out.-v.4\$
-orbiter_path$(ORBITER_PATH)\$
-define-F-finite_field-q.16-end\$
-define-P-projective_space.2.F-end\$
-with-P-do\$
-projective_space_activity\$
-classify_arcs\$
-poset_classification_control\$
-problem_label-hyperoval_q16-W-depth.18\$
-report.-end\$
-target_size.18\$
-d.2\$
-end\$
-end\$
-pdflatex-hyperoval_q16_poset.tex
-open-hyperoval_q16_poset.pdf

hyperoval_16_conic.type:
$(ORBITER_PATH)orbiter.out.-v.2\$
-define-F-finite_field-q.16-end\$
-define-P-projective_space.2.F-end\$
-define-H_16_1-geometric_object.P\$
-set.$(HYPEROVAL_16_144)\$
-end\$

pdflatex-frame_q8_poset.tex
-pdflatex-hyperoval_q16_poset.tex
-pdflatex-frame_q8_poset.pdf
open-frame_q8_poset.pdf
open-hyperoval_q16_poset.pdf
5771  \[\text{-with}\ H_{16\cdot1}\ \text{-do}\]
5772  \[\text{-combinatorial\ object\ activity}\]
5773  \[\text{-save}\]
5774  \[\text{-end}\]
5775  \[\text{-with}\ H_{16\cdot1}\ \text{-do}\]
5776  \[\text{-combinatorial\ object\ activity}\]
5777  \[\text{-conic\ type}\cdot6\]
5778  \[\text{-end}\]
5779  \[\text{-print\ symbols}\]
5780
5781  \[\text{hyperoval}_{16\cdot1}\text{nonconical\ type:}\]
5782  \[\text{\$ORBITER\_PATH}\text{orbiter.out}\text{-v.2}\]
5783  \[\text{\$define\ F\_finite\_field\_q16\_end}\]
5784  \[\text{\$define\ P\_projective\_space\_2\_F\_end}\]
5785  \[\text{\$define\ H_{16\cdot1}\_geometric\_object\_P}\]
5786  \[\text{\$set\ \$\\text{\$\text{HYPEROVAL}_{16\cdot144}}\}\]
5787  \[\text{-end}\]
5788  \[\text{-with}\ H_{16\cdot1}\ \text{-do}\]
5789  \[\text{-combinatorial\ object\ activity}\]
5790  \[\text{-save}\]
5791  \[\text{-end}\]
5792  \[\text{-with}\ H_{16\cdot1}\ \text{-do}\]
5793  \[\text{-combinatorial\ object\ activity}\]
5794  \[\text{-non\ conical\ type}\]
5795  \[\text{-end}\]
5796  \[\text{-print\ symbols}\]
5797
5798
5799  \#\text{We\ found\ 17028\ non\-conical\ 6\subsets}
5800  \#\text{Eckardt\-point\-number\-distribution: }\text{\$13^{252}}\text{, }\text{\$9^{720}}\text{, }\text{\$5^{2304}}\text{, }\text{\$3^{13752}}\text{.}
5801
5802
5803  \[\text{hyperoval}_{16\cdot2}\text{nonconical\ type:}\]
5804  \[\text{\$ORBITER\_PATH}\text{orbiter.out\-v.2}\]
5805  \[\text{\$define\ F\_finite\_field\_q16\_end}\]
5806  \[\text{\$define\ P\_projective\_space\_2\_F\_end}\]
5807  \[\text{\$define\ H_{16\cdot2}\_geometric\_object\_P}\]
5808  \[\text{\$set\ \$\\text{\$\text{HYPEROVAL}_{16\cdot16320}}\}\]
5809  \[\text{-end}\]
5810  \[\text{-with}\ H_{16\cdot2}\ \text{-do}\]
5811  \[\text{-combinatorial\ object\ activity}\]
5812  \[\text{-save}\]
5813  \[\text{-end}\]
5814  \[\text{-with}\ H_{16\cdot2}\ \text{-do}\]
5815  \[\text{-combinatorial\ object\ activity}\]
5816  \[\text{-non\ conical\ type}\]
We found $\binom{17}{5}$ non-conical 6-subsets.

Eckardt point number distribution: $45^{68}, 13^{2040}, 5^{4080}$

#neighbors_of_0_with_4_removed.csv

<table>
<thead>
<tr>
<th>Row</th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>12</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>11</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>10</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>9</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>8</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>9</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>8</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>6</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>12</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>11</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

#END

hyperoval_16_stab_0_disjoint_sets_graph:

```
$ (ORBITER_PATH) orbiter.out -v 2
```

```
$ -define=G -graph -disjoint_sets_graph
```

```
$ -neighbors_of_0_with_4_removed.csv
```

```
$ -end
```

```
$ -with=G -do
```

```
$ -graph_theoretic_activity
```

```
$ -find_cliques
```

```
$ -target_size=4
```

```
$ -end
```

```
$ -end
```

```
$ -print_symbols
```

#5_cliques_of_size_4
#clique

0:

0, 2, 3, 9, 10

6, 7, 11, 13, 17

15, 5, 8, 15, 16

16, 1, 6, 12, 14

partition: (1, 6, 12, 14 | 2, 3, 9, 10 | 5, 8, 15, 16 | 7, 11, 13, 17)

4 is missing, it is the nucleus

0 is missing is the chosen point

nonconical 6-arcs are used for classifying cubic surfaces:

nc_arcs_16:

$\$(ORBITER_PATH)oriter.out-v.4\$

-define F: finite_field-q.16-end

-define P: projective_space-2.F-end

-with P:do

-projective_space_activity

-classify_arcs

-poset_classification_control

-problem_label nc_arcs_q16_d2

-W: depth 6

-report: end

-end

-target_size 6

-d.2

-conic_test

-end

-pdflatex nc_arcs_q16_d2_poset.tex
nc_arcs_32_E13:

$\$(ORBITER\_PATH)\$\$orbiter.out -v 4 \$

-define-F\-finite_field\-q:32\-end\$

-define-P\-finite_field\-q:32\-end\$

-define-P\-projective_space\-2\-F\-end\$

-define-P\-projective_space\-activity\$

-classify_arcs\$

-poset_classification_control\$

-problem_label\nc_arcs_32\_d2\$

-depth-6\$

-draw_poset\-draw_options\-end\$

-report\-end\$

-end\$

-target_size-6\$

-test_nb_Eckardt_points\-13\$

-d-2\$

-conic_test\$

-end\$

-end

pdflatex nc_arcs_q32\_d2\_poset.tex

open nc_arcs_q32\_d2\_poset.pdf

F64\_work:

$\$(ORBITER\_PATH)\$\$orbiter.out -v 3 \$

-define-F\-finite_field\-q:64\-end\$

-define-f\-formula\"f"\"f"\"a*a+a\"\$

-with-F\-do\-finite_field\-activity\$

-evaluate-f\"a=2\"\-end

F64\_frob:

$\$(ORBITER\_PATH)\$\$orbiter.out -v 3 \$

-define-F\-finite_field\-q:64\-end\$

-define-f\-formula\"f"\"f"\"a*a*a*a*a*a*a\"\$

-with-F\-do\-finite_field\-activity\$
5958  ▷  ▷  ▷  -evaluate:"a=61"-end
5959
5960
5961 #surfaces-with-13-Eckardt-points-have:O CN=0,98,99
5962
5963 surface_64_0:
5977  ▷  pdflatex:surface_catalogue_q64_iso0_with_group.tex
5978  ▷  open:surface_catalogue_q64_iso0_with_group.pdf
5979
5980
5981
5982
5983 #makes-it-slow:
5984  ▷  ▷  ▷  ▷  -test_nb_Eckardt_points:13:\n5985  ▷  ▷  ▷  ▷  ▷  -report:-select_orbits_by_level:6:-select_orbits_by_stabilizer_order_m
5986  ▷  ▷  ▷  ▷  ▷  ▷  multiple_of:24:-end:\n5987  ▷  ▷  ▷  -User_time:0:3
5988
5989
5990 nc_arcs_128:
581
nc_arcs_256_E13:
$$(ORBITER\_PATH)orbiter.out-v.8$$
-define_F-finite_field-q.256-end
-define_P-projective_space-2_F-use_projectivity_subgroup-end
-with_P-do
-projective_space_activity
-classify_arcs
-poset_classification_control
-problem_label-nc_arcs_q256_d2-\-
-target_size-6
-test_nb_Eckardt_points.13
-d.2
-end

Example_F64:
$$(ORBITER\_PATH)orbiter.out-v.3$$
-define_F-finite_field-q.64-end
-define_P-projective_space-3_F-end
-with_P-do
-projective_space_activity
-define_surface-S64_abcd_52.8.52-q.64
6051 ▶ ▶ ▶ ▶ -family_general_abcd_52_8_8_52--end\r
6052 ▶ ▶ -end\r
6053 ▶ ▶ -with_S64_abcd_52_8_8_52--do\r
6054 ▶ ▶ ▶ -cubic_surface_activity\r
6055 ▶ ▶ ▶ -report\r
6056 ▶ ▶ -end\r
6057 ▶ pdflatex:surface_family_general_abcd_q64_a52_b8_c8_d52_report.tex
6058
6059
6060
6061
6062 six_arcs_4_nbE13:
6063 ▶ $(ORBITER_PATH)orbiter.out--v.3\r
6064 ▶ ▶ -define_F--finite_field--q4--end\r
6065 ▶ ▶ -define_P--projective_space--2_F--end\r
6066 ▶ ▶ -with_P--do\r
6067 ▶ ▶ -projective_space_activity\r
6068 ▶ ▶ -control_six_arcs--problem_label--sixarcs_q4--end\r
6069 ▶ ▶ -six_arcs_not_on_conic--filter_by_nb_Eckardt_points_13--end
6070
6071
6072 six_arcs_8_nbE13:
6073 ▶ $(ORBITER_PATH)orbiter.out--v.3\r
6074 ▶ ▶ -define_F--finite_field--q8--end\r
6075 ▶ ▶ -define_P--projective_space--2_F--end\r
6076 ▶ ▶ -with_P--do\r
6077 ▶ ▶ -projective_space_activity\r
6078 ▶ ▶ -control_six_arcs--problem_label--sixarcs_q8--end\r
6079 ▶ ▶ -six_arcs_not_on_conic--filter_by_nb_Eckardt_points_13--end
6080
6081 six_arcs_16_nbE13:
6082 ▶ $(ORBITER_PATH)orbiter.out--v.3\r
6083 ▶ ▶ -define_F--finite_field--q16--end\r
6084 ▶ ▶ -define_P--projective_space--2_F--end\r
6085 ▶ ▶ -with_P--do\r
6086 ▶ ▶ -projective_space_activity\r
6087 ▶ ▶ -control_six_arcs--problem_label--sixarcs_q16--end\r
6088 ▶ ▶ -six_arcs_not_on_conic--filter_by_nb_Eckardt_points_13--end
6089
6090 six_arcs_32_nbE13:
6091 ▶ $(ORBITER_PATH)orbiter.out--v.3\r
6092 ▶ ▶ -define_F--finite_field--q32--end\r
6093 ▶ ▶ -define_P--projective_space--2_F--end\r
6094 ▶ ▶ -with_P--do\r
6095 ▶ ▶ -projective_space_activity\r
6096 ▶ ▶ -control_six_arcs--problem_label--sixarcs_q32--end\r
6097 ▶ ▶ -six_arcs_not_on_conic--filter_by_nb_Eckardt_points_13--end

583
six_arcs_64_nbE13:
$\texttt{ORBITER\_PATH}/\texttt{orbiter.out} -v.3\$
\texttt{define-F\_finite_field-q.64\_end}\$
\texttt{define-P\_projective_space-2\_F\_end}\$
\texttt{with-P\_do}\$
\texttt{projective_space_activity}\$
\texttt{control_six_arcs\_problem_label\_sixarcs_q64\_end}\$
\texttt{six_arcs_not_on_conic\_filter_by_nb_Eckardt_points.13\_end}\$

User-time: 0:7
9-arcs: ago: 4,8,24^5,48^2

six_arcs_128_nbE13:
$\texttt{ORBITER\_PATH}/\texttt{orbiter.out} -v.3\$
\texttt{define-F\_finite_field-q.128\_end}\$
\texttt{define-P\_projective_space-2\_F\_end}\$
\texttt{with-P\_do}\$
\texttt{projective_space_activity}\$
\texttt{control_six_arcs\_problem_label\_sixarcs_q128\_end}\$
\texttt{six_arcs_not_on_conic\_filter_by_nb_Eckardt_points.13\_end}\$

1-min:39-sec
12-arcs, ago: 4^3,24^9

six_arcs_256_nbE13:
$\texttt{ORBITER\_PATH}/\texttt{orbiter.out} -v.3\$
\texttt{define-F\_finite_field-q.256\_end}\$
\texttt{define-P\_projective_space-2\_F\_end}\$
\texttt{with-P\_do}\$
\texttt{projective_space_activity}\$
\texttt{control_six_arcs\_problem_label\_sixarcs_q256\_end}\$
\texttt{six_arcs_not_on_conic\_filter_by_nb_Eckardt_points.13\_end}\$

27-minutes-on-ripoff
User-time: 29:11 on-ripoff 7/30/21

five_arcs_q13:
$\texttt{ORBITER\_PATH}/\texttt{orbiter.out} -v.4\$
\texttt{define-F\_finite_field-q.13\_end}\$
\texttt{define-P\_projective_space-2\_F\_end}\$
\section*{Cubic Curves $\mathbb{P}^2_4$:}

\begin{verbatim}
$\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\time...
#open-Cubic_curves_q4.pdf

cubic_curves_PG_2_4.draw:
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
$\langle ORBITER\_PATH\rangle orbiter.out.-v.3:\$
##### Chapter 7: Cubic Surfaces

##### Section 7.1: Cubic Surfaces Creation

```
SECTION_CUBIC_SURFACES_CREATION:

surface_4_0:

$ (ORBITER_PATH) orbiter.out -v 3:

-def F -finite_field -q 4 -end:

-def P -projective_space 3 F -end:

-with P -do:

-projective_space_activity:

-def_surface S -q 4 -catalogue 0 -end:

-end:

-with S -do:

-cubic_surface_activity:

-report:

-report_with_group:

-end

pdflatex surface_catalogue_q4_iso0_report.tex

open surface_catalogue_q4_iso0_report.pdf

pdflatex surface_catalogue_q4_iso0_with_group.tex

open surface_catalogue_q4_iso0_with_group.pdf
```
Family general F7:

```bash
# Fermat with 18 Eckardt points
# no-automorphism group, so no --report_with_group and no --all_quartic_curves

# Joel:
```
\begin{verbatim}
6333  \> \> -define-F\-finite_field\-q\13\-end\\n6334  \> \> -define-P\-projective_space\-3\-F\-end\\n6335  \> \> -with-P\-do\\n6336  \> \> -projective_space_activity\\n6337  \> \> \> -define_surface_S\_2\_1\-q\13\\n6338  \> \> \> -family_Eckardt\-4\-12\-end\\n6339  \> \> \> -end\\n6340  \> \> \> -with_S\_2\_1\-do\\n6341  \> \> \> -cubic_surface_activity\\n6342  \> \> \> \> -report\\n6343  \> \> \> \> -report_with_group\\n6344  \> \> \> \> -end\n6345
6346
6347
6348
6349
6350
6351
6352  surface_8_0\_catalogue:\n6353  \> $(\text{ORBITER_PATH})\text{orbiter.out\-v\:3}\n6354  \> \> -define-F\-finite_field\-q\8\-end\n6355  \> \> -define-P\-projective_space\-3\-F\-end\n6356  \> \> -with-P\-do\n6357  \> \> -projective_space_activity\n6358  \> \> -define_surface_S\_8\_0\-q\8\-catalogue\-0\-end\n6359  \> \> -end\n6360  \> \> -with_S\_8\_0\-do\n6361  \> \> -cubic_surface_activity\n6362  \> \> \> -report\n6363  \> \> \> -report_with_group\n6364  \> \> \> -end\n6365  \> pdflatex\-surface_catalogue_q8\_iso0\_report.tex
6366  \> open\-surface_catalogue_q8\_iso0\_report.pdf
6367  \> pdflatex\-surface_catalogue_q8\_iso0\_with_group.tex
6368  \> open\-surface_catalogue_q8\_iso0\_with_group.pdf
6369
6370
6371
6372  surface_8_0\_clean:\n6373  \> $(\text{ORBITER_PATH})\text{orbiter.out\-v\:3}\n6374  \> \> -define-F\-finite_field\-q\8\-end\n6375  \> \> -define-P\-projective_space\-3\-F\-end\n6376  \> \> -with-P\-do\n6377  \> \> -projective_space_activity\n6378  \> \> -define_surface_S\_8\_0\-q\8\-catalogue\-0\n6379  \> \> \> -select\_double\_six\"15,11,22,19,24,5,16,10,23,20,25,4\"
\end{verbatim}
-select_double_six:"3,2,1,0,5,4,9,8,7,6,11,10"\ 
-transform_inverse:"1,4,4,0,6,0,0,6,2,0,1,7,0,4,0,0"\ 
-transform:"4,4,0,0,0,0,1,0,0,0,0,0,1,0,0,0"\ 
-transform_inverse:"2,0,0,0,0,2,0,0,0,0,2,0,1,1,2,3,0"\ 
-end\ 
-with-S8_0--do-\ 
cubic_surface_activity\ 
-report\ 
-report_with_group\ 
-end\ 
pdflatex:surface_catalogue_q8_iso0_report.tex\ 
open:surface_catalogue_q8_iso0_report.pdf\ 

#-clean-equation-for-Tekirdag-1:\ 
surface_8_0b:\ 
$(ORBITER_PATH)orbiter.out--v.3\ 
define-F-finite_field-q-8--end\ 
define-P-projective_space-3-F--end\ 
-with-P--do\ 
-projective_space_activity\ 
define_surface_S8_0-q-8-catalogue-0\ 
-select_double_six:"15,11,22,19,24,5,16,10,23,20,25,4"\ 
-select_double_six:"3,2,1,0,5,4,9,8,7,6,11,10"\ 
-transform_inverse:"1,0,0,0,0,1,0,6,0,0,1,6,0,0,0,1,0"\ 
-transform_inverse:"3,1,1,0,0,1,0,0,0,1,0,0,0,0,1,0"\ 
-transform_inverse:"2,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0"\ 
-end\ 

-with-S8_0--do-\ 
cubic_surface_activity\ 
-report\ 
-report_with_group\ 
-end\ 
pdflatex:surface_catalogue_q8_iso0_with_group.tex\ 
open:surface_catalogue_q8_iso0_with_group.pdf\ 

#-writes-tangents.txt\ 

#-13_0-has-4-Eckardt-points\ 
#-13_1-has-6-Eckardt-points\ 
#-13_2-has-9-Eckardt-points
Eckardt_13:

$\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \$

$\$\$>-define F -finite_field -q 13 -end\$

$\$\$>-define P -projective_space 3 F -end\$

$\$\$>-with P -do \$

$\$\$>-define \text{projective\_space}\_S\_q13 -q 13 \$

$\$\$>-define \text{family\_Eckardt}\_3\_1 -end\$

$\$\$>-with S\_q13 -do \$

$\$\$>-define \text{cubic\_surface\_activity}\$

$\$\$>-define \text{report}\$

$\$\$>-define \text{report\_with\_group}\$

$\$\$>-end\$

$\$\$>-with S\_q13 -do \$

$\$\$>-define \text{cubic\_surface\_activity}\$

$\$\$>-define \text{report}\$

$\$\$>-define \text{report\_with\_group}\$

$\$\$>-end\$

$\$\$>pdflatex \text{surface\_family\_Eckardt\_q13\_a3\_b1\_with\_group.tex}$

$\$\$>open \text{surface\_family\_Eckardt\_q13\_a3\_b1\_with\_group.pdf}$

$\$\$>pdflatex \text{surface\_catalogue\_q13\_iso0\_report.tex}$

$\$\$>open \text{surface\_catalogue\_q13\_iso0\_report.pdf}$

$\$\$>$clean\text{ equation}\_for\_Tekirdag-2: \$

$\$\$>surface\_16\_0: \$

$\$\$>$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \$
6474 \triangleright \triangleright -\text{define-F-finite_field-q}\,16\,-\text{end}\backslash
6475 \triangleright \triangleright -\text{define-P-projective_space-3-F-end}\backslash
6476 \triangleright \triangleright -\text{with-P-do}\backslash
6477 \triangleright \triangleright -\text{projective_space_activity}\backslash
6478 \triangleright \triangleright \triangleright -\text{define_surface S}\,16\,0-q\,16\,-\text{catalogue-0}\backslash
6479 \triangleright \triangleright \triangleright -\text{transform-1,0,0,0,1,0,12,0,0,1,2,0,0,0,1,0}\cdot\text{end}\backslash
6480 \triangleright \triangleright \triangleright -\text{transform-15,11,4,0,0,12,0,0,12,0,0,0,0,1,3}\cdot\text{end}\backslash
6481 \triangleright \triangleright \triangleright -\text{end}\backslash
6482 \triangleright \triangleright -\text{end}\backslash
6483 \triangleright \triangleright -\text{with-S}\,16\,0\,-\text{do}\backslash
6484 \triangleright \triangleright -\text{cubic_surface_activity}\backslash
6485 \triangleright \triangleright \triangleright -\text{report}\backslash
6486 \triangleright \triangleright \triangleright -\text{report_with_group}\backslash
6487 \triangleright \triangleright -\text{end}\backslash
6488 \triangleright \text{pdflatex-surface_catalogue_q16_iso0_with_group.tex}\backslash
6489 \triangleright \text{open-surface_catalogue_q16_iso0_with_group.pdf}\backslash
6490
6491
6492 \triangleright \triangleright -\text{transform_inverse-3,0,0,0,0,1,1,0,0,0,1,0,0,0,0,1,0}\cdot\text{end}\backslash
6493 \triangleright \triangleright -\text{transform_inverse-13,12,1,0,12,13,1,0,0,0,1,0,0,0,1,0}\cdot\text{end}\backslash
6494 \triangleright \triangleright -\text{transform_inverse-1,0,0,0,0,1,0,0,12,12,1,0,0,0,1,0}\cdot\text{end}\backslash
6495 \triangleright \triangleright -\text{transform_inverse-12,0,0,0,0,12,0,0,0,0,1,0,0,0,0,1,0}\cdot\text{end}\backslash
6496
6497 \#-\text{rank-of_lines-(-66591,-26737,-4093,-69904,-28376,-26470,-70160,-69855,-26208,5}\backslash
6498 \#-\text{rank-of_lines-(-5847,369,-32230,529,-30293,70068,2178,261,-28666,8575,105,31694,-0,51784,}\backslash
6499 \#-\text{rank-of_lines-(-0,25209,22193,-49862,274)-end}\backslash
6499
6500
6501 \#-\text{ai::29181,4677,29950,33,-62496,-429}\backslash
6502 \#-\text{bi::1,9205,37,29964,29364,21501}\backslash
6503
6504 \#-\text{Tekirdag-1}\backslash
6505
6506 \text{G13}\_8:\backslash
6507 \triangleright \$(\text{ORBITER_PATH})\text{orbiter.out}\,-\text{v-3}\cdot\backslash
6508 \triangleright \triangleright -\text{define-F-finite_field-q}\,8\,-\text{end}\backslash
6509 \triangleright \triangleright -\text{define-P-projective_space-3-F-end}\backslash
6510 \triangleright \triangleright -\text{with-P-do}\backslash
6511 \triangleright \triangleright -\text{projective_space_activity}\backslash
6512 \triangleright \triangleright \triangleright -\text{define_surface T1-family_G13\_2-q}\,8\,-\text{end}\backslash
6513 \triangleright \triangleright \triangleright -\text{end}\backslash
6514 \triangleright \triangleright -\text{with-T1-do}\backslash
6515 \triangleright \triangleright -\text{cubic_surface_activity}\backslash
6516 \triangleright \triangleright \triangleright -\text{report}\backslash

592
Tekirdag-2:

F13_8:

$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot3\$

-define-F\cdot-\text{finite\_field}\cdot-q\cdot8\cdot-\text{end}\$

-define-P\cdot-\text{projective\_space}\cdot3\cdot-F\cdot-\text{end}\$

-define-P\cdot-\text{finite}\cdot\text{field}\cdot-q\cdot8\cdot-\text{end}\$

-report\$

-report\cdot\text{with}\cdot\text{group}\$

-end

Tekirdag-3:

F13_16:

$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot3\$

-define-F\cdot-\text{finite\_field}\cdot-q\cdot16\cdot-\text{end}\$

-define-P\cdot-\text{projective\_space}\cdot3\cdot-F\cdot-\text{end}\$

-define-P\cdot-\text{finite}\cdot\text{field}\cdot-q\cdot16\cdot-\text{end}\$

-report\$

-report\cdot\text{with}\cdot\text{group}\$

-end

F13_32:
6564 \>
$\$(ORBITER\_PATH)\text{orbiter.out}\_v\_3$
6565 \>
\> -define F\_finite_field\_q\_64\_end$
6566 \>
\> -define P\_projective_space\_3\_F\_end$
6567 \>
\> -with P\_do$
6568 \>
\> -projective_space\_activity$
6569 \>
\> \> -define\_surface\_T3\_family\_F13\_2\_q\_32\_end$
6570 \>
\> -end$
6571 \>
\> -with T3\_do$
6572 \>
\> -cubic\_surface\_activity$
6573 \>
\> \> -report$
6574 \>
\> \> \> -report\_with\_group$
6575 \>
\> -end
6576 \>
\> \text{pdflatex}\_surface\_family\_F13\_q32\_a2\_with\_group\_tex$
6577 \>
\> \text{open}\_surface\_family\_F13\_q32\_a2\_with\_group\_pdf$
6578
6579
6580 \# Kapadokya-1:
6581
6582 F13\_64a:
6583 \>
$\$(ORBITER\_PATH)\text{orbiter.out}\_v\_3$
6584 \>
\> -define F\_finite_field\_q\_64\_end$
6585 \>
\> -define P\_projective_space\_3\_F\_end$
6586 \>
\> -with P\_do$
6587 \>
\> -projective_space\_activity$
6588 \>
\> \> -define\_surface\_K1\_family\_F13\_2\_q\_64\_end$
6589 \>
\> -end$
6590 \>
\> -with K1\_do$
6591 \>
\> -cubic\_surface\_activity$
6592 \>
\> \> -report$
6593 \>
\> \> \> -report\_with\_group$
6594 \>
\> -end
6595
6596
6597 \# Kapadokya-2:
6598
6599 F13\_64b:
6600 \>
$\$(ORBITER\_PATH)\text{orbiter.out}\_v\_3$
6601 \>
\> -define F\_finite_field\_q\_64\_end$
6602 \>
\> -define P\_projective_space\_3\_F\_end$
6603 \>
\> -with P\_do$
6604 \>
\> -projective_space\_activity$
6605 \>
\> \> -define\_surface\_K2\_family\_F13\_18\_q\_64\_end$
6606 \>
\> -end$
6607 \>
\> -with K2\_do$
6608 \>
\> -cubic\_surface\_activity$
6609 \>
\> \> -report$
6610 \>
\> \> \> -report\_with\_group$

594
Colorado1:
$(ORBITER_PATH)orbiter.out-v.3\n$-define-F-finite_field-q:128-end\n$-define-P-projective_space-3-F-end\n$-with-P-do\n$-define_surface-CO-1-q:128-catalogue-0-end\n$-transform_inverse:"1,0,0,0,0,1,0,96,0,0,1,96,0,0,0,1,0"-end\n$-with-CO-1-do-end\n$-cubic_surface_activity-end\n$-report-end\n$-report_with_group-end\n
#recognize the arcs from Colorado-1,2,3:

Colorado2:
$(ORBITER_PATH)orbiter.out-v.3\n$-define-F-finite_field-q:128-end\n$-define-P-projective_space-3-F-end\n$-with-P-do\n$-define_surface-CO-2-q:128-catalogue-926-end\n$-transform_inverse:"1,0,0,0,0,1,0,32,0,0,1,32,0,0,0,1,0"-end\n$-with-CO-2-do-end\n$-cubic_surface_activity-end\n$-report-end\n$-report_with_group-end\n$-end-end\n
Colorado3:
$(ORBITER_PATH)orbiter.out-v.3\n$-define-F-finite_field-q:128-end\n$-define-P-projective_space-3-F-end\n$-with-P-do\n$-define_surface-CO-3-q:128-catalogue-928-end
#-transform_inverse:"1,0,0,0,0,1,0,59,0,0,1,59,0,0,0,1,0"

- end

- with-CO-3-do:
- cubic_surface_activity

- report

- report_with_group

- end

#-Colorado-1:

F13_128a:

$$(ORBITER\ PATH)orbiter.out-v.3$$

-define-F-finite_field-q.128-end

-define-P-projective_space-3-F-end

- with-P-do:

- projective_space_activity

- define_surface-CO-1-family_F13-2-q.128-end

- end

- with-CO-1-do:

- cubic_surface_activity

- report

- report_with_group

- end

#-Colorado-2:

F13_128b:

$$(ORBITER\ PATH)orbiter.out-v.3$$

-define-F-finite_field-q.128-end

-define-P-projective_space-3-F-end

- with-P-do:

- projective_space_activity

- define_surface-CO-2-family_F13-6-q.128-end

- end

- with-CO-2-do:

- cubic_surface_activity

- report

- report_with_group

- end

#-Colorado-3:

F13_128c:

$$(ORBITER\ PATH)orbiter.out-v.3$$

-define-F-finite_field-q.128-end
-define P -projective_space 3 -F -end \ 
-with P -do \ 
-projective_space_activity \ 
-define_surface CO 3 -family F 13 14 -q 128 -end \ 
-end \ 
-with CO 3 -do \ 
cubic_surface_activity \ 
-report \ 
-report_with_group \ 
-end \ 
move_two_lines: \ 
$(ORBITER\_PATH) orbiter.out -v 5 \ 
-F -finite_field -q 8 -end \ 
-do -finite_field_activity \ 
-move_two_lines_in_hyperplane_stabilizer 65 4680 72 657 -end \ 
\ 
F alpha beta gamma delta: \ 
$(ORBITER\_PATH) orbiter.out -v 3 \ 
-F -finite_field -q 7 -end \ 
-do -finite_field_activity \ 
-parse_and_evaluate \ 
"F alpha beta gamma delta " x0,x1,x2,x3 " \ 
$(F ALPHA\_BETA\_GAMMA\_DELTA) \ 
"alpha=2,beta=3,gamma=4,delta=5" \ 
-end \ 
do -Tpng F alpha beta gamma delta.gv > F alpha beta gamma delta.png \ 
\ 
F abcd Eckardt q31: \ 
$(ORBITER\_PATH) orbiter.out -v 3 \ 
-F -finite_field -q 31 -end \ 
-F -projective_space 3 -F -end \ 
-with F -do \ 
-projective_space_activity \ 
-define_surface F abcd -q 31 \ 
-by_equation "F abcd" \ 
"DF {a,b,c,d}\D" "X0,X1,X2,X3" \ 
$(F abcd_eqn) \ 
"a=2,b=30,c=30,d=2" \ 
"D=2,b=30,c=30,d=2\D" \ 
-end \ 

F_abcd:

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3}\$

\$\text{define-F-\_finite_field-\_q.7-\_end}\$

\$\text{with-F-\_do}\$

\$\text{finite_field\_activity}\$

\$\text{parse\_and\_evaluate-}\text{"Fabcd.\_X0,\_X1,\_X2,\_X3"}\$

\$\text{define-\_P-\_projective\_space-3-\_F-\_end}\$

\$\text{projective_space\_activity}\$

\$sweep.4.27.4.27.\_q.7-\_by\_equation-}\text{"Fabcd\"}\$

\$\text{DF}\_\{a,b,c,d\}\text{\"X0,\_X1,\_X2,\_X3\"}\$

\$\text{define-\_P-\_do}\$

\$\text{projective_space\_activity}\$

\$\text{sweep.4.27.\_sweep.4.27.\_\_q.7-\_\_by\_equation-}\text{"Fabcd\"}\$

\$\text{DF}\_\{a,b,c,d\}\text{\"X0,\_X1,\_X2,\_X3\"}\$

\$\text{a=2,b=3,\_c=4,\_d=5}\$

\$\text{Da=2,b=3,\_c=4,\_d=5}\$

\$\text{end-\_P-\_do}\$

\$\text{end-\_projective_space\_activity}\$

\$\text{end-\_finite_field\_activity}\$

\$\text{end-\_with-F-\_do}\$

\$\text{end-\_define-F-\_finite_field-\_q.7-\_end}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$

\$\text{end-\_\_define-P-\_do}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$

\$\text{end-\_\_define-P-\_do}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$

\$\text{end-\_\_define-P-\_do}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$

\$\text{end-\_\_define-P-\_do}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$

\$\text{end-\_\_define-P-\_do}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$

\$\text{end-\_\_define-P-\_do}\$

\$\text{end-\_\_define-P-\_projective_space-3-\_F-\_end}\$
\begin{verbatim}
define_surface F_2345 -q7 
by_equation "F_alpha_beta_gamma_delta" 
"DF{\alpha,\beta,\gamma,\delta}D".x0,x1,x2,x3. 
$(F_{\alpha,\beta,\gamma,\delta})$ 
"alpha=2,beta=3,gamma=4,delta=5" 
"1,5,0,0,3,6,0,0,1,1,3,0,5,5,0,3,1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4" 
end 
end 
-override group 6:2 
"1,5,0,0,3,6,0,0,1,1,3,0,5,5,0,3,1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4,1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4",
end

-cubic_surface_activity 
do 
report_with_group 
with F_2345 
do 
report 
report_with_group 
end
\end{verbatim}
6846 ▶ ▶ -define F-finite_field-q7-end\ 
6847 ▶ ▶ -define P-projective_space-3-F-end\ 
6848 ▶ ▶ -with P-do\ 
6849 ▶ ▶ -projective_space_activity\ 
6850 ▶ ▶ -sweep_4_15_lines-sweep_4_q7-q7-\ 
6851 ▶ ▶ -by_equation-\ 
6852 ▶ ▶ ▶ "F_alpha_beta_gamma_delta".\ 
6853 ▶ ▶ ▶ "DF_{\alpha,\beta,\gamma,\delta}\".\ 
6854 ▶ ▶ ▶ "x0,x1,x2,x3"\ 
6855 ▶ ▶ ▶ $(F_ALPHA_BETA_GAMMA_DELTA)\ 
6856 ▶ ▶ ▶ "alpha=2,beta=3,gamma=4,\delta=5"\ 
6857 ▶ ▶ ▶ "D\alpha=2,\beta=3,\gamma=4,\delta=5\"\ 
6858 ▶ ▶ -end\ 
6859   
6860  #User-time:0:30
6861  # 348-parameter_sets
6862  #F_alpha_beta_gamma_delta_q7_points.txt
6863  #F_alpha_beta_gamma_delta_q7_sweep.csv
6864  #F_alpha_beta_gamma_delta_q7_sweep4_15_data.csv
6865  
6866  
6867  
6868  
6869  F_alpha_beta_gamma_delta_q7_r...
-by_equation"F_alpha_beta_gamma_delta"
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\$(F_ALPHA_BETA_GAMMA_DELTA)\$
"\DF\{\alpha,\beta,\gamma,\delta\}\D"."x0,x1,x2,x3".
\texttt{\$\textit{ORBITER\_PATH}\texttt{orbiter.out-v.3\textbackslash} \textbackslash}
\texttt{\textbackslash define-F-finite_field-\texttt{q.49-end\textbackslash} \textbackslash}
\texttt{\textbackslash define-P-projective_space-3-F-end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash with-P-do\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash projective_space_activity\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash classify_surfaces_with_double_sixes-Surf27-W-end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash with-Surf27-do\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash classification_of_cubic_surfaces_with_double_sixes_activity\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash recognize-\textbackslash} \texttt{\textbackslash -q.49\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash -by_coefficients:2,5,1,16,4,10,1,18,4,8,4,9,3,11,4,14\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash -end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash -print_symbols}
\texttt{\textbackslash}
\texttt{\textbackslash McKean.15lines_q7:}
\texttt{\textbackslash $(\textit{ORBITER\_PATH})\texttt{orbiter.out-v.3\textbackslash} \textbackslash}
\texttt{\textbackslash define-F-finite_field-\texttt{q.7-end\textbackslash} \textbackslash}
\texttt{\textbackslash define-P-projective_space-3-F-end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash with-P-do\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash projective_space_activity\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash define_surface:S\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash -by_coefficients:$(\textit{SURFACE\_MCKEAN.15\_LINES})-q.7\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash -end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash -end\textbackslash} \texttt{\textbackslash}
\texttt{\textbackslash \#pdflatex-surface_by_coefficients_q7_report.tex}
\texttt{\textbackslash \#open-surface_by_coefficients_q7_report.pdf}
\texttt{\textbackslash}
\texttt{\textbackslash \#2-Eckardt-points
Section 7.2: Cubic Surfaces and Quartic Curves.

F.4.4.3.3.q7:

-define F - finite_field - q7 - end:
-define P - projective_space 3 F - end:
-withe P - do:
-projective_space_activity:
-define_surface - q7 - by_equation:
  "F_alpha_beta_gamma_delta":
  "DF_{\alpha, \beta, \gamma, \delta}\D":
  "x0, x1, x2, x3":
  $(F_{\alpha, \beta, \gamma, \delta})$
  "alpha=4, beta=4, gamma=3, delta=3":
  \"\D\alpha=4, \beta=4, \gamma=3, \delta=3\D":
  -end:
  -end:

#pdflatex surface_equation_F_alpha_beta_gamma_delta_q7_report.tex
#open surface_equation_F_alpha_beta_gamma_delta_q7_report.pdf

#has 4 Eckardt points

F_alpha_beta_gamma_delta_points.txt:
-define F - finite_field - q7 - end:
-define P - projective_space 3 F - end:
-withe P - do:
-projective_space_activity:
-sweep nothing:
-define_surface - q7 - by_equation:
  "F_alpha_beta_gamma_delta":
  "DF_{\alpha, \beta, \gamma, \delta}\D":
  "x0, x1, x2, x3":
  $(F_{\alpha, \beta, \gamma, \delta})$
  "alpha=2, beta=3, gamma=4, delta=5":
  \"\D\alpha=2, \beta=3, \gamma=4, \delta=5\D":
  -end:
  -end:
quartic_curve_9_0_report::
$\text{(ORBITER_PATH)}$orbiter.out-\text{-}v.3\n> -define\text{-}F\text{-}finite_field\text{-}q.9\text{-}end\n> -define\text{-}P\text{-}projective_space\text{-}2\text{-}F\text{-}end\n> -with\text{-}P\text{-}do\n> -projective_space_activity\n> -define_quartic_curve\text{-}C\text{-}q.9\n> -catalogue\text{-}0\text{-}end\n> -end\n> -with\text{-}C\text{-}do\n> -quartic_curve_activity\n> -report\n> -end\n> pdflatex\text{-}quartic_curve_catalogue_q9_iso0_report.tex
> open\text{-}quartic_curve_catalogue_q9_iso0_report.pdf
quartic_curve_13_0_report::
$\text{(ORBITER_PATH)}$orbiter.out-\text{-}v.3\n> -define\text{-}F\text{-}finite_field\text{-}q.13\text{-}end\n> -define\text{-}P\text{-}projective_space\text{-}2\text{-}F\text{-}end\n> -with\text{-}P\text{-}do\n> -projective_space_activity\n> -define_quartic_curve\text{-}C\text{-}q.13\text{-}catalogue\text{-}0\text{-}end\n> -end\n> -with\text{-}C\text{-}do\n> -quartic_curve_activity\n> -report\n> -end\n> pdflatex\text{-}quartic_curve_catalogue_q13_iso0_report.tex
> open\text{-}quartic_curve_catalogue_q13_iso0_report.pdf
quartic_curve_13_1_report::
$\text{(ORBITER_PATH)}$orbiter.out-\text{-}v.3\n> -define\text{-}F\text{-}finite_field\text{-}q.13\text{-}end\n> -define\text{-}P\text{-}projective_space\text{-}2\text{-}F\text{-}end\n> -with\text{-}P\text{-}do\n> -projective_space_activity\n> -define_quartic_curve\text{-}C\text{-}q.13\text{-}catalogue\text{-}1\text{-}end\n> -end\n> -with\text{-}C\text{-}do\n
surface_4_0_quartic_curves:

surface_4_0_quartic_curves:

NB_CUBIC_SURFACES_Q7=1

quartic_curves_q7:

quartic_curves_q7:
quartic_curves_q7_classify:
$$(\text{ORBITER\_PATH})\text{orbiter.out}\!-\!v.3\!\backslash\n$$
-list_arguments\!
-define F \-finite\_field\-q\7\-end\!
-define P \-projective\_space\-2\-F\-end\!
-with P \-do\!
-projective\_space\_activity\!
-classify\_quartic\_curves\_with\_substructure\!
-surface\_catalogue\_q7\_iso\%d\_quartics.csv\!
-$$$(\text{ORBITER\_PATH})\text{orbiter.out}\!-\!v.3\!\backslash\n$$
-export_all\_quartic\_curves\!
-end\!
-print\_symbols\!

NB\_CUBIC\_SURFACES\_Q13=4

quartic_curves_q13:
$$(\text{ORBITER\_PATH})\text{orbiter.out}\!-\!v.3\!\backslash\n$$
-list_arguments\!
-define F \-finite\_field\-q\13\-end\!
-define P \-projective\_space\-3\-F\-end\!
-loop L\!-\!0\!\backslash\n$$$(\text{ORBITER\_PATH})\text{orbiter.out}\!-\!v.3\!\backslash\n$$
-export_all\_quartic\_curves\!
-end\!
-print\_symbols\!

#pdflatex surface\_catalogue\_q13\_iso0\_quartics.tex
quartic_curves_q13.classify:

\begin{verbatim}
$\text{(ORBITER \ PATH)}\text{orbiter.out -v 3}\$
\end{verbatim}

\begin{verbatim}
define F -finite_field -q 13 -end\$
define P -projective_space 2 F -end\$
with P -do

classify_quartic_curves_with_substructure\$

define surface.catalogue_q13_iso%d.quartics.csv

3 4 4 -quartic_curves_q13\$

end\$

end\$

print_symbols\$

\end{verbatim}
quartic_curves_q17.classify:
$\text{\verb|$(ORBITER_PATH)orbiter.out-v.3|}$
$\text{\verb|list_arguments|}$
$\text{\verb|define F finite_field q17 end|}$
$\text{\verb|define P projective_space 2 F end|}$
$\text{\verb|with P do|}$
$\text{\verb|projective_space activity|}$
$\text{\verb|classify_quartic_curves_with_substructure|}$
$\text{\verb|surface_catalogue_q17_iso%d_quartics.csv|}$
$\text{\verb|$(NB\_CUBIC\_SURFACES\_Q17)\_3\_4\_quartic\_curves\_q17|}$
$\text{\verb|end|}$
$\text{\verb|print_symbols|}$

#User.time: 2:33
#q17
#The number of types of quartic curves is 7
#idx:: ago
#0:: 24
#1:: 24
#2:: 4
#3:: 96
#4:: 6
#5:: 8
#6:: 2

NB\_CUBIC\_SURFACES\_Q19=10

quartic_curves_q19:
quartic_curves_q19.classify:

$$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 3$$

- define-F -finite_field -q 19 -end
- define-P -projective_space:2:F -end
- with-P -do
- projective_space_activity
- classify_quartic_curves_with_substructure
- surface_catalogue_q19_iso%d_quartics.csv
- $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 3$
- list_arguments
- define-F -finite_field -q 19 -end
- define-P -projective_space:2:F -end
- with-P -do
- projective_space_activity
- set_stabilizer: 4
- export_all_quartic_curves
- report
- end
- print_symbols

# writes:
quartic_curves_q19_canonical_data.csv
quartic_curves_q19_canonical.tex

# 14 isomorphism types:
ago-dist: 4^1, 9^1, 2^4, 6^2, 8^3, 24^3

quartic_curves_q19_set_stabilizer:

$$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 3$$
- list_arguments
- define-F -finite_field -q 19 -end
- define-P -projective_space:2:F -end
- with-P -do
- projective_space_activity
- set_stabilizer: 4
- surface_catalogue_q19_iso%d_quartics.csv
$\text{surface}_13\_0\_\text{quartics}:$

\$\text{(ORBITER\_PATH)}\text{orbiter.out}\!-\!v\!-\!3\$$

\$\text{define}\!-\!F\!-\!\text{finite_field}\!-\!q\!\!-\!13\!-\!\text{end}\$$

\$\text{define}\!-\!P\!-\!\text{projective_space}\!-\!3\!F\!-\!\text{end}\$$

\$\text{define}\!-\!S13\!\_\!0\!-\!q\!13\!-\!\text{catalogue}\!-\!0\!-\!\text{end}\$$

\$\text{with}\!-\!P\!-\!\text{do}\$$

\$\text{define}\!-\!S13\!\_\!0\!-\!\text{do}\$$

\$\text{cubic_surface_activity}\$$

\$\text{export\_all\_quartic\_curves}\$$

\$\text{end}\$$

\$\\text{pdflatex}\!\text{surface\_catalogue\_q13\_iso0\_quartics.tex}\$$

\$\\text{open}\!\text{surface\_catalogue\_q13\_iso0\_quartics.pdf}\$$

$\text{surface}_13\_1\_\text{quartics}:$

\$\text{(ORBITER\_PATH)}\text{orbiter.out}\!-\!v\!-\!3\$$

\$\text{define}\!-\!F\!-\!\text{finite_field}\!-\!q\!\!-\!13\!-\!\text{end}\$$

\$\text{define}\!-\!P\!-\!\text{projective_space}\!-\!3\!F\!-\!\text{end}\$$

\$\text{define}\!-\!S13\!\_\!1\!-\!q\!13\!-\!\text{catalogue}\!-\!1\!-\!\text{end}\$$

\$\text{with}\!-\!S13\!\_\!1\!-\!\text{do}\$$

\$\text{cubic_surface_activity}\$$

\$\text{export\_all\_quartic\_curves}\$$

\$\text{end}\$$

\$\\text{pdflatex}\!\text{surface\_catalogue\_q13\_iso1\_quartics.tex}\$$

\$\\text{open}\!\text{surface\_catalogue\_q13\_iso1\_quartics.pdf}\$$

$\text{quartic\_curve}_13\_2\_\text{group}:$
quartic_curve_25_report:

> $(ORBITER_PATH)orbiter.out -v.3
> -define:F=finite_field=\text{q}25-end
> -define:P=projective_space=2\text{F}-end
> -loop:L=0-18-1\-
> -with:P-do
> -define\-

\text{-projective_space_activity}

> -define_quartic_curve=QC25\-
> -q=25-catalogue\-
> -end\-

> -loop:L=0-18-1\-
> -with:QC25\-
> -do
> -quartic_curve_activity
> -report
> -end\-

> -end_loop
> -print_symbols\-
> -loop:L=0-18-1\-
> -with:QC25\-
> -do
> -quartic_curve_activity
> -report
> -end\-

> -print_symbols
> pdflatex\-
> quartic_curve_catalogue_q25_iso0_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso1_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso2_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso3_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso4_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso5_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso6_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso7_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso8_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso9_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso10_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso11_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso12_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso13_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso14_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso15_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso16_report.tex
> pdflatex\-
> quartic_curve_catalogue_q25_iso17_report.tex
> gs\-sDEVICE=pdfwrite\-r120\-oquartic_curve_catalogue_q25.pdf\-
> quartic_curve_catalogue_q25_iso0_report.pdf
> quartic_curve_catalogue_q25_iso1_report.pdf
> quartic_curve_catalogue_q25_iso2_report.pdf
> quartic_curve_catalogue_q25_iso3_report.pdf
> quartic_curve_catalogue_q25_iso4_report.pdf
> quartic_curve_catalogue_q25_iso5_report.pdf

quartic_curve_13_table:\n$(ORBITER\ PATH)orbiter.out-v.3\n-define-F-finite_field-q.13-end\n-define-P-projective_space.2.F-end\n-with-P-do\n-projective_space_activity\n-table_of_quartic_curves\n-end

#quartic_curves_q13_info.csv

quartic_curve_19_table:\n$(ORBITER\ PATH)orbiter.out-v.3\n-define-F-finite_field-q.19-end\n-define-P-projective_space.2.F-end\n-with-P-do\n-projective_space_activity\n-table_of_quartic_curves\n-end

quartic_curve_19_table_latex:\n$(ORBITER\ PATH)orbiter.out-v.3\n-csv_file_latex-1-quartic_curves_q19_info.csv
~/bin/tth-quartic_curves_q19_info.tex

quartic_curve_25_table:\n$(ORBITER\ PATH)orbiter.out-v.3\n
define F - finite field - q 25 - end \n\ndefine P - projective space - 2 F - end \nwith P - do \nprojective space activity \n\ntable of quartic curves \nend \n
# quartic_curves_q25_info.csv

quartic_curve_27_table:  
\ndefine F - finite field - q 27 - end \n\ndefine P - projective space - 2 F - end \nwith P - do \nprojective space activity \ntable of quartic curves \nend  

# quartic_curves_q27_info.csv

quartic_curve_29_table:  
\ndefine F - finite field - q 29 - end \n\ndefine P - projective space - 2 F - end \nwith P - do \nprojective space activity \ntable of quartic curves \nend  

# quartic_curves_q29_info.csv

quartic_curve_31_table:  
\ndefine F - finite field - q 31 - end \n\ndefine P - projective space - 2 F - end \nwith P - do \nprojective space activity \ntable of quartic curves \nend  

# quartic_curves_q31_info.csv
surface_25_12:
\$(\text{ORBITER_PATH})\text{orbiter.out-}\text{-v.3}\$
\>$\text{-define }F\text{-finite_field}-q\text{-25-}\text{-end}\$
\>$\text{-define }P\text{-projective_space-3 }F\text{-end}\$
\>$\text{-with }P\text{-do}\$
\>$\text{-projective_space_activity}\$
\>$\text{-define_surface }S25\text{\_12-}q\text{-25-}\text{-catalogue}\text{-12-}\text{-end}\$
\>$\text{-end}\$
\>$\text{-with }S25\text{\_12-}\text{-do}\$
\>$\text{-cubic_surface_activity}\$
\>$\text{-report}\$
\>$\text{-report_with_group}\$
\>$\text{-end}\$
\>$\text{pdflatex}\text{-surface_catalogue_q25_iso12_with_group.tex}\$
\>$\text{open}\text{-surface_catalogue_q25_iso12_with_group.pdf}\$

surface_25_12_t1:
\$(\text{ORBITER_PATH})\text{orbiter.out-}\text{-v.3}\$
\>$\text{-define }F\text{-finite_field}-q\text{-25-}\text{-end}\$
\>$\text{-define }P\text{-projective_space-3 }F\text{-end}\$
\>$\text{-with }P\text{-do}\$
\>$\text{-projective_space_activity}\$
\>$\text{-define_surface }S25\text{\_12-}q\text{-25-}\text{-catalogue}\text{-12-}\text{-end}\$
\>$\text{-transform}\text{"1,0,0,16,0,1,0,18,0,0,1,1,0"}\$
\>$\text{-end}\$
\>$\text{-with }S25\text{\_12-}\text{-do}\$
\>$\text{-cubic_surface_activity}\$
\>$\text{-report}\$
\>$\text{-report_with_group}\$
\>$\text{-end}\$
\>$\text{pdflatex}\text{-surface_catalogue_q25_iso12_with_group.tex}\$
\>$\text{open}\text{-surface_catalogue_q25_iso12_with_group.pdf}\$

surface_25_12_t2:
\$(\text{ORBITER_PATH})\text{orbiter.out-}\text{-v.3}\$
\>$\text{-define }F\text{-finite_field}-q\text{-25-}\text{-end}\$
\>$\text{-define }P\text{-projective_space-3 }F\text{-end}\$
\>$\text{-with }P\text{-do}\$
\>$\text{-projective_space_activity}\$
\>$\text{-define_surface }S25\text{\_12-}q\text{-25-}\text{-catalogue}\text{-12-}\text{-end}\$
\>$\text{-transform}\text{"1,0,0,16,0,1,0,18,0,0,1,1,0"}\$
surface_25_12_t3:
$\{(\text{ORBITER PATH})\text{orbiter.out: -v:3:} \}
-define F- finite_field -q 25 -end-
-define P- projective_space 3 F- end-
-with P- do-
-projective_space_activity-
-define_surface S25_12 - q 25 - catalogue 12-
-transform "1, 0, 0, 16, 0, 1, 0, 18, 0, 0, 0, 1, 1, 0"
-transform_inverse "16, 0, 1, 0, 3, 5, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0"
-transform "3, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0"
-end-
-cubic_surface_activity-
-report-
-report_with_group-
-end-
-pdflatex -surface_catalogue q25 iso12 with group.tex
-open -surface_catalogue q25 iso12 with group.pdf

surface_25_12_t4:
$\{(\text{ORBITER PATH})\text{orbiter.out: -v:3:} \}
-define F- finite_field -q 25 -end-
-define P- projective_space 3 F- end-
-with P- do-
-projective_space_activity-
-define_surface S25_12 - q 25 - catalogue 12-
-transform "1, 0, 0, 16, 0, 1, 0, 18, 0, 0, 0, 1, 1, 0"
-transform_inverse "16, 0, 1, 0, 3, 5, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0"
-transform "3, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0"
-end-
-cubic_surface_activity-
-report-
-report_with_group-
-end-
-pdflatex -surface_catalogue q25 iso12 with group.tex
-open -surface_catalogue q25 iso12 with group.pdf
7645 ▷ ▷ -with S25_12-do:\n7646 ▷ ▷ -cubic_surface_activity:\n7647 ▷ ▷ ▷ -report:\n7648 ▷ ▷ ▷ -report_with_group:\n7649 ▷ ▷ -end\n7650 ▷ pdflatex:surface_catalogue_q25_iso12_with_group.tex\n7651 ▷ open:surface_catalogue_q25_iso12_with_group.pdf\n7652\n7653\n7654 surface_25_12_t5:\n7655 ▷ $(ORBITER PATH)orbiter.out-v.3:\n7656 ▷ ▷ -define F-finite_field-q 25-end:\n7657 ▷ ▷ -define P-projective_space-3 F-end:\n7658 ▷ ▷ -with P-do:\n7659 ▷ ▷ -projective_space_activity:\n7660 ▷ ▷ ▷ -define_surface S25_12-q 25-catalogue-12:\n7661 ▷ ▷ ▷ -transform "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,0":\n7662 ▷ ▷ ▷ -transform_inverse "16,0,1,0,3,5,1,0,0,0,1,0,-0,0,0,1,0":\n7663 ▷ ▷ ▷ -transform "3,0,0,0,-0,1,0,0,1,0,0,0,0,1,0":\n7664 ▷ ▷ ▷ -transform_inverse "1,0,0,0,0,1,0,0,0,0,1,0,13,2,2,1,0":\n7665 ▷ ▷ ▷ -transform_inverse "1,0,0,0,0,1,0,0,13,1,1,0,-0,0,0,1,0":\n7666 ▷ ▷ ▷ -end:\n7667 ▷ ▷ -end:\n7668 ▷ ▷ -with S25_12-do:\n7669 ▷ ▷ -cubic_surface_activity:\n7670 ▷ ▷ ▷ -report:\n7671 ▷ ▷ ▷ -report_with_group:\n7672 ▷ ▷ -end\n7673 ▷ pdflatex:surface_catalogue_q25_iso12_with_group.tex\n7674 ▷ open:surface_catalogue_q25_iso12_with_group.pdf\n7675\n7676 PG_2_25:\n7677 ▷ $(ORBITER PATH)orbiter.out:\n7678 ▷ ▷ -define F-finite_field-q 25-end:\n7679 ▷ ▷ -define P-projective_space-2 F-end:\n7680 ▷ ▷ -with P-do-projective_space_activity-cheat_sheet-end\n7681 ▷ pdflatex:PG_2_25.tex:\n7682 ▷ open:PG_2_25.pdf\n7683\n7684\n7685\n7686\n7687 PG_2_25_lines:\n7688 ▷ $(ORBITER PATH)orbiter.out-v.5:\n7689 ▷ -orbiter_path $(ORBITER PATH)\n7690 ▷ ▷ -define G-linear_group-PGGL 3-25-end:\n7691 ▷ ▷ -define G_on_lines-modified_group-from G:\n
617
\begin{verbatim}
7692 \> \> \> \> \> \> \> -on_k_subspaces.2\;
7693 \> \> \> \> \> \> \> -end\;
7694 \> \> \> \> \> \> \> -with_G_on_lines.-do\;
7695 \> \> \> \> \> \> \> -group_theoretic_activity\;
7696 \> \> \> \> \> \> \> -poset_classification_control\;
7697 \> \> \> \> \> \> \> -problem_label-PGGL.3.25\;
7698 \> \> \> \> \> \> \> -depth.3.-draw_poset.-draw_options.-radius.200.-end.-report.-end\;
7699 \> \> \> \> \> \> \> -recognize."0,25,650".\;
7700 \> \> \> \> \> \> \> -recognize."430,16,364".\;
7701 \> \> \> \> \> \> \> -end\;
7702 \> \> \> \> \> \> \> -orbits_on_subsets.3\;
7703 \> \> \> \> \> \> \> -report\;
7704 \> \> \> \> \> \> \> -end
7705 \> \> pdflatex-PGGL.3.25.poset.tex
7706 \> \> open-PGGL.3.25.poset.pdf
7707
7708 surface_25.12_t6:
7709 \> $(\text{ORBITER_PATH})\text{oritzer.out.-v.3}\
7710 \> \> -define_F.-finite_field.-q.25.-end\;
7711 \> \> -define_P.-projective_space.3_F.-end\;
7712 \> \> -with_P.-do\;
7713 \> \> -projective_space_activity\;
7714 \> \> \> \> \> \> -define_surface:S25.12.-q.25.-catalogue.12\;
7715 \> \> \> \> \> \> -transform:"1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0".\;
7716 \> \> \> \> \> \> -transform_inverse:"16,0,1,0,3,5,1,0,-0,0,1,0,0,-0,0,1,0,0".\;
7717 \> \> \> \> \> \> -transform:"3,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,-0".\;
7718 \> \> \> \> \> \> -transform_inverse:"1,0,0,0,0,1,0,0,0,0,1,0,0,1,3,2,2,1,-0".\;
7719 \> \> \> \> \> \> -transform_inverse:"1,0,0,0,0,1,0,0,-13,1,1,0,-0,0,1,0,-0".\;
7720 \> \> \> \> \> \> -transform:"3,8,8,0,22,13,22,0,-14,19,15,0,-0,0,0,1,1,-1".\;
7721 \> \> \> \> \> \> -transform_inverse:"16,0,0,0,-0,16,0,0,21,21,21,0,-0,0,1,0,-0".\;
7722 \> \> \> \> \> \> -end\;
7723 \> \> \> \> \> \> -end\;
7724 \> \> \> \> \> \> -with:S25.12.-do\;
7725 \> \> \> \> \> \> -cubic_surface_activity\;
7726 \> \> \> \> \> \> -report\;
7727 \> \> \> \> \> \> -report_with_group\;
7728 \> \> \> \> \> \> -end
7729 \> pdflatex-surface_catalogue.q25.iso12_with_group.tex
7730 \> open-surface_catalogue.q25.iso12_with_group.pdf
7731
7732
7733
7734 PG_2.25_stab_of_triangle:
7735 \> $(\text{ORBITER_PATH})\text{oritzer.out.-v.5}\
7736 \> \> -oritzer_path.$(\text{ORBITER_PATH}).\;
7737 \> \> -define_G.-linear_group.-PGGL.3.25\;
7738 \> \> \> \> \> \> -subgroup_by_generators."triangle_stab":6912.7.\;
\end{verbatim}
surface_25_12_t7: 
$(ORBITER\_PATH)\$orbiter.out -v 3\n-define F -finite_field -q 25 -end\n-define P -projective_space -3 F -end\n-define_surface -S25_12 -q 25 -catalogue -12\n-transform -1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0\n-transform_inverse -16,0,1,0,-3,5,1,0,-0,0,1,0,-0,0,1,0,-0\n-transform_inverse -1,0,0,0,-1,0,0,-0,1,0,-0,0,0,1,-0\n-transform_inverse -1,0,0,0,-1,0,0,-0,1,0,-0,13,2,2,1,-0\n-transform -3,8,8,0,22,13,22,0,14,19,15,0,-0,0,0,1,-1\n-transform_inverse -16,0,0,0,-16,0,0,21,21,21,0,0,0,1,0,-0\n-transform -1,0,0,0,0,5,0,0,-0,0,17,0,-0,0,0,1,1,-0\n-end\n-end\n-with S25_12 -do\n-cubic_surface_activity\n-report\n-report_with_group\n-end\n\$pdflatex surface_catalogue_q25_iso12_with_group.tex\nopen surface_catalogue_q25_iso12_with_group.pdf
surface_25_12_t8:
$\$(\text{ORBITER\_PATH})\text{orbiter.out\_v.3}\$
$\text{define-F\_finite_field-q-25\_end}\$
$\text{define-P\_projective_space-3-F\_end}\$
$\text{with-P\_do}\$
$\text{projective\_space\_activity}\$
$\text{define_surface-S25_12-q-25\_catalogue-12}\$
$\text{transform-}1,0,0,16,0,1,0,18,0,0,1,8,0,0,1,1,0\text{^{-1}}\$
$\text{transform_inverse-}16,0,1,0,3,5,1,0,0,1,0,0,0,1,0\text{^{-1}}\$
$\text{transform-}3,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0\text{^{-1}}\$
$\text{transform_inverse-}1,0,0,0,0,1,0,0,0,1,0,13,2,2,1,0\text{^{-1}}\$
$\text{transform-}1,0,0,0,0,1,0,0,0,0,1,0\text{^{-1}}\$
$\text{transform_inverse-}3,8,8,0,22,13,22,0,14,19,15,0,0,0,1,1\text{^{-1}}\$
$\text{transform_inverse-}16,0,0,0,0,16,0,0,21,21,0,0,0,1,0\text{^{-1}}\$
$\text{transform-}1,0,0,0,0,5,0,0,0,0,17,0,0,0,1,1\text{^{-1}}\$
$\text{transform-}1,0,0,0,0,1,0,0,0,0,24,0\text{^{-1}}\$
$\text{end}\$
$\text{with-S25_12\_do}\$
$\text{-cubic\_surface\_activity}\$
$\text{-report}\$
$\text{-report\_with\_group}\$
$\text{-end}\$
$\text{pdflatex\_surface\_catalogue_q25\_iso12\_with\_group\_tex}\$
$\text{open\_surface\_catalogue_q25\_iso12\_with\_group\_pdf}\$

surface_25_12_t9:
$\$(\text{ORBITER\_PATH})\text{orbiter.out\_v.3}\$
$\text{define-F\_finite_field-q-25\_end}\$
$\text{define-P\_projective_space-3-F\_end}\$
$\text{with-P\_do}\$
$\text{projective\_space\_activity}\$
$\text{define_surface-S25_12-q-25\_catalogue-12}\$
$\text{transform-}1,0,0,16,0,1,0,18,0,0,1,8,0,0,1,1,0\text{^{-1}}\$
$\text{transform_inverse-}16,0,1,0,3,5,1,0,0,1,0,0,0,1,0\text{^{-1}}\$
$\text{transform-}3,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0\text{^{-1}}\$
$\text{transform_inverse-}1,0,0,0,0,1,0,0,0,0,1,0,13,2,2,1,0\text{^{-1}}\$
$\text{transform_inverse-}1,0,0,0,0,1,0,0,0,0,1,0\text{^{-1}}\$
$\text{transform-}3,8,8,0,22,13,22,0,14,19,15,0,0,0,1,1\text{^{-1}}\$
$\text{transform_inverse-}16,0,0,0,0,16,0,0,21,21,0,0,0,1,0\text{^{-1}}\$
$\text{transform-}1,0,0,0,0,5,0,0,0,0,17,0,0,0,1,1\text{^{-1}}\$
$\text{transform-}1,0,0,0,0,1,0,0,0,0,24,0\text{^{-1}}\$
$\text{end}\$
$\text{with-S25_12\_do}\$

620
surface_25_12_t8_quartic_curves:
\$(ORBITER_PATH)orbiter.out --v.3\n\$define-F-\{finite_field-q:25\}-\$end\n\$define-P-\{projective_space:3-F\}-\$end\n\$with-P-\$do\n\$projective_space_activity\n\$define_surface-S25_12\-q:25\-\$catalogue:12\n\$transform_inverse-"1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0."
\$transform_inverse-"16,0,1,0,3,5,1,0,0,0,1,0,0,0,0,1,0,-0."
\$transform_inverse-"16,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,-0."
\$transform_inverse-"1,0,0,0,0,1,0,0,0,1,3,2,2,1,0,-0."
\$transform_inverse-"1,0,0,0,0,1,0,0,0,1,1,0,0,0,0,1,0,-0."
\$transform_inverse-"1,0,0,0,0,1,0,0,1,3,1,1,0,0,0,1,0,-0."
\$transform_inverse-"1,0,0,0,0,1,0,0,1,1,0,0,0,0,1,0,-0."
\$transform_inverse-"1,0,0,0,0,1,0,0,0,1,0,0,0,0,24,0,-0."
\$all_quartic_curves\n\$end\n\$with-S25_12-\$do\n\$cubic_surface_activity\n\$end\n\$with-S25_12-\$do\n\$cubic_surface_activity\n\$export_all_quartic_curves\n\$end
\$define_F\{finite_field\-_q:13\}-\$end
\$define_P\{projective_space\-_2_F\}-\$end
\$with_P\$do
\$projective_space_activity\n
quartic_curve_13_0_surface:
\$(ORBITER_PATH)orbiter.out --v.3\n\$define-F-\{finite_field\-_q:13\}-\$end
\$define_P\{projective_space\-_2_F\}-\$end
\$with_P\$do
\$projective_space_activity\n
621
-define_quartic_curve QC13_0.q13 catalogue 0. -end \n-define QC13 0. -q 13. -catalogue 0. -end \n-with QC13_0. -do \n-quartic_curve_activity \n-create_surface \n-end \n#surface_equation: 0, 0, 0, 3, 0, 1, 0, 12, 7, 0, 4, 4, 0, 7, 12, 6, 2, 5, 10 \n#9, 2, 3, 4, 1, 6, 12, 8, 7, 9, 4, 11, 4, 12, 7, 14, 12, 15, 6, 16, 2, 17, 5, 18, 10, 19 \nquartic_curve_13_0_surface_create: \n$(_ORBITER_PATH) orbiter.out -v 3 \n-define F -finite_field -q 13 -end \n-define P -projective_space 3 F -end \n-with P -do \n-projective_space_activity \n-define_surface -by_coefficients \n"9, 2, 3, 4, 1, 6, 12, 8, 7, 9, 4, 11, 4, 12, 7, 14, 12, 15, 6, 16, 2, 17, 5, 18, 10, 19" \n-q 13 -end \n-end \n-with S -do \ncubic_surface_activity \n-report \n-end \n-with Surf27 -do \nclassify_surfaces_with_double_sixes Surf27 -W -end \n-end \n-with Surf27 -do \n-section 7.3: cubic_surfaces_classification \nSECTION_CUBIC_SURFACES_CLASSIFICATION: \nsurface_classify q4: \n$(_ORBITER_PATH) orbiter.out -v 5 \n-define F -finite_field -q 4 -end \n-define P -projective_space 3 F -end \n-with P -do \n-projective_space_activity \n-classify_surfaces_with_double_sixes Surf27 -W -end \n-end \n-with Surf27 -do
7927  ▶ ▶  -classification_of_cubic_surfaces_with_double_sixes_activity.txt
7928  ▶ ▶    ▶ -report:end.txt
7929  ▶ ▶  -end.txt
7930  ▶ ▶  -print_symbols
7931  ▶  pdflatex:Surfaces_q4.tex
7932  ▶  open:Surfaces_q4.pdf
7933  #time:0:00
7935
7936
7937
7938  surface_classify_q4_arc_lifting_two_lines:
7939  ▶ $(ORBITER_PATH)orbiter.out-v.10.txt
7940  ▶ ▶  -define:F:-finite_field:-q4:-end.txt
7941  ▶ ▶  -define:P:-projective_space:3:F:-end.txt
7942  ▶ ▶  -with:P:-do\textbackslash n
7943  ▶ ▶  -projective_space_activity.txt
7944  ▶ ▶ ▶ -control_six_arcs:-problem_label:sixarcs_q4:-end.txt
7945  ▶ ▶ ▶ -classify_surfaces_through_arcs_and_two_lines.txt
7946  ▶ ▶  -end
7947  ▶  pdflatex:surfaces_arc_lifting_4.tex
7948  ▶  open:surfaces_arc_lifting_4.pdf
7949
7950
7951
7952
7953  surface_classify_q7:
7954  ▶ $(ORBITER_PATH)orbiter.out-v.5.txt
7955  ▶ ▶  -define:F:-finite_field:-q7:-end.txt
7956  ▶ ▶  -define:P:-projective_space:3:F:-end.txt
7957  ▶ ▶  -with:P:-do\textbackslash n
7958  ▶ ▶  -projective_space_activity.txt
7959  ▶ ▶ ▶ -classify_surfaces_with_double_sixes:Surf27:-W:-end.txt
7960  ▶ ▶  -end.txt
7961  ▶ ▶  -with:Surf27:-do\textbackslash n
7962  ▶ ▶ ▶ -classification_of_cubic_surfaces_with_double_sixes_activity.txt
7963  ▶ ▶ ▶  -report:end.txt
7964  ▶ ▶  -end.txt
7965  ▶ ▶  -print_symbols
7966  ▶  pdflatex:Surfaces_q7.tex
7967  ▶  open:Surfaces_q7.pdf
7968
7969
7970  surface_classify_q13:
7971  ▶ $(ORBITER_PATH)orbiter.out-v.5.txt
7972  ▶ ▶  -define:F:-finite_field:-q13:-end.txt
7973  ▶ ▶  -define:P:-projective_space:3:F:-end.txt
7974 \> \> -with P-do-\-
7975 \> \> -projective_space_activity-\-
7976 \> \> \> -classify_surfaces_with_double_sixes-C-W-end-\-
7977 \> \> \> -end-\-
7978 \> \> \> -with C-do-\-
7979 \> \> \> -classification_of_cubic_surfaces_with_double_sixes_activity-\-
7980 \> \> \> -report-end-\-
7981 \> \> \> -end-\-
7982 \> \> \> -print_symbols
7983 \> \> \> pdflatex Surfaces_q13.tex
7984 \> \> \> open Surfaces_q13.pdf
7985
7986
7987
7988 # Section 7.4: Cubic Surfaces -- Isomorphism Testing and Recognition
7989
7990
7991
7992 SECTION_CUBIC_SURFACES_ISOMORPHISM_TESTING_AND_RECOGNITION:
7993
7994
7995 surface_recognize_q7_abcd_2.3.3.4:
7996 \> $(ORBITER_PATH)orbiner.out-v.3-\-
7997 \> \> -define F-\>finite_field-\>q.7-end-\-
7998 \> \> -define P-\>projective_space-3F-end-\-
7999 \> \> -with P-do-\-
8000 \> \> -projective_space_activity-\-
8001 \> \> \> -classify_surfaces_with_double_sixes-Surf-\>W-end-\-
8002 \> \> \> -end-\-
8003 \> \> \> -with Surf-do-\-
8004 \> \> \> \> -classification_of_cubic_surfaces_with_double_sixes_activity-\-
8005 \> \> \> \> -recognize-\-
8006 \> \> \> \> \> \> -q.7-\-
8007 \> \> \> \> \> \> \> -family_general_abcd_2.3.3.4-\-
8008 \> \> \> \> \> \> \> -end-\-
8009 \> \> \> \> \> \> \> -end-\-
8010 \> \> \> \> \> \> \> -end
8011
8012
8013 surface_isomorph_16:
8014 \> $(ORBITER_PATH)orbiner.out-v.3-\-
8015 \> \> -define F-\>finite_field-\>q.16-end-\-
8016 \> \> -define P-\>projective_space-3F-end-\-
8017 \> \> -with P-do-\-
8018 \> \> -projective_space_activity-\-
8019 \> \> \> -classify_surfaces_with_double_sixes-Surf27-\>W-end-\-
8020 \> \> \> -end-\-
- classification of cubic surfaces with double sixes activity

- isomorphism testing

- q_{16}-by_coefficients

"1,5,1,8,1,11,1,12,6,14,6,15,7,18,7,19"-end

- q_{16}-by_coefficients

"13,6,3,8,3,11,13,13,1,19"-end

-end

print_symbols

#1-min:8-sec on Mac from scratch (with all data files removed)

surface_recognize_8:

$(ORBITER_PATH)orbiter.out -v:3

-define:F -finite_field -q:8-end

-define:P -projective_space 3F-end

-with:P-do

-projective_space_activity

-classify_surfaces_with_double_sixes Surf27 -W-end

-end

-with Surf27 -do

-classification_of_cubic_surfaces_with_double_sixes_activity

-recognize

- q:8

-by_coefficients: "1,6,1,8,1,11,1,13,1,19"

-end

-end

-print_symbols

surface_recognize_F13.q4:

$(ORBITER_PATH)orbiter.out -v:3

-define:F -finite_field -q:4-end

-define:P -projective_space 3F-end

-with:P-do

-projective_space_activity

-classify_surfaces_with_double_sixes Surf27 -W-end

-end

-with Surf27 -do

-classification_of_cubic_surfaces_with_double_sixes_activity

-identify F13
section 7.5: cubic surfaces of dickson type
SECTION_CUBIC_SURFACES_DICKSON:

D6_q2:

```bash
$ (ORBITER_PATH) orbiter.out -v 3

define F: finite_field -q 2 -end
define P: projective_space 3 F -end
with P -do

projective_space activity

define surface S_D6_q2 -q 2 -by_coefficients $(D6) -end
with S_D6_q2 -do

cubic_surface_activity

report

end

pdflatex surface_by_coefficients_q2_report.tex
open surface_by_coefficients_q2_report.pdf
mv surface_by_coefficients_q2_points.txt surface_by_coefficients_q2_D6_points.txt
```

D3_q4:

```bash
$ (ORBITER_PATH) orbiter.out -v 3

define F: finite_field -q 4 -end
define P: projective_space 3 F -end
with P -do

projective_space activity

define surface S_D3_q4 -q 4 -by_coefficients $(D3) -end
with S_D3_q4 -do

cubic_surface_activity

report

end

pdflatex surface_by_coefficients_q4_report.tex
open surface_by_coefficients_q4_report.pdf
mv surface_by_coefficients_q4_points.txt surface_by_coefficients_q4_D3_points.txt
```

# surface_by_coefficients_q4_points.txt
\begin{verbatim}
D4_q8:
$\$(ORBITER_PATH) orbiter.out -v 3 \$
\-define F -finite_field -q 8 -end \-
\-define P -projective_space 3 F -end \-
\-with P -do \-
\-projective_space_activity \-
\-define_surface S_D4_q8 -q 8 -by_coefficients $(D4) -end \-
\-end \-
\-with S_D4_q8 -do \-
\-cubic_surface_activity \-
\-report \-
\-end \-
\pdflatex surface_by_coefficients_q8_report.tex
\open_surface_by_coefficients_q8_report.pdf
\mv surface_by_coefficients_q8_points.txt surface_by_coefficients_q8_D4_points.txt

D6_q4:
$\$(ORBITER_PATH) orbiter.out -v 3 \$
\-define F -finite_field -q 4 -end \-
\-define P -projective_space 3 F -end \-
\-with P -do \-
\-projective_space_activity \-
\-define_surface S_D6_q4 -q 4 -by_coefficients $(D6) -end \-
\-end \-
\-with S_D6_q4 -do \-
\-cubic_surface_activity \-
\-report \-
\-end \-
\pdflatex surface_by_coefficients_q4_report.tex
\open_surface_by_coefficients_q4_report.pdf
\mv surface_by_coefficients_q4_points.txt surface_by_coefficients_q4_D6_points.txt

# D6 has 7 lines over GF(4)
\end{verbatim}
D8_q4:

```bash
$ (ORBITER_PATH)orbiter.out -v.3 \n```

```bash
define F finite_field q.4 -end \n```

```bash
define P projective_space 3 F -end \n```

```bash
with P -do \n```

```bash
projective_space_activity \n```

```bash
define_surface S D8 q.4 -q.4 -by_coefficients $(D8) -end \n```

```bash
end \n```

```bash
with S D8 q.4 -do \n```

```bash
-cubic_surface_activity \n```

```bash
report \n```

```bash
end \n```

```bash
pdflatex surface_by_coefficients_q4_report.tex \n```

```bash
open_surface_by_coefficients_q4_report.pdf \n```

```bash
mv surface_by_coefficients_q4_points.txt surface_by_coefficients_q4_D8_points.txt \n```

D1_q8:

```bash
$ (ORBITER_PATH)orbiter.out -v.3 \n```

```bash
define F finite_field q.8 -end \n```

```bash
define P projective_space 3 F -end \n```

```bash
with P -do \n```

```bash
projective_space_activity \n```

```bash
define_surface S D1_q8 -q.8 -by_coefficients $(D1) -end \n```

```bash
end \n```

```bash
with S D1_q8 -do \n```

```bash
-cubic_surface_activity \n```

```bash
report \n```

```bash
end \n```

```bash
pdflatex surface_by_coefficients_q8_report.tex \n```

```bash
open_surface_by_coefficients_q8_report.pdf \n```

```bash
mv surface_by_coefficients_q8_points.txt \n```

#surface_by_coefficients_q8_points.txt

```bash
## cleaning D1 with 15 lines over F2 and 27 lines over F4: \n```

```bash
D1_q4 with select double six: \n```

```bash
$ (ORBITER_PATH)orbiter.out -v.3 \n```

```bash
define F finite_field q.4 -end \n```

```bash
define P projective_space 3 F -end \n```

```bash
with P -do \n```

```bash
projective_space_activity \n```

```bash
define_surface S D1_q4 -q.4 -by_coefficients $(D1) \n```

629
-select_double_six "3,9,15,19,22,26,4,10,14,18,21,25".
-end-
-cubic_surface_activity-
-report-
-end
mv:surface_by_coefficients_q4_report.tex-D1_q4.tex
pdflatex D1_q4.tex
open D1_q4.pdf

D1_q4_with_select_double_six_b:
$(ORBITER_PATH)orbiter.out-v:3-
defineF:-finite_field-q:4-end-
defineP:-projective_space-3F-end-
-withP:-do-
-projective_space_activity-

define_surface_S_D1_q4:-q:4-by_coefficients $(D1)-
-select_double_six "3,9,15,19,22,26,4,10,14,18,21,25"-
-select_double_six "1,2,3,4,5,0,7,8,9,10,11,6"-
-end
-withS_D1_q4:-do-
-cubic_surface_activity-
-report-
-end
mv:surface_by_coefficients_q4_report.tex-D1_q4.tex
pdflatex D1_q4.tex
open D1_q4.pdf

#ToDo: now_projective_space_activity:
D1_q4_trans:
$(ORBITER_PATH)orbiter.out-v:5:-defineF:-finite_field-q:4-end-
-withF:-do:-finite_field_activity-
-move_two_lines_in_hyperplane_stabilizer_text-
"1,0,0,0,0,0,1,1":"0,1,1,1,1,0,1,0"
"1,0,0,0,0,0,0,1":"0,1,0,1,1,0,1,0"
-end
D1_q4_with_select_double_six_c:
$(ORBITER_PATH)orbiter.out-v:3-
defineF:-finite_field-q:4-end-
defineP:-projective_space-3F-end-
-withP:-do-
-projective_space_activity\"  
-define_surface-S_D1_q4-q4-by_coefficients-D1\"  
-select_double_six\"3,9,15,19,22,26,4,10,14,18,21,25\"  
-select_double_six\"1,2,3,4,5,0,7,8,9,10,11,6\"  
-transform\"1,0,0,0,0,1,0,0,0,1,0,0,1,1,0\"  
-end\"  
-end\"  
-select_double_six\"1,2,3,4,5,0,7,8,9,10,11,6\"  
-transform\"1,0,0,0,0,1,0,0,0,1,0,0,1,1,0\"  
-end\"  
-end\"  
-with-S_D1_q4-do-\"  
cubic_surface_activity\"  
-report\"  
-end\"  
-mv-surface_by_coefficients_q4_report.tex-D1_q4.tex  
pdflatex-D1_q4.tex  
open-D1_q4.pdf  
  
orbits_cubic_surfaces_q3:\  
$\$(ORBITER_PATH)orbiter.out--v.4\$  
-define-G-linear_group-PGL.4.3-end\"  
-with-G-do\"  
cubic_surface_activity\"  
-report\"  
-end\"  
-pdflatex-poly_orbits_d3_n3_q3.tex  
open-poly_orbits_d3_n3_q3.pdf  
  
#this_takes.3:days_and_about.150:GB:memory_on_ripoff  
orbits_cubic_curves_q2_again:\  
$\$(ORBITER_PATH)orbiter.out--v.4\$  
-define-G\"  
-linear_group-PGL.3.2\"  
-end\"  
-with-G-do\"  
cubic_surface_activity\"  
-orbits_on_polynomials-3\"  
-end\"  
pdflatex-poly_orbits_d3_n2_q2.tex  
open-poly_orbits_d3_n2_q2.pdf  
orbits_cubic_curves_q3:\  
$\$(ORBITER_PATH)orbiter.out--v.4\$
\begin{verbatim}
8345 \>>> -define G:\n8346 \>>> -linear_group-PGL.3.3:\n8347 \>>> -end:\n8348 \>>> -with G-do:\n8349 \>>> -group_theoretic_activity:\n8350 \>>> -orbits_on_polynomials.3:\n8351 \>>> -end:\n8352 \>>> pdflatex-poly_orbits_d3_n2_q3.tex
8353 \>>> open-poly_orbits_d3_n2_q3.pdf
8354
8355
8356
8357 \# compute and analyze properties over F2
8358
8359 poly_orbits_d3_n3_q2_F2.csv: poly_orbits_d3_n3_q2.csv
8360 \>>> $(ORBITER_PATH) orbiter.out -v 4:\n8361 \>>> \-define F\-finite_field\-q\-2\-end:\n8362 \>>> \-define P\-projective_space\-3\-F\-end:\n8363 \>>> \-with P\-do:\n8364 \>>> \-projective_space_activity:\n8365 \>>> \-table of cubic surfaces compute properties:\n8366 \>>> poly_orbits_d3_n3_q2.csv: 2.0:\n8367 \>>> -end:\n8368
8369
8370 Dickson_q2_analyze: poly_orbits_d3_n3_q2_F2.csv
8371 \>>> $(ORBITER_PATH) orbiter.out -v 4:\n8372 \>>> \-define F\-finite_field\-q\-2\-end:\n8373 \>>> \-define P\-projective_space\-3\-F\-end:\n8374 \>>> \-with P\-do:\n8375 \>>> \-projective_space_activity:\n8376 \>>> \-cubic_surface_properties_analyze:\n8377 \>>> poly_orbits_d3_n3_q2_F2.csv: 2.0:\n8378 \>>> -end:\n8379 \>>> pdflatex-poly_orbits_d3_n3_q2_F2_report.tex
8380 \>>> open-poly_orbits_d3_n3_q2_F2_report.pdf
8381
8382 \# compute and analyze properties over F4
8383
8384 poly_orbits_d3_n3_q2_F4.csv: poly_orbits_d3_n3_q2.csv
8385 \>>> $(ORBITER_PATH) orbiter.out -v 4:\n8386 \>>> \-define F\-finite_field\-q\-4\-end:\n8387 \>>> \-define P\-projective_space\-3\-F\-end:\n8388 \>>> \-with P\-do:\n8389 \>>> \-projective_space_activity:\n8390 \>>> \-table of cubic surfaces compute properties:\n8391 \>>> poly_orbits_d3_n3_q2.csv: 2.0:\n\end{verbatim}
Dickson_q4_analyze::poly_orbits_d3_n3_q2_F4.csv

Dickson_q8_analyze::poly_orbits_d3_n3_q2_F8.csv

Dickson_q16_analyze::poly_orbits_d3_n3_q2_F16.csv

# compute and analyze properties over F8

cubic_surface_properties_analyze

table_of_cubic_surfaces_compute_properties

cubic_surface_properties_analyze

# compute and analyze properties over F16

Cubic_surface_properties_analyze

table_of_cubic_surfaces_compute_properties

cubic_surface_properties_analyze
Dickson_q16.analyze::poly_orbits_d3_n3_q2_F16.csv
$(ORBITER\_PATH)\texttt{orbiter.out}\_v\_4$
$\define F\_\text{finite\_field}\_q\_16\_end$
$\define P\_\text{projective\_space}\_3\_F\_\text{end}$
$\text{with}\_P\_\text{do}$
$\text{projective\_space\_activity}$
$\text{cubic\_surface\_properties\_analyze}$
$\text{poly\_orbits\_d3\_n3\_q2\_F16.csv\_2}$
$\text{end}$
dflatex poly_orbits_d3_n3_q2_F16_report.tex
open\_poly\_orbits\_d3\_n3\_q2\_F16\_report.pdf
}

SECTION\_CUBIC\_SURFACES\_ATLAS\_AND\_TABLES:

MAKE\_TABLE\_OF\_CUBIC\_SURFACES\(_\text{define}\_P\_\text{projective\_space}\_3\_F\_\text{end})$

$\text{with}\_P\_\text{do}$
$\text{projective\_space\_activity}$
$\text{table\_of\_cubic\_surfaces}$
$\text{end}$

cubic\_surfaces\_tables\_17:
$(ORBITER\_PATH)\texttt{orbiter.out}\_v\_3$
$\define F\_\text{finite\_field}\_q\_17\_end$
$\text{MAKE}\_\text{TABLE}\_OF\_CUBIC\_SURFACES$

cubic\_surfaces\_table\_latex\_17$
$(ORBITER\_PATH)\texttt{orbiter.out}\_v\_3\_\text{csv\_file}\_latex\_1$
$\text{table\_of\_cubic\_surfaces}\_q17\_info.csv$

cubic\_surfaces\_tables\_up\_to\_17:
$(ORBITER\_PATH)\texttt{orbiter.out}\_v\_3\_\text{define}\_F\_\text{finite\_field}\_q\_4\_\text{end}$\text{MAKE}\_\text{TABLE}\_OF\_CUBIC\_SURFACES$
$(ORBITER\_PATH)\texttt{orbiter.out}\_v\_3\_\text{define}\_F\_\text{finite\_field}\_q\_7\_\text{end}$(MAKE\_TABLE\_OF\_CUBIC\_SURFACES)$
$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

$(MAKE

635
cubic_surfaces_tables_latex:.

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.61\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.64\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.67\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.71\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.73\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.79\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.81\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.83\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.89\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.97\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.101\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.103\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.107\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.109\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.113\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.121\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.127\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)

$(ORBITER_PATH)orbi\textunderscore{}ter.out-v.3\textunderscore{}defineF\textunderscore{}finite\textunderscore{}field\textunderscore{}q.128\textunderscore{}end$(MAKE\textunderscore{}TABLE\_OF\textunderscore{}CUBIC\textunderscore{}SURFACES)
fo.csv
8537 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q11\_info.csv
8538 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q13\_info.csv
8539 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q16\_info.csv
8540 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q17\_info.csv
8541 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q19\_info.csv
8542 cubic_surfaces.tables.latex.big:
8543 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q23\_info.csv
8544 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q25\_info.csv
8545 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q27\_info.csv
8546 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q29\_info.csv
8547 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q31\_info.csv
8548 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q32\_info.csv
8549 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q37\_info.csv
8550 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q41\_info.csv
8551 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q43\_info.csv
8552 $(ORBITER\_PATH)\$ (v.3 - csv_file_latex-0\_table_of\_cubic\_surfaces\_q9\_info.csv
8553 #$(ORBITER\_PATH)\$ (v.3 - csv_file_latex-1\_quartic\_curves\_q9\_info.csv
8554 pdflatex-quartic_curves_q13\_info.tex
8555 open-quartic_curves_q13\_info.pdf
8556 ~/bin/tth-quartic_curves_q13\_info.tex
8557 open-quartic_curves_q13\_info.html
8558 surface_table:
8559 $(ORBITER\_PATH)\$ (v.3 - make_table_of\_surfaces
8560 pdflatex-surfaces_report.tex
8561 open-surfaces_report.pdf
8562
8563
8564
8565
8566
8567
8568

637
surface_atlas:
- $(ORBITER\ PATH)\ orbiter.out\ -v\ 3\ -create\ surface\ atlas\ 97
- ~/bin/tth\ surface\ atlas.tex

surface_reports:
- $(ORBITER\ PATH)\ orbiter.out\ -v\ 3\ -create\ surface\ reports\ 4,7,8,9,11

Surface_Orb0_q4:
- $(ORBITER\ PATH)\ orbiter.out\ -v\ 3\ -linear\ group\ -PGL\ 4\ -wedge\ -end\ -group\ theoretic\ activity\ -control\ six\ arcs\ -end\ -define\ surface\ -label\ txt\ "Orb0\ q4"\ -label\ tex\ "Rank-$(Orb0)"\ -by_rank\ $(Orb0)\ -q\ 2\ -q\ 4\ -end\ -end

create_dickson_makefiles_q2:
- $(ORBITER\ PATH)\ orbiter.out\ -create\ files\ -file_mask\ dickson\ q2\ %03d\ -N\ 141\ -line_numeric\ "Surface_Orb%d_q2:"
- \"t\D(ORBITER\ PATH)/orbiter.out\ -v\ 3\ \"\ -line_numeric\ "t\t-linear\ group\ -PGL\ 4\ -wedge\ -end\ \"\ -line_numeric\ "t\t-group\ theoretic\ activity\ \"\ -line_numeric\ "t\t-control\ six\ arcs\ -end\ \"\ -line_numeric\ "t\t-define\ surface\ -label\ txt\ "Orb%d_q2\"\ -label\ tex\ "Rank-\D(Orb%d)\ -by_rank\ \D(Orb%d)\ -q\ 2\ -q\ 2\ -end\ \"\ -line_numeric\ "t\t-end\ \"\ | 638
8615 ▶ ▶ -line_numeric:\"t\t-end\\n\n8616 ▶ ▶ -line_numeric:\"\n\n8617 ▶ ▶ -end
8618 ▶ cat dickson_q2.*->dickson_q2
8619
8620 create_dickson_makefiles_q4:
8622 ▶ $(ORBITER\PATH)\orbiter.out-\>create_files-\>file_mask-dickson_q4_%03d \n8623 ▶ ▶ -N:141\n8624 ▶ ▶ -line_numeric:\"Surface_Orb%d_q4:\"\n8625 ▶ ▶ -line_numeric:\"\t\D(ORBITER\PATH)/orbiter.out-v.3\\n\n8626 ▶ ▶ -line_numeric:\"\t\-linear_group-PGL.4.4-wedge-end\\n\n8627 ▶ ▶ -line_numeric:\"\t\-group_theoretic_activity\\n\n8628 ▶ ▶ -line_numeric:\"\t\-control_six_arcs-end\\n\n8629 ▶ ▶ -line_numeric:\"\t\-define_surface-label_txt\"Orb%d_q4\\n\n8630 ▶ ▶ ▶ -label_txt\"Rank-\D(Orb%d\\n\n8631 ▶ ▶ ▶ -label_for_summary\"Orb%d\"-by_rank\\D(Orb%d.2-q.4\\n\n8632 ▶ ▶ -line_numeric:\"\t\t-end\\n\n8633 ▶ ▶ -line_numeric:\"\t\t-end\\n\n8634 ▶ ▶ -line_numeric:\"\\n\n8635 ▶ ▶ -end
8636 ▶ cat dickson_q4.*->q8
8637
8638 create_dickson_makefiles_q8:
8639 ▶ $(ORBITER\PATH)\orbiter.out-\>create_files-\>file_mask-dickson_q8_%03d \n8640 ▶ ▶ -N:141\n8641 ▶ ▶ -line_numeric:\"Surface_Orb%d_q8:\"\n8642 ▶ ▶ -line_numeric:\"\t\D(ORBITER\PATH)/orbiter.out-v.3\\n\n8643 ▶ ▶ -line_numeric:\"\t\-linear_group-PGL.4.8-wedge-end\\n\n8644 ▶ ▶ -line_numeric:\"\t\-group_theoretic_activity\\n\n8645 ▶ ▶ -line_numeric:\"\t\-control_six_arcs-end\\n\n8646 ▶ ▶ -line_numeric:\"\t\-define_surface-label_txt\"Orb%d_q8\\n\n8647 ▶ ▶ ▶ -label_txt\"Rank-\D(Orb%d\\n\n8648 ▶ ▶ ▶ -label_for_summary\"Orb%d\"-by_rank\\D(Orb%d.2-q.8\\n\n8649 ▶ ▶ -line_numeric:\"\t\t-end\\n\n8650 ▶ ▶ -line_numeric:\"\\n\n8651 ▶ ▶ -end
8652 ▶ cat dickson_q8.*->q8
8653
8654 create_dickson_makefiles_q16:
8655 ▶ $(ORBITER\PATH)\orbiter.out-\>create_files-\>file_mask-dickson_q16_%03d \n8656 ▶ ▶ -N:141\n8657 ▶ ▶ -line_numeric:\"Surface_Orb%d_q16:\"\n8658 ▶ ▶ -line_numeric:\"\t\D(ORBITER\PATH)/orbiter.out-v.3\\n\n8659 ▶ ▶ -line_numeric:\"\t\-linear_group-PGL.4.16-wedge-end\\n\n8660 ▶ ▶ -line_numeric:\"\t\-group_theoretic_activity\\n\n8661 ▶ ▶ -line_numeric:\"\t\-control_six_arcs-end\\n
639
create_dickson_makefiles_q32:
$(ORBITER_PATH)orbiter.out--create_files--file_mask-dickson_q32_%03d\n-N:141.\n
create_dickson_makefiles_q64:
$(ORBITER_PATH)orbiter.out--create_files--file_mask-dickson_q64_%03d\n-N:141.\n
create_dickson_makefiles_top_line_q2:
$(ORBITER_PATH)orbiter.out--create_files--file_mask-dickson_q2_top\
-N:1-repeat:141:0:1."Surface_q2:\"--command:\"tSurface_Orb%d_q2:\".\n
end
create_dickson_makefiles_top_line_q4:

create_dickson_makefiles_top_line_q8:

create_dickson_makefiles_top_line_q16:

create_dickson_makefiles_top_line_q32:

create_dickson_makefiles_top_line_q64:

create_dickson_makefiles_join_q2:

create_dickson_makefiles_join_q4:
8756 dickson_q2_latex:
8757 $\text{ORBITER\ PATH}/\text{orbiter.out}-\text{csv\ file\ latex-q2.csv}$
8758
8759
8760
8761 #problem:
8762 #orthogonal::lines_on_point_by_line_rank_i=1366//4225-pt=15447347-pt2=15225451
8763 #orthogonal::lines_on_point_by_line_rank_before_rank_line
8764 #orthogonal::rank_line_p1=15447347\ p2=15225451
8765
8766 create_dickson_atlas:
8767 $\text{ORBITER\ PATH}/\text{orbiter.out}-\text{create_dickson_atlas}$
8768 $\text{/bin/tth\ dickson_surfaces.tex}$
8769
8770
8771 quartic_curve_tables_23::
8772 $\text{ORBITER\ PATH}/\text{orbiter.out}-v.3$
8773 $\text{define}\ F\text{-finite_field}\ q.23\text{-end}$
8774 $\text{define}\ P\text{-projective_space}\ 2\text{-end}$
8775 $\text{with}\ P\text{-do}$
8776 $\text{-projective_space_activity}$
8777 $\text{-table_of_quartic_curves}$
8778 $\text{-end}$
8779
8780 quartic_curve_tables::
8781 $\text{ORBITER\ PATH}/\text{orbiter.out}-v.3$
8782 $\text{define}\ F\text{-finite_field}\ q.9\text{-end}$
8783 $\text{define}\ P\text{-projective_space}\ 2\text{-end}$
8784 $\text{with}\ P\text{-do}$
8785 $\text{-projective_space_activity}$
8786 $\text{-table_of_quartic_curves}$
8787 $\text{-end}$
8788
8789 $\text{ORBITER\ PATH}/\text{orbiter.out}-v.3$
8790 $\text{define}\ F\text{-finite_field}\ q.13\text{-end}$
8791 $\text{define}\ P\text{-projective_space}\ 2\text{-end}$
8792 $\text{with}\ P\text{-do}$
8793 $\text{-projective_space_activity}$
8794 $\text{-table_of_quartic_curves}$
8795 $\text{-end}$
8796 $\text{ORBITER\ PATH}/\text{orbiter.out}-v.3$
8797 $\text{define}\ F\text{-finite_field}\ q.17\text{-end}$
8798 $\text{define}\ P\text{-projective_space}\ 2\text{-end}$
8799 $\text{with}\ P\text{-do}$
8800 $\text{-projective_space_activity}$
8801 $\text{-table_of_quartic_curves}$
8802 $\text{-end}$
quartic_curve_tables_latex: 
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.1-test.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q9_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q13_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q17_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q19_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q25_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q27_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q29_info.csv
> $(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.0-quartic_curves_q31_info.csv
> #$(ORBITER_PATH) orbiter.out --v.3 -csv_file_latex.1-quartic_curves_q9_info.csv
> #pdf_latex-quartic_curves_q13_info.tex
> #open-quartic_curves_q13_info.pdf
> #~/bin/tth-quartic_curves_q13_info.tex
> #open-quartic_curves_q13_info.html

# Chapter 8: Ring Theory
# Section 8.1: Polynomials over Finite Fields
SECTION_POLYNOMIALS:
# check which polynomials are irreducible and which are primitive:
sift_polynomials_deg3.q2:
> $(ORBITER_PATH) orbiter.out --v.2 \
> > -define F=finite_field -q 2 -end \
> > -with F=do \
> > -finite_field_activity -sift_polynomials.8.16 -end
sift_polynomials_deg4.q2:
$\langle$ORBITER_PATH$\rangle orbiter.out -v.2$
\>$-define F-finite_field-q.2-end$
\>$-with F-do$
$\>$-finite_field_activity-sift_polynomials.16:32-end

poly_division:
$\langle$ORBITER_PATH$\rangle orbiter.out -v.2$
\>$-define F-finite_field-q.2-end$
\>$-with F-do$
\>$-finite_field_activity$
\>$-polynomial_division"1,0,0,0,0,0,0,0,0,1"."1,0,1,1"-end

poly_division2:
$\langle$ORBITER_PATH$\rangle orbiter.out -v.2$
\>$-define F-finite_field-q.2-end$
\>$-with F-do$
\>$-finite_field_activity$
\>$-polynomial_division A B -end

poly_gcd:
$\langle$ORBITER_PATH$\rangle orbiter.out -v.2$
\>$-define F-finite_field-q.2-end$
\>$-with F-do$
\>$-finite_field_activity$
\>$-extended_gcd_for_polynomials"1,0,0,0,0,0,0,0,0,1"."1,0,1,1"-end

poly_mult_mod1:
$\langle$ORBITER_PATH$\rangle orbiter.out -v.2$
\>$-define F-finite_field-q.7-end$
\>$-with F-do$
\>$-finite_field_activity$
\>$-polynomial_mult_mod"1,2,3"."3,4,5"."6,0,0,1"-end

poly_mult_mod2:
$\langle$ORBITER_PATH$\rangle orbiter.out -v.2$
\>$-define F-finite_field-q.7-end$
\>$-with F-do$
\>$-finite_field_activity$

644
8897 \triangleright - polynomial\_mult\_mod"3,1,2"."5,3,4"."6,0,0,1".-end
8898
8899
8900
8901 poly\_mult\_F4:
8902 \triangleright $(\text{ORBITER PATH})\text{or}\text{biter.out}\text{-v\_2}\
8903 \triangleright \triangleright \text{-define F-\text{finite field}-q2-end}\
8904 \triangleright \triangleright \text{-with F-do}\
8905 \triangleright \triangleright \text{-finite field activity}\
8906 \triangleright \triangleright \text{-polynomial\_mult\_mod"1,1"."1,1"."1,1".-end}\
8907 \triangleright $(\text{ORBITER PATH})\text{or}\text{biter.out}\text{-v\_2}\
8908 \triangleright \triangleright \text{-define F-\text{finite field}-q2-end}\
8909 \triangleright \triangleright \text{-with F-do}\
8910 \triangleright \triangleright \text{-finite field activity}\
8911 \triangleright \triangleright \text{-polynomial\_mult\_mod"0,1"."1,1"."1,1".-end}\
8912 \triangleright $(\text{ORBITER PATH})\text{or}\text{biter.out}\text{-v\_2}\
8913 \triangleright \triangleright \text{-define F-\text{finite field}-q2-end}\
8914 \triangleright \triangleright \text{-with F-do}\
8915 \triangleright \triangleright \text{-finite field activity}\
8916 \triangleright \triangleright \text{-polynomial\_mult\_mod"0,1"."0,1"."1,1".-end}\
8917
8918
8919
8920
8921
8922 mult\_polynomials\_2.5_7:
8923 \triangleright $(\text{ORBITER PATH})\text{or}\text{biter.out}\text{-v\_2}\
8924 \triangleright \triangleright \text{-define F-\text{finite field}-q2-end}\
8925 \triangleright \triangleright \text{-with F-do}\
8926 \triangleright \triangleright \text{-finite field activity}-\text{mult\_polynomials\_5_7-end}\
8927 \triangleright \text{pdflatex-polynomial\_mult\_5_7.tex}\
8928 \triangleright \text{open-polynomial\_mult\_5_7.pdf}\
8929
8930 polynomial\_division\_ranked\_2.27.13:
8931 \triangleright $(\text{ORBITER PATH})\text{or}\text{biter.out}\text{-v\_2}\
8932 \triangleright \triangleright \text{-define F-\text{finite field}-q2-end}\
8933 \triangleright \triangleright \text{-with F-do}\
8934 \triangleright \triangleright \text{-finite field activity}\
8935 \triangleright \triangleright \triangleright \text{-polynomial\_division\_ranked\_27.13}\
8936 \triangleright \triangleright \text{-end}\
8937 \triangleright \text{pdflatex-polynomial\_division\_27.13.tex}\
8938 \triangleright \text{open-polynomial\_division\_27.13.pdf}\
8939
8940
8941
8942
8943 mult\_polynomials\_2.8_15:
polynomial_division_ranked_2_120_25:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity\_mult\_polynomials\_8\_15\_end}\$
pdflatex\-polynomial\_mult\_8\_15.tex
open\-polynomial\_mult\_8\_15.pdf

#the-answer-is:5

mult\_polynomials\_2\_7\_7:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
mult\_polynomials\_7\_7\end
pdflatex\-polynomial\_mult\_7\_7.tex
open\-polynomial\_mult\_7\_7.pdf

mult\_polynomials\_2\_4\_6:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
mult\_polynomials\_4\_6\end
pdflatex\-polynomial\_mult\_4\_6.tex
open\-polynomial\_mult\_4\_6.pdf

polynomial\_division\_ranked\_2\_4\_6\_13:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
pol\_division\_ranked\_2\_4\_6\_13\end

#the-answer-is:5

mult\_polynomials\_2\_7\_7:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
mult\_polynomials\_7\_7\end
pdflatex\-polynomial\_mult\_7\_7.tex
open\-polynomial\_mult\_7\_7.pdf

mult\_polynomials\_2\_4\_6:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
mult\_polynomials\_4\_6\end
pdflatex\-polynomial\_mult\_4\_6.tex
open\-polynomial\_mult\_4\_6.pdf

polynomial\_division\_ranked\_2\_4\_6\_13:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
pol\_division\_ranked\_2\_4\_6\_13\end

#the-answer-is:5

d mult\_polynomials\_7\_7:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
mult\_polynomials\_7\_7\end
pdflatex\-polynomial\_mult\_7\_7.tex
open\-polynomial\_mult\_7\_7.pdf

mult\_polynomials\_4\_6:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
mult\_polynomials\_4\_6\end
pdflatex\-polynomial\_mult\_4\_6.tex
open\-polynomial\_mult\_4\_6.pdf

polynomial\_division\_ranked\_2\_4\_6\_13:
$\$(\texttt{ORBITER\_PATH})\texttt{orbiter.out}\ -v\cdot 2\$
$\$\texttt{-define F=finite_field\_q=2\_end}\$
$\$\texttt{-with F=do}\$
$\$\texttt{-finite_field\_activity}\$
pol\_division\_ranked\_2\_4\_6\_13\end
8996 mult_polynomials_1024_999_997:
8997 $(\text{ORBITER PATH})$orbiter.out-\-v.2\$
8998 \triangleright -\text{define F- finite field- q.2- end}\$
8999 \triangleright -\text{with F- do}\$
9000 \triangleright -\text{finite field activity}\$
9001 \triangleright \triangleright -\text{mult polynomials 999-997}\$
9002 \triangleright \triangleright -\text{end}
9003 \triangleright \text{pdflatex polynomial_mult_999_997.tex}
9004 \triangleright \text{open polynomial_mult_999_997.pdf}
9005
9006
9007 polynomial_division_ranked_2_349147_1033:
9008 $(\text{ORBITER PATH})$orbiter.out-\-v.2\$
9009 \triangleright -\text{define F- finite field- q.2- end}\$
9010 \triangleright -\text{with F- do}\$
9011 \triangleright -\text{finite field activity}\$
9012 \triangleright \triangleright -\text{polynomial division ranked 349147-1033}\$
9013 \triangleright \triangleright -\text{end}
9014 \triangleright \text{pdflatex polynomial_division_349147_1033.tex}
9015 \triangleright \text{open polynomial_division_349147_1033.pdf}
9016
9017
9018
9019 mult_polynomials_1024_999_997_check:
9020 $(\text{ORBITER PATH})$orbiter.out-\-v.3\$
9021 \triangleright \triangleright -\text{define F- finite field- q.1024- end}\$
9022 \triangleright \triangleright -\text{with F- do}\$
9023 \triangleright \triangleright -\text{finite field activity- parse and evaluate}\$
9024 \triangleright \triangleright "test"."".""a*b".""a=999,b=997".-end
9025
9026 \# evaluates to 61
9027
9028
9029 mult_polynomials_17_12:
9030 $(\text{ORBITER PATH})$orbiter.out-\-v.2\$
9031 \triangleright \triangleright -\text{define F- finite field- q.2- end}\$
9032 \triangleright \triangleright -\text{with F- do}\$
9033 \triangleright \triangleright -\text{finite field activity}\$
9034 \triangleright \triangleright -\text{mult polynomials 17-12- end}
9035 \triangleright \text{pdflatex polynomial_mult_17_12.tex}
9036 \triangleright \text{open polynomial_mult_17_12.pdf}
9037 \triangleright
#gives 204

crc32:  

 Berlekamp matrix crc32:  

 #N.:=2^32-1.:=3.:=5.:=17.:=257.:=65537
 #N.:=1431655765
 #N.:=858993459
 #N.:=252645135
 #N.:=16711935
 #N.:=65535
 TWO_TO_THE_32_MINUS_2:=4294967294
\[ \text{Berlekamp\_matrix\_2.3:} \]

\[ \text{#the\_polynomial\_X^3+X+1\_is\_irreducible\_over\_GF(2)\_because\_the\_rank\_of\_the\_Berlekamp\_matrix\_is\_2.} \]

\[ \text{Berlekamp\_matrix\_2.4:} \]

\[ \text{#the\_polynomial\_X^3+X+1\_is\_irreducible\_over\_GF(2)\_because\_the\_rank\_of\_the\_Berlekamp\_matrix\_is\_2.} \]
define F -finite_field -q 2 -end \\
define v -vector -field F -dense "1,1,0,0,1" -end \\
with F -do \\
finites_field_activity \\
Berlekamp_matrix v -end

#define F -finite_field -q 4 -end \\
#define v -vector -field F -dense "1,3,0,1" -end \\
with F -do \\
finites_field_activity \\
Berlekamp_matrix v -end

#define F -finite_field -q 19 -end \\
#define v -vector -field F -dense "18,1,1" -end \\
with F -do \\
finites_field_activity \\
polynomial_find_roots v -end

#define F -finite_field -q 19 -end \\
#define v -vector -field F -dense "1,3,1" -end \\
with F -do \\
finites_field_activity \\
polynomial_find_roots v -end

#define F -finite_field -q 19 -end \\
#define v -vector -field F -dense "1,3,1" -end \\
with F -do \\
finites_field_activity \\
polynomial_find_roots v -end

#define F -finite_field -q 19 -end \\
#define v -vector -field F -dense "1,3,1" -end \\
with F -do \\
finites_field_activity \\
polynomial_find_roots v -end
find roots_d:
$$(ORBITER\_PATH)\text{orbiter.out}-v.2:\$

find roots_e:
$$(ORBITER\_PATH)\text{orbiter.out}-v.2:\$

roots_over_F2:
$$(ORBITER\_PATH)\text{orbiter.out}-v.2:\$

roots_over_F8:
$$(ORBITER\_PATH)\text{orbiter.out}-v.2:\$

#degree and then order of the field of coefficients:

irred_3_2:
9224  $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3:\$
9225  \triangleright  -\text{define}\,-\text{finite\_field}\,-q^2\,-\text{end}\,$
9226  \triangleright  -\text{with}\,-\text{do}\,$
9227  \triangleright  -\text{finite\_field\_activity}\,$
9228  \triangleright  -\text{make\_table\_of\_irreducible\_polynomials}\,3\,-\text{end}
9229  pdflatex\text{Irred}_q^2\text{d}_3.tex
9230  open\text{Irred}_q^2\text{d}_3.pdf
9231
9232
9233  irred\_4\_2:
9234  $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3:\$
9235  \triangleright  -\text{define}\,-\text{finite\_field}\,-q^2\,-\text{end}\,$
9236  \triangleright  -\text{with}\,-\text{do}\,$
9237  \triangleright  -\text{finite\_field\_activity}\,$
9238  \triangleright  -\text{make\_table\_of\_irreducible\_polynomials}\,4\,-\text{end}
9239  pdflatex\text{Irred}_q^2\text{d}_4.tex
9240  open\text{Irred}_q^2\text{d}_4.pdf
9241
9242  #\,3\text{-polys}
9243
9244  irred\_5\_2:
9245  $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3:\$
9246  \triangleright  -\text{define}\,-\text{finite\_field}\,-q^2\,-\text{end}\,$
9247  \triangleright  -\text{with}\,-\text{do}\,$
9248  \triangleright  -\text{finite\_field\_activity}\,$
9249  \triangleright  -\text{make\_table\_of\_irreducible\_polynomials}\,5\,-\text{end}
9250  pdflatex\text{Irred}_q^2\text{d}_5.tex
9251  open\text{Irred}_q^2\text{d}_5.pdf
9252
9253  #\,6\text{-polys}
9254
9255  irred\_6\_2:
9256  $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3:\$
9257  \triangleright  -\text{define}\,-\text{finite\_field}\,-q^2\,-\text{end}\,$
9258  \triangleright  -\text{with}\,-\text{do}\,$
9259  \triangleright  -\text{finite\_field\_activity}\,$
9260  \triangleright  -\text{make\_table\_of\_irreducible\_polynomials}\,6\,-\text{end}
9261  pdflatex\text{Irred}_q^2\text{d}_6.tex
9262  open\text{Irred}_q^2\text{d}_6.pdf
9263
9264  #\,9\text{-polys}
9265
9266  irred\_7\_2:
9267  $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.3:\$
9268  \triangleright  -\text{define}\,-\text{finite\_field}\,-q^2\,-\text{end}\,$
9269  \triangleright  -\text{with}\,-\text{do}\,$
9270  \triangleright  -\text{finite\_field\_activity}\,$
-make_table_of_irreducible_polynomials
pdflatex Irred_q2_d7.tex
open Irred_q2_d7.pdf

#18 polys

irred_8_2:
$(ORBITER_PATH)orbiter.out-v.3\$
-define F-\finite_field-q.2-end\$
-with F-do\$
-finite_field_activity\$
-make_table_of_irreducible_polynomials
pdflatex Irred_q2_d8.tex
open Irred_q2_d8.pdf

#30 polys

irred_9_2:
$(ORBITER_PATH)orbiter.out-v.3\$
-define F-\finite_field-q.2-end\$
-with F-do\$
-finite_field_activity\$
-make_table_of_irreducible_polynomials
pdflatex Irred_q2_d9.tex
open Irred_q2_d9.pdf

#56 polys

irred_10_2:
$(ORBITER_PATH)orbiter.out-v.3\$
-define F-\finite_field-q.2-end\$
-with F-do\$
-finite_field_activity\$
-make_table_of_irreducible_polynomials
pdflatex Irred_q2_d10.tex
open Irred_q2_d10.pdf

#99 polys

irred_2_4:
$(ORBITER_PATH)orbiter.out-v.3\$
-define F-\finite_field-q.4-end\$
-with F-do\$
-finite_field_activity\$
-make_table_of_irreducible_polynomials
pdflatex Irred_q4_d2.tex
open Irred_q4_d2.pdf
#6-polys

irred_3_4:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.6:\}$

\$\text{-define F\_finite_field\_q4\_end}$

\$\text{-with F\_do}$

\$\text{-finite_field\_activity}$

\$\text{-make_table_of_irreducible_polynomials\_3\_end}$

\$\text{pdflatex\_Irred\_q4\_d3.tex}$

\$\text{open\_Irred\_q4\_d3.pdf}$

#20-polys

searchPrimitivePoly_2:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.3:\}$

\$\text{-search_forprimitivepolynomial\_in\_range\_2\_2\_2\_10\_\#\_\_grep://}$

#stuck in factoring: $2^{61} - 1$ (which is prime)

searchPrimitivePoly_3:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.6\}$

\$\text{-search_forprimitivepolynomial\_in\_range\_3\_3\_2\_60}$

searchPrimitivePoly_4:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.6\}$

\$\text{-search_forprimitivepolynomial\_in\_range\_4\_4\_2\_30}$

searchPrimitivePoly_5:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.6\}$

\$\text{-search_forprimitivepolynomial\_in\_range\_5\_5\_2\_30}$

searchPrimitivePoly_7:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.6\}$

\$\text{-search_forprimitivepolynomial\_in\_range\_7\_7\_2\_20}$

searchPrimitivePoly_8:

\$\text{(ORBITER\_PATH)\texttt{orbiter.out}}\text{-v.6\}$
-search_for_primitive_polynomial_in_range 8·8·2·20

search_primitive_poly_9:
  $(ORBITER_PATH)orbiter.out --v 6

search_primitive_poly_11:
  $(ORBITER_PATH)orbiter.out --v 6

search_primitive_poly_13:
  $(ORBITER_PATH)orbiter.out --v 6

search_primitive_poly_degree_16:
  $(ORBITER_PATH)orbiter.out --v 6

search_primitive_poly_32:
  $(ORBITER_PATH)orbiter.out --v 6

SECTION IDEALS:

arcs_5_2_q11:
  $(ORBITER_PATH)orbiter.out --v 4
  --define F -finite_field -q 11 -end
  --define P -projective_space -2 F -end
  --with P -do
  --projective_space_activity
  --classify arcs
  --poset_classification_control
#2-orbits:
#0:1:2:3:37
#0:1:2:3:49

arcs_5_2_q11_ideal:
$\$(ORBITER_PATH)orbiter.out-v.2\$

-define-F-finite_field-q.11-end
-define-R-polynomial_ring
-field-F
-number_of_variables-3
-homogeneous_of_degree-2
-monomial_ordering_lex
-variables:"x0,x1,x2"."x_0,x_1,x_2"
-end

-define-C-combinatorial_objects
-file_of_points-arcs_5_2_q11_lvl_5
-end

-with-C-do
-combinatorial_object_activity
-ideal-R
-end

#(-0,1,2,3,37)
generator-0/-1-is:7*x0*x1+5*x0*x2+10*x1*x2
We-found:12-points-on-the-generator-of-the-ideal
They-are:(-0,1,2,3,37,54,74,80,93,105,121,128)

#(-0,1,2,3,49)
generator-0/-1-is:4*x0*x1+8*x0*x2+10*x1*x2
looping-over-all-generators-of-the-ideal:
generator-0/-1-is:(0,4,-8,0,10,0):
We-found:12-points-on-the-generator-of-the-ideal
They-are:(-0,1,2,3,41,49,58,77,83,95,109,130)
surface_9lines_4E_ideal:

#The ideal has dimension 2
#generators for the ideal:

F_9="x0*x0*x1-x0*x1*x1-x0*x1*x3-x2*x2*x3--x2*x3*x3"

F_9_q7:


we create 20.5-subsets of PG(2,11) at random. Note that PG(2,11) has 133 points.

random_k_subsets:

#random_k_subsets_n133_k5_nb20.csv

# We compute the line intersections:

line_type_in_PG_2_11:

random_arc_5_2_q11_ideal:

random_5_20_k5_n133.csv

# the second one is an arc: 3, 33, 40, 83, 102
\begin{verbatim}
9552 ▼ ▼ -define R-polynomial_ring\ 9553 ▼ ▼ ▼ -field F\ 9554 ▼ ▼ ▼ ▼ -number_of_variables 3\ 9555 ▼ ▼ ▼ ▼ -homogeneous_of_degree 2\ 9556 ▼ ▼ ▼ ▼ -monomial_ordering lex\ 9557 ▼ ▼ ▼ ▼ -variables "x0,x1,x2" x_0,x_1,x_2\ 9558 ▼ ▼ ▼ ▼ -end\ 9559 ▼ ▼ ▼ -define C-combinatorial_objects\ 9560 ▼ ▼ ▼ ▼ -set_of_points 3,33,40,83,102\ 9561 ▼ ▼ ▼ ▼ -end\ 9562 ▼ ▼ ▼ -with C-do\ 9563 ▼ ▼ ▼ ▼ -combinatorial_object_activity\ 9564 ▼ ▼ ▼ ▼ -ideal R\ 9565 ▼ ▼ ▼ ▼ -end\ 9566 #generator 0/1 is 10*x0*x0 + 3*x0*x1 + 8*x0*x2 + 2*x1*x1 + 10*x2*x2
9567 #We found 12 points on the generator of the ideal
9568 #They are: (-3,15,19,33,40,42,46,50,83,88,102,108)
9569 9570 9571 9572 9573 9574 9575 9576 9577 9578 9579 9580 9581 9582 9583 9584 9585 9586 9587 9588 9589 inverse_mod a:
9590 ▼ $(ORBITER\_PATH)\texttt{orbiter.out} -v.2 -inverse\_mod 18059241 58014043
9591 9592 9593 jacobi_{35,41}:
9594 ▼ $(ORBITER\_PATH)\texttt{orbiter.out} -v.5 -jacobi_{35,41}
9595 ▼ pdflatex jacobi_{35,41}.tex
9596 ▼ open jacobi_{35,41}.pdf
9597 9598 659
\end{verbatim}
Section 9.2: Representation Theory

representation on polynomials of degree 3:
$\text{(ORBITER_PATH)orbiter.out}\cdot-v\cdot4$\endquote 
$\text{-define}\cdot-G\cdot-linear\cdot\text{group}\cdot-PGL\cdot4\cdot3\cdot-end$\endquote 
$\text{-with}\cdot-G\cdot-do$\endquote
$\text{-group}\cdot\text{theoretic}\cdot\text{activity}$\endquote
$\text{-representation}\cdot\text{on}\cdot\text{polynomials}\cdot3$\endquote
$\text{-end}$
$\text{(ORBITER_PATH)orbiter.out}\cdot-v\cdot2$\endquote
$\text{-loop}\cdot L\cdot 0\cdot 9\cdot 1\cdot -\text{draw}\cdot\text{matrix}$\endquote
$\text{-input}\cdot\text{csv}\cdot\text{file}\cdot PGL\cdot4\cdot3\rep\cdot3\cdot%L.csv$\endquote
$\text{-box}\cdot\text{width}\cdot40\cdot-\text{bit}\cdot\text{depth}\cdot24\cdot-\text{partition}\cdot3\cdot20\cdot20\cdot-end$\endquote
$\text{-end}\cdot\text{loop}$
$\text{(ORBITER_PATH)orbiter.out}\cdot-v\cdot4$\endquote

representation tetrahedral group on polynomials of degree 3:
-define G-linear_group-GL.3.3:\
-subgroup_by_generators-tetra."12".2:\
"0,1,0,0,1,1,0,0,0,0,1,2,0,0,0,2,0":\-end:\
-with G-do:\
-group_theoretic_activity:\

Section 9.3: Cryptography:

EC_add:

EC_cyclic_subgroup:

EC_points.13:

#write GL.3.3 Subgroup tetra.12_rep.3.0.csv

SECTION_CRYPTOGRAPHY:
EC_points_199:

```bash
$(ORBITER_PATH)orbiter.out -v.2
```

- `define F = finite_field -q 199
- `with F
- `finite_field_activity
- `EC_points "EC_5_7_q199" 5-7

```
EC_Koblitz_encoding:

```
$(ORBITER_PATH)orbiter.out -v.6 -seed 17
```

- `define F = finite_field -q 199
- `with F
- `finite_field_activity
- `EC_Koblitz_encoding 5-7 67 147,164 DEABBE

```
EC_bsgs:

```
$(ORBITER_PATH)orbiter.out -v.2
```

- `define F = finite_field -q 199
- `with F
- `finite_field_activity
- `EC_bsgs 5-7 147,164 212

```
"172,158,45,195,50,22,10,103,55,33,50,22,145,105,31,74,73,155,67,60,25,6
```

```
EC_bsgs_decode:

```
$(ORBITER_PATH)orbiter.out -v.2
```

- `define F = finite_field -q 199
- `with F
- `finite_field_activity
- `EC_bsgs_decode 5-7 129,176 212

```
"127,188,51,141,85,29,106,90,41,105,179,71,171,2,16,197,183,72,27,129,37,10
```

```
"50,179,169,13,153,169,115,116,188,110,176
```

```
- end
```

```
- end
```

```
- end
```

```
- end
```

```
- end
```

662
NTRU
\[ N=7 \]
\[ P=3 \]
\[ Q=41 \]
\[ D=2 \]
\[ NTRUE\_XN1=\"-1,0,0,0,0,0,0,1,\" \]
\[ ALICE\_PRIVATE\_F=\"-1,0,1,1,-1,0,1\" \]
\[ ALICE\_PRIVATE\_G=\"0,-1,-1,0,1,0,1\" \]
\[ ALICE1:\]
\[ (ORBITER\_PATH)orbiter.out -v 2 \]
\[ define F=finite_field -q \( NTRU\_Q \) -end \]
\[ with F=do \]
\[ extended_gcd_for_polynomials \]
\[ \( NTRUE\_XN1 \) \( (ALICE\_PRIVATE\_F) \) \]
\[ end \]
\[ #F_\( q(x) \) = 8x^6 + 26x^5 + 31x^4 + 21x^3 + 40x^2 + 2x + 37 \]
\[ ALICE\_PRIVATE\_FQ=\"37,2,40,21,31,26,8\" \]
\[ ALICE2:\]
\[ (ORBITER\_PATH)orbiter.out -v 2 \]
\[ define F=finite_field -q \( NTRU\_P \) -end \]
\[ with F=do \]
\[ finite_field_activity \]
\[ extended_gcd_for_polynomials \]
\[ \( NTRUE\_XN1 \) \( (ALICE\_PRIVATE\_F) \) \]
\[ end \]
\[ #F_\( p(x) = x^6 + 2x^5 + x^3 + 2x^2 + x + 1 \]
\[ ALICE\_PRIVATE\_FP=\"1,1,1,0,2,1\" \]
\[ ALICE\_Public\_Key:\]
\[ (ORBITER\_PATH)orbiter.out -v 2 \]
\[ define F=finite_field -q \( NTRU\_Q \) -end \]
\[ with F=do \]
\[ finite_field_activity \]
\[ polynomial_mult_mod \( (ALICE\_PRIVATE\_F) \) \]
\[ (ALICE\_PRIVATE\_G) \( (NTRUE\_XN1) \) \]
\[ end \]
\[ #C_\( X = 20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30 \]
ALICE_PUBLIC_KEY="30,26,8,38,2,40,20"
BOB_MESSAGE="1,-1,1,1,0,-1"
BOB_ONE_TIME_KEY="-1,1,0,0,0,-1,1"

NTRU encrypt:
$\{(\text{ORBITER PATH})\text{orbiter.out} -v 2 \\
define F -finite_field -q $(NTRU Q) -end \\
with F -do \\
-finite_field_activity \\
-NTRU_encrypt $(NTRU N) $(NTRU P) $(ALICE_PUBLIC_KEY) \\
$(BOB_ONE_TIME_KEY) $(BOB_MESSAGE) -end$

#E(X)=31X^6+19X^5+4X^4+2X^3+40X^2+3X+25
BOB_ENCRYPT="25,3,40,2,4,19,31"

NTRU decrypt1:
$\{(\text{ORBITER PATH})\text{orbiter.out} -v 2 \\
define F -finite_field -q $(NTRU Q) -end \\
with F -do \\
-finite_field_activity \\
-polynomial_mult_mod $(ALICE_PRIVATE_F) \\
-polynomial_center_lift $(ALICE_C1) -end$

#C(X)=X^6+10X^5+33X^4+40X^3+40X^2+X+40
ALICE_C1="40,1,40,40,33,10,1"

NTRU decrypt2:
$\{(\text{ORBITER PATH})\text{orbiter.out} -v 2 \\
define F -finite_field -q $(NTRU Q) -end \\
with F -do \\
-finite_field_activity \\
polynomial_center_lift $(ALICE_C1) -end$

#A(X)=X^6+10X^5-8X^4-X^3-X^2+X-1
ALICE_C2="-1,1,-1,-1,-8,10,1"

NTRU decrypt3:
$\{(\text{ORBITER PATH})\text{orbiter.out} -v 2 \\
define F -finite_field -q $(NTRU Q) -end \\
with F -do \\
-finite_field_activity \\
polynomial_reduce_mod_p $(ALICE_C2) -end$
#A(X)=X^6+X^5+X^4+2X^3+2X^2+X+2
ALICE_C3="2,1,2,1,1,1"

NTRU_decrypt4:

$\text{(ORBITER_PATH)}$orbiter.out -v.2
  -define F=finite_field -q$(NTRU_Q).end
  -with F=do
  -finite_field_activity
  -polynomial_mult_mod$(ALICE_PRIVATE_FP).
  $(ALICE_C3)$=$(NTRUE_XN1).
  -end

#C(X)=2X^5+X^3+X^2+2X+1
ALICE_C4="1,2,1,1,0,2"

NTRU_decrypt5:

$\text{(ORBITER_PATH)}$orbiter.out -v.2
  -define F=finite_field -q$(NTRU_P).end
  -with F=do
  -finite_field_activity
  -polynomial_center_lift$(ALICE_C4).end

#A(X)=\text{-}X^5+X^3+X^2-X+1
plaintext=BOB_MESSAGE

inv_59_mod:

$\text{(ORBITER_PATH)}$orbiter.out -v.2 -inverse_mod 59:10200
the-inverse-of-59-mod:10200-is:2939

RSA_e:

$\text{(ORBITER_PATH)}$orbiter.out -v.2
  -RSA:59:10403:2:1921,1605,1804,2116,0518

RSA_d:

$\text{(ORBITER_PATH)}$orbiter.out -v.2
  -RSA:2939:10403:2:902,3509,9833,3548,5181

665
im1: $(ORBITER\_PATH)\texttt{orbiter.out} -v -2 -inverse \mod 869 \cdot 1843488
#the inverse of 869 \mod 1843488 is 386093
#\textsc{FUNFACTOR:}

\begin{verbatim}
RSA_e1:
  \$\(\texttt{ORBITER\_PATH}\texttt{orbiter.out} -v -2\)
  -RSA 386093 \cdot 1846303 \cdot 3 = "62114,60103,201518"

RSA_d1:
  \$\(\texttt{ORBITER\_PATH}\texttt{orbiter.out} -v -2\)
  -RSA 869 \cdot 1846303 \cdot 3 = "1248407,345776,317846"
\end{verbatim}

\begin{verbatim}
im1061: \$\(\texttt{ORBITER\_PATH}\texttt{orbiter.out} -v -2\)
  -inverse \mod 1061 \cdot 25320204
#the inverse of 1061 \mod 25320204 is 2076209
\end{verbatim}

\begin{verbatim}
RSA_e2:
  \$\(\texttt{ORBITER\_PATH}\texttt{orbiter.out} -v -2\)
  -RSA_{\text{encyrypt text:}} 2076209 \cdot 25330309 \cdot 3 = \text{creamcheese}
  \text{creamcheese is:} 408918,1735142,239809,654636
\end{verbatim}

\begin{verbatim}
RSA_d2:
  \$\(\texttt{ORBITER\_PATH}\texttt{orbiter.out} -v -2\)
  -RSA_{\text{encyrypt text:}} 386093 \cdot 1846303 = \text{creamcheese}
  \text{creamcheese is:} 19019931,1619805,740498,2671344
\end{verbatim}

\begin{verbatim}
\#7253 \cdot 8171 = 59264263
\end{verbatim}
im3:
  ➤ $(ORBITER\_PATH)\texttt{orbiter.out}-v.2\$
  ➤ ➤ $\text{inverse\_mod\,2909\,59248840}$
  ➤ $\text{the\,inverse\,of\,2909\,mod\,59248840\,is\,4358629}$

RSA\textsubscript{e3}:
  ➤ $(ORBITER\_PATH)\texttt{orbiter.out}-v.2\$
  ➤ ➤ $\text{-RSA\,encrypt\,text\,2909\,59264263\,3\,encrypted}$

RSA\textsubscript{d3}:
  ➤ $(ORBITER\_PATH)\texttt{orbiter.out}-v.2\$
  ➤ ➤ $\text{-RSA\,decrypt\,text\,583\,62251979\,3\,venividivici}$

im4:
  ➤ $(ORBITER\_PATH)\texttt{orbiter.out}-v.2\$
  ➤ ➤ $\text{-RSA\,decrypt\,text\,583\,62251979\,venividivici}$

RSA\textsubscript{e4}:
  ➤ $(ORBITER\_PATH)\texttt{orbiter.out}-v.2\$
  ➤ ➤ $\text{-RSA\,decrypt\,text\,583\,62251979\,venividivici}$

#51403,182516,200504=\texttt{encrypted}

#e=.

#e=.

im4:
  ➤ $(ORBITER\_PATH)\texttt{orbiter.out}-v.2\$
  ➤ ➤ $\text{-RSA\,decrypt\,text\,583\,62251979\,3\,venividivici}$
im5:  $(ORBITER\_PATH)orbiter.out\:-v\:-inverse\_mod\:173\:52504368
#the inverse of 173 mod 52504368 is 38543669

RSA_e5:
$(ORBITER\_PATH)orbiter.out\:-v\:-2\:\\ - RSA\_encrypt\_text\:38543669\:52518863\:3\:fascinating

#-RSA\_encrypt\_text\:38543669\:52518863\:fascinating
#31526751,8962078,51045732,51894467

RSA_d5:
$(ORBITER\_PATH)orbiter.out\:-v\:-2\:\\ - RSA\:173\:52518863\:"31526751,8962078,51045732,51894467"

RSA_d6:
$(ORBITER\_PATH)orbiter.out\:-v\:-2\:\\ - RSA\:47177497\:55040413\:"28702119,48926559"

smooth:
$(ORBITER\_PATH)orbiter.out\:-v\:-2\:\\ -sift\_smooth\:100000\:100\:"2,3,5,7,11,13,17,19"

# #
# 7853\:*7673\:=60256069
# The inverse of \(9017 \mod 60240544\) is \(14430473\)

RSA_e8:

```
$(ORBITER_PATH)orbiter.out -v 2
```

- RSA_encrypt_text-9017-60256069-3-strawberry

```
quadratic_sieve:
```

```
$\text{ thoải } 1002001$
```

####

pseudoprime3:

```
$(ORBITER_PATH)orbiter.out -v 5
```

- pdf\latex\pseudoprime_3.tex
- open-pseudoprime_3.pdf

```
pseudoprime10:
```

```
$\text{ thoải } 4460190157$
```

```
PR10_test1:
```

```
$\text{ thoải } 974586571$
```
10068  > $(ORBITER_PATH)orbiter.out-v.5-power_mod-974586571-15222492-4460190157
10069  > $(ORBITER_PATH)orbiter.out-v.5-power_mod-974586571-284796-4460190157
10070
10071
10072 pseudoprime11:
10074  > $(ORBITER_PATH)orbiter.out-v.5-
10076  > pdflatex pseudoprime_11.tex
10077  > open-pseudoprime_11.pdf
10078
10079  #63814633367
10080
10081  #product:is:284625399616057168619
10082
10083 pseudoprime20:
10085  > $(ORBITER_PATH)orbiter.out-v.5-
10087  > pdflatex pseudoprime_20.tex
10088  > open-pseudoprime_20.pdf
10089
10090
10091 PR10:
10092  > $(ORBITER_PATH)orbiter.out-v.5-
10093  > -primitive_root:4460190157
10094
10095  #mistake!:long:integer:overflow
10096  #a:primitive:root:modulo:165222861:is:1293
10097
10098
10099
10100
10101 pseudoprime50:
10102  > $(ORBITER_PATH)orbiter.out-v.5-
10103  > -seed:2531011-find_pseudoprime:50-5:0:0
10104  > pdflatex pseudoprime_50.tex
10105  > open-pseudoprime_50.pdf
10106
10107
10108  #91322792878581218181431392170986926262336688354473
10109
10110 pseudoprime51:
10111  > $(ORBITER_PATH)orbiter.out-v.5-
10113  > pdflatex pseudoprime_51.tex
10114  > open-pseudoprime_51.pdf

670
10115 10116 #754600727746834470214089702490004944659715367045417
10117 10118 # product 689122459660508196061999942326431573235295324400658436661744403244049
572914094379904326661586100241
10119 10120 pseudoprime30:
10121 10122 $(ORBITER\_PATH)orbiter.out -v 5 \$
10123 10124 -seed 2531011 -find pseudoprime 30 5 5 5
10125 10126 pdflatex pseudoprime_30.tex
10127 10128 open pseudoprime_30.pdf
10129 # 286525565474504516914595596387
10130 10131 pseudoprime31:
10132 $(ORBITER\_PATH)orbiter.out -v 5 \$
10133 -seed 2531011 -find pseudoprime 31 5 5 5
10134 10135 pdflatex pseudoprime_31.tex
10136 10137 open pseudoprime_31.pdf
10138 10139 # 2514911323283298698837184692002835573476743643265896783515097
10140 10141 maybe 2 seconds
10142 10143 pseudoprime33:
10144 $(ORBITER\_PATH)orbiter.out -v 5 \$
10145 -seed 2531011 -find pseudoprime 33 5 5 5
10146 10147 pdflatex pseudoprime_33.tex
10148 10149 open pseudoprime_33.pdf
10150 10151 # 371674199498295345543363004459891
10152 10153 10154 pseudoprime34:
10155 $(ORBITER\_PATH)orbiter.out -v 5 \$
10156 -seed 2531011 -find pseudoprime 34 5 5 5
10157 10158 pdflatex pseudoprime_34.tex
10159 10160 open pseudoprime_34.pdf
10161 10162 # 9309708224110488378214945245346817
10163 10164 # 3460178351758962531912872979731874528849142238619677890786061016947
10165 10166 # 18 sec
10167 10168

671
pseudoprime35:
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.5 \}}\)
\(\text{\texttt{-seed 2531011-find pseudoprime 35 5 5 5 \}}\)
\(\text{\texttt{pdflatex pseudoprime35.tex \}}\)
\(\text{\texttt{open-pseudoprime35.pdf \}}\)
#81329557792505271120435930267680203

pseudoprime36:
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.5 \}}\)
\(\text{\texttt{-seed 2531011-find pseudoprime 36 5 5 5 \}}\)
\(\text{\texttt{pdflatex pseudoprime36.tex \}}\)
\(\text{\texttt{open-pseudoprime36.pdf \}}\)
#162624680891993404333363207561599139

# factoring takes 46 seconds

MATH360_hw2:
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.3 \}}\)
\(\text{\texttt{-define F-finite_field-q 16-end \}}\)
\(\text{\texttt{-with F-do-finite_field_activity \}}\)
\(\text{\texttt{-parse_and_evaluate "test"."a+b":"a=8,b=14"-end \}}\)
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.3 \}}\)
\(\text{\texttt{-define F-finite_field-q 16-end \}}\)
\(\text{\texttt{-with F-do-finite_field_activity \}}\)
\(\text{\texttt{-parse_and_evaluate "test"."a*b":"a=9,b=13"-end \}}\)
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.3 \}}\)
\(\text{\texttt{-define F-finite_field-q 16-end \}}\)
\(\text{\texttt{-with F-do-finite_field_activity \}}\)
\(\text{\texttt{-parse_and_evaluate "test"."a*a":a=9-a\"-end \}}\)
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.3 \}}\)
\(\text{\texttt{-define F-finite_field-q 16-end \}}\)
\(\text{\texttt{-with F-do-finite_field_activity \}}\)
\(\text{\texttt{-parse_and_evaluate "test"."(a+b)*(a+b)":"a=5,b=7"-end \}}\)
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.3 \}}\)
\(\text{\texttt{-define F-finite_field-q 16-end \}}\)
\(\text{\texttt{-with F-do-finite_field_activity \}}\)
\(\text{\texttt{-parse_and_evaluate "test"."a*a+b*b":"a=5,b=7"-end \}}\)
\(\text{\texttt{F_256_Rijndahl: \}}\)
\(\text{\texttt{\$ (ORBITER\_PATH) orbiter.out -v.3 \}}\)

672
10208  ▷ -define F=finite_field=q 256 -override_polynomial=283 -end \n10209  ▷ -with F=do=finite_field_activity=cheat_sheet_GF -end
10210
10211
10212
10213
10214
10215
10216
10217 all_square_roots_mod_n 1549411:
10218  $(ORBITER_PATH) orbiter.out -v 3 -all_square_roots_mod_n 1075922 1549411
10219
10220
10221
10222 power_mod_211:
10223  $(ORBITER_PATH) orbiter.out -v 3 -power_mod_n 2 211
10224  $(ORBITER_PATH) orbiter.out -v 3 \n10225  ▷ -plot_function:power_mod_n a2_n211.csv
10226  $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix \n10227  ▷ -input_csv_file:power_mod_n a2_n211_graph.csv \n10228  ▷ -box_width:10 -bit_depth:8 -partition:3 211 211 -end
10229
10230 power_mod_231:
10231  $(ORBITER_PATH) orbiter.out -v 3 -power_mod_n 2 31
10232  $(ORBITER_PATH) orbiter.out -v 3 \n10233  ▷ -plot_function:power_mod_n a2_n31.csv
10234  $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix \n10235  ▷ -input_csv_file:power_mod_n a2_n31_graph.csv \n10236  ▷ -box_width:10 -bit_depth:8 -partition:3 31 31 -end
10237
10238 power_mod_331:
10239  $(ORBITER_PATH) orbiter.out -v 3 -power_mod_n 3 31
10240  $(ORBITER_PATH) orbiter.out -v 3 \n10241  ▷ -plot_function:power_mod_n a3_n31.csv
10242  $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix \n10243  ▷ -input_csv_file:power_mod_n a3_n31_graph.csv \n10244  ▷ -box_width:10 -bit_depth:8 -partition:3 31 31 -end
10245
10246
10247
10248  ###############################################################################
10249  # Chapter 10 -- Coding Theory
10250  ###############################################################################
10251
10252
10253
10254  ###############################################################################
10255 #Section 10.1: Coding Theory
10256
10257 SECTION CODING THEORY INTRODUCTION:
10258
10259
10260 10261 Hamming space 4 2 distance matrix:
10262 10263 $(ORBITER PATH)orbiter.out-Hamming_space_distance_matrix_4_2$
10264 10265 10266 ⊿ ⊿ $(ORBITER PATH)orbiter.out-v_2-draw_matrix$
10267 10268 10269 ⊿ ⊿ ⊿ ⊿ ⊿$(ORBITER PATH)orbiter.out-v_2-draw_matrix$
10270
10271 ⊿ ⊿ ⊿ $(ORBITER PATH)orbiter.out-v_2-draw_matrix$
10272 10273 10274 ⊿ ⊿ $(ORBITER PATH)orbiter.out-v_2-draw_matrix$
10275 10276 10277 10278
10279 Hamming 5 2 graph:
10280 10281 $(ORBITER PATH)orbiter.out-v_2$
10282 10283 10284 10285 10286 10287 10288 10289 10290 10291 10292 10293 10294
10295 10296 10297 10298 10299 10300 10301
10302
10302 -graph_theoretic_activity export_graphviz -end
10303 -with G do
10304 -graph_theoretic_activity save -end
10305 -with G do
10306 -graph_theoretic_activity automorphism_group -end
10307 pdflatex Hamming_5_2_code_5_2_3_report.tex
10308 open Hamming_5_2_code_5_2_3_report.pdf

10310 # group has order 32
10311
10312
10313
10314
10315
10316
10317 code_6:
10318 $(ORBITER_PATH)orbiter.out -v 2
10319 -general_code_binary 6 0,60,50,41,14,21,27,39
10320 $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix
10321 -input_csv_file code_matrix_8_8.csv
10322 -box_width 20 -bit_depth 24
10323 -partition 2 1,1,1,1,1,1,1,1 1,1,1,1,1,1,1,1 -end
10324 pdflatex code_6_8.tex
10325 open code_6_8.pdf
10326 open code_matrix_8_8_draw.bmp
10327
10328
10329 # linear code with generator matrix
10330 # 111100
10331 # 110010
10332 # 101001
10333
10334
10335
10336
10337 # Section 10.2: Hamming codes
10338
10339 SECTION_CODING THEORY HAMMING CODES:
10340
10341
10342
10343 Hamming_generator:
10344 $(ORBITER_PATH)orbiter.out -v 2
10345 -define F finite_field q 2 -end
10346 -define v vector -field F format 3
10347 -dense $(SIMPLEX_CODE_GENERATOR)
10348 -end

675
Hamming code words:

$\texttt{(ORBITER\_PATH)orbiter.out-v.2}\backslash$

$\texttt{-define-v-vector:-dense$(HAMMING\_CODE\_ROWS\_IN\_BINARY\_RANKS)$-end}$

$\texttt{-linear\_code\_through\_basis-7\_v}$

pdflatex\_code\_n7\_k4\_q2.tex

open\_code\_n7\_k4\_q2.pdf

Hamming weight enumerator:

$\texttt{(ORBITER\_PATH)orbiter.out-v.2}$

$\texttt{-define-v-vector:-dense$(HAMMING\_CODE\_GENERATOR)$}$

$\texttt{-weight\_enumerator-v-end}$

Hamming code diagram:

$\texttt{(ORBITER\_PATH)orbiter.out-v.2-code\_diagram-Hamming\_7\_4}\backslash$

$\texttt{-input\_csv\_file-Hamming\_7\_4\_diagram\_01\_7\_16.csv}\backslash$

$\texttt{-box\_width-25-bit\_depth-24-partition-4\_16-8-end}$

$\texttt{(ORBITER\_PATH)orbiter.out-v.2-draw\_matrix}$

$\texttt{-input\_csv\_file-Hamming\_7\_4\_diagram\_7\_16.csv}\backslash$

$\texttt{-box\_width-25-bit\_depth-14-partition-4\_16-8-end}$

$\texttt{(ORBITER\_PATH)orbiter.out-v.2-code\_diagram-Hamming\_7\_4\_word\_0\_0\_7-metric\_balls\_1}$

$\texttt{(ORBITER\_PATH)orbiter.out-v.2-code\_diagram-Hamming\_7\_4\_word\_1\_67\_7-metric\_balls\_1}$
$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_2^\text{\_37^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_3^\text{\_102^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_4^\text{\_22^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_5^\text{\_85^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_6^\text{\_51^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_7^\text{\_112^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_8^\text{\_15^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_9^\text{\_76^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_10^\text{\_42^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_11^\text{\_105^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_12^\text{\_25^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_13^\text{\_90^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_14^\text{\_60^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-code\_diagram}^\text{-Hamming_7_4\_word_15^\text{\_127^\text{-metric\_balls}}}^\text{1}$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2-loop\_L.0.16.1\_draw\_matrix}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_input\_csv\_file\_Hamming_7_4\_word_2\_L\_diagram_7.1.csv}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_box\_width\_25\_bit\_depth\_8\_partition\_4.16\_8\_end}\$

$\text{code\_Hamming\_systematic}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_define\_v\_vector\_dense\_HAMMING\_CODE\_ROWS\_IN\_BINARY\_RANKS}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_linear\_code\_through\_basis\_v}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_draw\_matrix}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_input\_csv\_file\_code\_matrix_16.8.csv}\$

$\text{(ORBITER\_PATH)or}^\text{b}\text{.out-v.2\_box\_width\_25\_bit\_depth\_8\_partition\_2.16\_8\_end}\$

$\text{open\_code\_matrix_16.8\_draw.bmp}\$

$\text{pdflatex\_code_7.16.tex}\$

$\text{open\_code_7.16.pdf}\$

677
Hamming RREF:

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
define F -finite_field -q 2 -end
```

```bash
define v -vector -format 4 -field F
```

```bash
dense $(HAMMING_CODE_GENERATOR)
```

```bash
end
```

```bash
with F -do
```

```bash
finite_field_activity
```

```bash
RREF v -end
```

```bash
pdflatex RREF_example_q2_4_7.tex
```

```bash
gs -sDEVICE=png16 -dFIXEDMEDIA -dDeviceWidthPoints=500 -dDeviceHeightPoints=450
```

```bash
l240 oHamming_dual_page%02d.png
```

```bash
RREF_example_q2_4_7.pdf
```

Hamming nullspace:

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
define F2 -finite_field -q 2 -end
```

```bash
define v -vector -format 4 -field F2
```

```bash
dense $(HAMMING_CODE_GENERATOR)
```

```bash
end
```

```bash
with F2 -do
```

```bash
finite_field_activity
```

```bash
nullspace v
```

```bash
normalize_from_the_right
```

```bash
end
```

```bash
pdflatex nullspace_4_7.tex
```

```bash
open nullspace_4_7.pdf
```

# check equations of the Hamming code:

```bash
# a4+a5+a6+a7=1+0+1+0=0 mod 2 OK.
```

```bash
# a2+a3+a6+a7=0+1+1+0=0 mod 2 OK.
```

```bash
# a1+a3+a5+a7=1+1+0+0=0 mod 2 OK.
```
Hamming\_long:

```
10483 \$\text{(ORBITER\_PATH)orbiter.out-v.2-long\_code.7.4}\$
10484 \$\text{0,5,6}$.\$
10485 \$\text{1,4,6}$.\$
10486 \$\text{2,4,5}$.\$
10487 \$\text{3,4,5,6}$.\$
10488 \$\text{(ORBITER\_PATH)orbiter.out-v.2-loop.16.1-draw\_matrix}\$
10489 \$\text{-input\_csv\_file-long\_code\_genma.\_n7.k4\_codeword_\%L.csv}\$
10490 \$\text{-box\_width.25-bit\_depth.8-partition.3.4.2-end}\$
10491 \$\text{-end\_loop}\$
10492
10493
10494 \#long\_code\_genma.\_n7.k4\_codeword_0.csv
10495 \#long\_code\_genma.\_n7.k4\_codeword_15.csv
10496 \#Weight\_distribution:(0,3^7,4^7,7)
10497
10498 Hamming\_code\_macwilliams:
10499 \$\text{(ORBITER\_PATH)orbiter.out-v.2}\$
10500 \$\text{-make\_macwilliams\_system.7.4.2}\$
10501 \$\text{pdflatex MacWilliams_n7.k4.q2.tex}\$
10502 \$\text{open MacWilliams_n7.k4.q2.pdf}\$
10503
10504 Hamming\_singer:
10505 \$\text{(ORBITER\_PATH)orbiter.out-v.3}\$
10506 \$\text{-define\_G-linear\_group-PGL.3.2-singer.1-end}\$
10507 \$\text{-with\_G-do}\$
10508 \$\text{-group\_theoretic\_activity}\$
10509 \$\text{-report}\$
10510 \$\text{-orbits\_on\_points}\$
10511 \$\text{-end}\$
10512 \$\text{pdflatex PGL.3.2.Singer.3.2.1\_report.tex}\$
10513 \$\text{open PGL.3.2.Singer.3.2.1\_report.pdf}\$
10514
10515 \#cycle.is.0,1,2,5,3,4,6
10516
10517 \#1001110
10518 \#0100111
10519 \#0011101
10520
```
Hamming cyclic generator:

```bash
$ (ORBITER_PATH)orbiter.out -v 2 \n  -define F -finite_field -q 2 -end \n  -define v -vector -format 3 -field F \n  -dense $(SIMPLEX_CODE_GENMA_CYCLIC) \n  -end \n  -with F -do -finite_field_activity \n  -nullspace v \n  -end \n
pdflatex nullspace_3_7.tex
```

```
open nullspace_3_7.pdf
```

Hamming cyclic long:

```bash
$ (ORBITER_PATH)orbiter.out -v 2 -long_code 7 4 \n  "0,4,6". \n  "1,4,5,6". \n  "2,4,5". \n  "3,5,6". \n
$(ORBITER_PATH)orbiter.out -v 2 -loop L 0:16:1 -draw_matrix \n  -input_csv_file long_code_genma_n7_k4_codeword_%L.csv \n  -box_width 25 -bit_depth 8 -partition 3:4:2 -end \n
-end_loop
```

```
open code_matrix_16_8_draw.bmp
```

```
pdflatex code_7_16.tex
```

```
open code_7_16.pdf
```

Hamming cyclic clean ns:

```bash
$ (ORBITER_PATH)orbiter.out -v 2 \n  -define F -finite_field -q 2 -end \n  -define v -vector -format 3 -field F \n  -dense $(SIMPLEX_CODE_GENMA_CYCLIC) \n  -end \n  -with F -do -finite_field_activity \n  -nullspace v \n```

```
open nullspace_3_7.pdf
```

```
open code_7_16.pdf
```
Hamming cyclic clean:

$(\text{ORBITER PATH})$orbiter.out -v.2 -normalize_from_the_right

Hamming cyclic clean long:

$(\text{ORBITER PATH})$orbiter.out -v.2 -long_code.7.4

Section 10.3: Coding Theory -- Golay codes
Golay23 code words:

```
define v-vector-dense(GOLAY_23_COLUMN_RANKS_PROJECTIVELY)-end
-linear_code_through_columns_of_parity_check_projectively:12-v
```

```
pdflatex code_n23_k12_q2.tex
open-code_n23_k12_q2.pdf
```

Golay23 code diagram:
```
-normalize
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
10662 ▷ ▷ -finite_field_activity\n10663 ▷ ▷ ▷ -polynomial_division_from_file\n10664 ▷ ▷ ▷ text.csv.13-end
10665 ▷ ▷ pdflatex polynomial_division_file_13.tex
10666 ▷ ▷ open-polynomial_division_file_13.pdf
10667
10668 #\usepackage[a1paper]{geometry}
10669
10670 encode_text_5bits_check:
10671 ▷ $(ORBITER_PATH)orbiter.out.--v.2\n10672 ▷ ▷ -define-F--finite_field--q.2--end\n10673 ▷ ▷ -with-F--do\n10674 ▷ ▷ -finite_field_activity\n10675 ▷ ▷ -polynomial_division_from_file.text_with_1error.csv.13-end
10676 ▷ ▷ pdflatex polynomial_division_file_13.tex
10677 ▷ ▷ open-polynomial_division_file_13.pdf
10678
10679 encode_constitution_field_induction:
10680 ▷ $(ORBITER_PATH)orbiter.out.--encode_text_5bits."We the People of the United States, in Order to form a more perfect Union, establish Justice, insure domestic Tran\n
10681 ▷ $(ORBITER_PATH)orbiter.out.--reformat."constitution.csv"."constitution_80.csv".8\n10682 ▷ $(ORBITER_PATH)orbiter.out.--field_induction."constitution.csv"."constitution_32\n10683 ▷ ▷ csv".16_\n10684 ▷ $(ORBITER_PATH)orbiter.out.--v.2.--draw_matrix\n10685 ▷ ▷ -input_csv_file.constitution_80.csv\n10686 ▷ ▷ -box_width.20.-bit_depth.24.-partition.5."21"."80".--end
10687 ▷ $(ORBITER_PATH)orbiter.out.--v.2.--draw_matrix\n10688 ▷ ▷ -input_csv_file.constitution_32.16.csv\n10689 ▷ ▷ -box_width.20.-bit_depth.24.-partition.5."21"."16".--end
10690
10691
10692
10693 encode_text_5bits_1error:
10694 ▷ $(ORBITER_PATH)orbiter.out.--encode_text_5bits\n10695 ▷ ▷ "Hithere"."text.csv"
10696 ▷ $(ORBITER_PATH)orbiter.out.--v.2\n10697 ▷ ▷ -define-F--finite_field--q.2--end\n10698 ▷ ▷ -with-F--do\n10699 ▷ ▷ -finite_field_activity\n10700 ▷ ▷ -polynomial_division_from_file_all_k_bit_error_patterns\n10701 ▷ ▷ ▷ text.csv.13.1--end

683
encode_text_zero_5bits_1error:
$\$(ORBITER\ PATH)\ orbiter.out-\ v\ 2\\$
$\define F=-finite\ field\ -q\ 2\ -end\$
$\with F\ -do\$
$\finite\ field\ activity\$
$\polynomial\ division\ from\ file\ all\ k\ bit\ error\ patterns\ text\ zero.csv\ 13\ 1\ -e$
end

encode_text_zero_5bits_2error:
$\$(ORBITER\ PATH)\ orbiter.out-\ v\ 2\\$
$\define F=-finite\ field\ -q\ 2\ -end\$
$\with F\ -do\$
$\finite\ field\ activity\$
$\polynomial\ division\ from\ file\ all\ k\ bit\ error\ patterns\ text\ zero.csv\ 13\ 2\ -e$
end

encode_text_5bits_2error:
$\$(ORBITER\ PATH)\ orbiter.out-\ encode\ text\ 5bits:\ "Hithere"\ "text.csv"$
$\$(ORBITER\ PATH)\ orbiter.out-\ v\ 2\\$
$\define F=-finite\ field\ -q\ 2\ -end\$
$\with F\ -do\$
$\finite\ field\ activity\$
$\polynomial\ division\ from\ file\ all\ k\ bit\ error\ patterns\ text\ 0\ 7.csv\ 13\ 2\ -e$
end

encode_text_5bits_error_0_7:
$\$(ORBITER\ PATH)\ orbiter.out-\ v\ 2\\$
$\define F=-finite\ field\ -q\ 2\ -end\$
$\with F\ -do\$
$\finite\ field\ activity\ -polynomial\ division\ from\ file\ text\ error_0_7.csv\ 13\ -e$
end

pdflatex-polynomial\ division_file_13.tex
open-polynomial\ division_file_13.pdf
10746 CRC_2_4_3:
10747 >> $(ORBITER_PATH)orbiter.out-v.3.-define-F.-finite_field.-q.2.-end\
10748   >> -with-F.-do.-finite_field_activity.-find_CRC_polynomials:2:4:3-end
10749 10750 #algebra_global::find_CRC_polynomials.info=4.check=3.nb_sol=3
10751 10752 #0:::1101
10753 10754 10755 CRC_2_4_4:
10756 10757 > $(ORBITER_PATH)orbiter.out-define-F.-finite_field.-q.2.-end\n10758 >   -with-F.-do.-finite_field_activity\n10759 >   -find_CRC_polynomials:2:4:4\n10760 >   -end
10761 10762 CRC_2_5_4:
10763 10764 > $(ORBITER_PATH)orbiter.out-define-F.-finite_field.-q.2.-end\n10765 >   -with-F.-do.-finite_field_activity\n10766 >   -find_CRC_polynomials:2:5:4\n10767 10768 CRC_2_6_4:
10769 10770 > $(ORBITER_PATH)orbiter.out-define-F.-finite_field.-q.2.-end\n10771 >   -with-F.-do.-finite_field_activity\n10772 >   -find_CRC_polynomials:2:6:4\n10773 10774 CRC_2_7_4:
10775 10776 > $(ORBITER_PATH)orbiter.out-define-F.-finite_field.-q.2.-end\n10777 >   -with-F.-do.-finite_field_activity\n10778 >   -find_CRC_polynomials:2:7:4\n10779 10780 CRC_2_8_4:
10781 10782 > $(ORBITER_PATH)orbiter.out-v.2.-define-F.-finite_field.-q.2.-end\n10783 >   -with-F.-do.-finite_field_activity\n10784 >   -find_CRC_polynomials:2:8:4\n10785 10786 CRC_2_9_4:
10787 10788 > $(ORBITER_PATH)orbiter.out-define-F.-finite_field.-q.2.-end\n10789 >   -with-F.-do.-finite_field_activity\n10790 >   -find_CRC_polynomials:2:9:4\n10791 10792 CRC_2_9_5:
10793 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10794 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10795 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{2}\texttt{9}\texttt{5}\:\\\n10796 \> \> \> \> -\texttt{end}\\\n10797 10798 CRC\texttt{2}_\texttt{10}\texttt{.5}: 10799 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10800 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10801 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{2}\texttt{10}\texttt{.5}\:\\\n10802 \> \> \> \> -\texttt{end}\\\n10803 10804 CRC\texttt{2}_\texttt{32}\texttt{.6}: 10805 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10806 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10807 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{2}\texttt{32}\texttt{.6}\:\\\n10808 \> \> \> \> -\texttt{end}\\\n10809 10810 CRC\texttt{2}_\texttt{64}\texttt{.8}: 10811 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10812 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10813 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{2}\texttt{64}\texttt{.8}\:\\\n10814 \> \> \> \> -\texttt{end}\\\n10815 10816 CRC\texttt{2}_\texttt{128}\texttt{.8}: 10817 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{v}1.-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10818 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10819 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{2}\texttt{128}\texttt{.8}\:\\\n10820 \> \> \> \> -\texttt{end}\\\n10821 10822 CRC\texttt{2}_\texttt{128}\texttt{.16}: 10823 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{v}1.-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10824 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10825 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{2}\texttt{128}\texttt{.16}\:\\\n10826 \> \> \> \> -\texttt{end}\\\n10827 10828 CRC\texttt{3}_\texttt{128}\texttt{.8}: 10829 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{v}1.-\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10830 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10831 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{3}\texttt{128}\texttt{.8}\:\\\n10832 \> \> \> \> -\texttt{end}\\\n10833 10834 CRC\texttt{3}_\texttt{128}\texttt{.10}: 10835 \> $(\textsc{ORBITER} \textsc{PATH})\texttt{orbiter.out}-\texttt{v}1.\:\\\n10836 \> \> -\texttt{define}\texttt{F}-\texttt{finite}\texttt{field}-q\texttt{2}-\texttt{end}\:\\\n10837 \> \> -\texttt{with}\texttt{F}-\texttt{do}\texttt{-finite}\texttt{field}\texttt{activity}\:\\\n10838 \> \> \> -\texttt{find}\texttt{CRC}\texttt{polynomials}\texttt{3}\texttt{128}\texttt{.10}\:\\\n10839 \> \> \> \> -\texttt{end}
10840
10841
10842
10843 CRC_2_256_8:
10844  $(\text{ORBITER\_PATH})\text{orbiter.out}\text{-define}\_F\text{-finite\_field}\_q\_2\text{-end}\$
10845  $\text{-with}\_F\text{-do}\_finite\_field\_activity\$
10846  $\text{-find}\_CRC\_polynomials\_2\_256\_8\$
10847  $\text{-end}$
10848
10849 #no-solution
10850
10851 CRC_2_256_10:
10852  $(\text{ORBITER\_PATH})\text{orbiter.out}\text{-define}\_F\text{-finite\_field}\_q\_2\text{-end}\$
10853  $\text{-with}\_F\text{-do}\_finite\_field\_activity\$
10854  $\text{-find}\_CRC\_polynomials\_2\_256\_10\$
10855  $\text{-end}$
10856
10857 #algebra\_global::find\_CRC\_polynomials\_info=256\_check=10\_nb\_sol=368
10858 #0:13\_on\_Mac
10859
10860
10861
10862 CRC_2_256_12:
10863  $(\text{ORBITER\_PATH})\text{orbiter.out}\text{-v}\_1\text{-define}\_F\text{-finite\_field}\_q\_2\text{-end}\$
10864  $\text{-with}\_F\text{-do}\_finite\_field\_activity\$
10865  $\text{-find}\_CRC\_polynomials\_2\_256\_12\$
10866  $\text{-end}$
10867
10868 #algebra\_global::find\_CRC\_polynomials\_info=256\_check=12\_nb\_sol=2654
10869 #User\_time::2:0
10870
10871
10872 CRC_3_256_10:
10873  $(\text{ORBITER\_PATH})\text{orbiter.out}\text{-define}\_F\text{-finite\_field}\_q\_2\text{-end}\$
10874  $\text{-with}\_F\text{-do}\_finite\_field\_activity\$
10875  $\text{-find}\_CRC\_polynomials\_3\_256\_10\$
10876  $\text{-end}$
10877
10878
10879 #algebra\_global::find\_CRC\_polynomials\_info=256\_check=10\_nb\_sol=140
10880 #Orbiter\_session\_finished.
10881 #User\_time::8:7
10882
10883
10884 CRC_3_256_16:
10885  $(\text{ORBITER\_PATH})\text{orbiter.out}\text{-v}\_1\text{-define}\_F\text{-finite\_field}\_q\_2\text{-end}\$
10886  $\text{-with}\_F\text{-do}\_finite\_field\_activity\$

687
find CRC polynomials 3-256-16\end
find CRC polynomials 3-8-4\end
find CRC polynomials 3-64-8\end
find CRC polynomials 3-128-10\end
# 12/26/2020: 243 polynomials in 0:56 minutes on Mac
# Section 10.5: Coding Theory: Reed-Muller codes
SECTION CODING THEORY REED_MULLER_CODES:
RM_3.1_code_words:\n\n刘海燕
\n\n\n$\$(ORBITER_PATH)\$\$orbiter.out-v.2:\n\n$-define-v-vector-dense$\$(REED_MULLER_3.1 BASIS_IN_BINARY)$-end$\n\n\n$-linear_code_through_basis-8v\n\n$pdflatex-code_n8_k4_q2.tex\n\n$open-code_n8_k4_q2.pdf\n\n$\#Codewords: (0,255,170,85,204,51,102,153,240,15,90,165,60,195,150,105)\n\nRM_3.1_Hamming_space_diagram:\n\n$\$(ORBITER_PATH)\$\$orbiter.out-v.2-code_diagram"RM_3.1"-end$\n\n$-metric_balls-1\n\n$\$(ORBITER_PATH)\$\$orbiter.out-v.2-draw_matrix$\n\n$-input_csv_file\$\$RM_3.1_diagram_01_8.16.csv\$\n\n$-box_width-25-bit_depth-8-partition-4.16.16-end\n\n$pdflatex-code_n8_k4_q2.tex\n\n$open-code_n8_k4_q2.pdf$\n\nRM_3.1_split:\n\n$\$(ORBITER_PATH)\$\$orbiter.out-split_by_values\$\$RM_3.1_holes_8.16.csv\$\n\nRM_3.1_holes_draw:\n\n$\$(ORBITER_PATH)\$\$orbiter.out-v.2$\n\n$-loop-L-0.3.1$\n\n$-draw_matrix$\n\n$-input_csv_file\$\$RM_3.1_holes_8.16.value$L.csv\$\n\n$-box_width-25-bit_depth-8-partition-5.16.16$\n\n$-end$\n\nRM_3.1_hole0:\n\n$\$(ORBITER_PATH)\$\$orbiter.out-v.3$\n\n$-define-F-finite_field-q.2-end$\n\n$-with-F-do-finite_field_activity$\n\n$-algebraic_normal_form\$\$RM_3.1_holes_8.16.value0.csv\$\$8-end$\n\n#E_0+E_1+E_2+E_3+E_4
#E.1=\text{X}0X^8\cdot7+:\text{X}1X^8\cdot7+:\text{X}2X^8\cdot7+:\text{X}3X^8\cdot7+:\text{X}4X^8\cdot7+:\text{X}5X^8\cdot7+:\text{X}6X^8\cdot7+\cdot\text{X}7X^8\cdot7

10987

10988

10989

10990

10991

10992

10993

10994

10995

10996

10997

10998
10999
11000 RM_4_1:
11001 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.2}\$
11002 \text{>> -linear\_code\_through\_columns\_of\_parity\_check-5}\$
11003 \text{>> $(REED\_MULLER\_4\_1\_COLUMNS\_OF\_PARTITY\_CHECK)}$
11004 \text{pdflatex\_code\_n16\_k5\_q2.tex}$
11005 \text{open\_code\_n16\_k5\_q2.pdf}$
11006
11007
11008 \#-codewords\_n16\_k5\_q2.csv
11009
11010
11011
11012 RM_4_1\.diagram:
11013 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.2}\$
11014 \text{>> -code\_diagram\_from\_file\_"RM_4_1\"\_}$
11015 \text{>> codewords\_n16\_k5\_q2.csv-16-}$
11016 \text{>> #-enhance-4-}$
11017 \text{>> #-metric\_balls-3}$
11018
11019 RM_4_1\.diagram\.draw:
11020 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.2\_draw\_matrix}\$
11021 \text{>> -input\_csv\_file\_RM_4_1\_diagram_01\_16\_32.csv}$
11022 \text{>> -box\_width-25\_bit\_depth-8\_partition-10\_256\_256\_end}$
11023 \text{open\_RM_4_1\_diagram_01\_16\_32\_draw.bmp}$
11024
11025
11026 RM_4_1\.split:
11027 \$\text{(ORBITER\_PATH)}\text{orbiter.out\_split\_by\_values\_RM_4_1\_holes\_16\_32.csv}$
11028
11029 RM_4_1\.diagram\.draw\_holes:
11030 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.2\_draw\_matrix}\$
11031 \text{>> -input\_csv\_file\_RM_4_1\_holes\_16\_32.csv}$
11032 \text{>> -box\_width-25\_bit\_depth-8\_partition-10\_256\_256\_end}$
11033 \text{>> $(ORBITER\_PATH)\text{orbiter.out-v.2\_loop\_L-0-7-1\_draw\_matrix}\$}
11034 \text{>> -input\_csv\_file\_RM_4_1\_holes\_16\_32\_value\_L.csv}$
11035 \text{>> -box\_width-25\_bit\_depth-8\_partition-10\_256\_256\_end\_end\_loop}$
11036
11037
11038
11039 RM_4_1\.diagram\.metric\.balls:
11040 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.2\_code\_diagram\_from\_file\_"RM_4_1\"\_}$
11041 \text{>> codewords\_n16\_k5\_q2.csv-16\_metric\_balls-3}$
11042 \text{>> $(ORBITER\_PATH)\text{orbiter.out-v.2\_draw\_matrix}\$}
11043 \text{>> -input\_csv\_file\_RM_4_1\_diagram_16\_32.csv}$
11044 \text{>> -box\_width-25\_bit\_depth-8\_partition-10\_256\_256\_end}$
11045
691
11046
11047
11048 RM_4_1_hole0:
11049 \$\{(ORBITER_PATH)orbiter.out\}-v.3.-define F\text{-}finite\_field\-q\,2.-end\$
11050 \$\{\text{-}with F\text{-}do\text{-}finite\_field\_activity\}\$
11051 \$\{\text{-}algebraic\_normal\_form}\text{-RM}_4\_1\_holes\_16\_32\_value0.csv\text{-end}\$
11052
11053
11054
11055
11056
11057
11058
11059 Reed\_Muller_6:
11060 \$\{(ORBITER_PATH)orbiter.out\}-v.2.-long\_code\_64\_7\$
11061 \$\{\text{-}set\text{-}builder\text{-}loop\_0\_64\_1.-end\}\$
11062 \$\{\text{-}set\text{-}builder\text{-}loop\_0\_32\_1.-affine\_function\_2\_1.-end\}\$
11063 \$\{\text{-}set\text{-}builder\text{-}loop\_0\_16\_1.-affine\_function\_4\_2.-clone\text{-}with\_affine\_function\_4\_3.-end\}\$
11064 \$\{\text{-}set\text{-}builder\text{-}set\text{-}builder\text{-}set\text{-}builder\text{-}loop\_0\_4\_1.-affine\_function\_1\_4.-end\}\$
11065 \$\{\text{-}set\text{-}builder\text{-}set\text{-}builder\text{-}loop\_0\_8\_1.-affine\_function\_1\_8.-end\}\$
11066 \$\{\text{-}clone\text{-}with\_affine\_function\_1\_12.-end\text{-}clone\text{-}with\_affine\_function\_1\_16.-end\}\$
11067 \$\{\text{-}end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\}\$
11068 \$\{\text{-}set\text{-}builder\text{-}set\text{-}builder\text{-}loop\_0\_8\_1.-affine\_function\_1\_8.-end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\}\$
11069 \$\{\text{-}set\text{-}builder\text{-}set\text{-}builder\text{-}clone\text{-}with\_affine\_function\_1\_48.-end\}\$
11070 \$\{\text{-}set\text{-}builder\text{-}loop\_0\_32\_1.-affine\_function\_1\_32.-end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\text{-}clone\text{-}with\_affine\_function\_1\_32.-end\\$
11071 \$\{(ORBITER_PATH)orbiter.out\}-v.2.-draw\_matrix\$
11072 \$\{\text{-}input\_csv\text{-}file}\text{-}long\_code\_genma\_n64\_k7.csv\text{-end}\$
11073 \$\{\text{-}box\_width\text{-}25\text{-}bit\_depth\text{-}8\text{-}partition\text{-}3\_7\_64\text{-}end\}\$
11074 \text{open}\_long\_code\_genma\_n64\_k7\_draw.bmp
11075
11076
11077
11078
11079
11080 RM_6:
11081 \$\{(ORBITER_PATH)orbiter.out\}-v.2.-long\_code\_64\_7\$
11082 \$\{(RM}_6\_GENERATOR\_1\}\$
11083 \$\{(RM}_6\_GENERATOR\_2\}\$
11084 \$\{(RM}_6\_GENERATOR\_3\}\$
11085 \$\{(RM}_6\_GENERATOR\_4\}\$
11086 \$\{(RM}_6\_GENERATOR\_5\}\$
11087 \$\{(RM}_6\_GENERATOR\_6\}\$
11088 \$\{(RM}_6\_GENERATOR\_7\}\$
11089 \$\{(ORBITER_PATH)orbiter.out\}-v.2.-draw\_matrix\$

692
11090 ▶ ▶ -input_csv_file_long_code_genma_n64_k7.csv\n11091 ▶ ▶ -box_width 25 -bit_depth 8 -partition 3:7:64 -end
11092 ▶ ▶ $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix\n11093 ▶ ▶ -input_csv_file_long_code_genma_n64_k7_cw0.csv\n11094 ▶ ▶ -box_width 25 -bit_depth 8 -partition 3:8:8 -end
11095 ▶ ▶ $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix\n11096 ▶ ▶ -input_csv_file_long_code_genma_n64_k7_cw1.csv\n11097 ▶ ▶ -box_width 25 -bit_depth 8 -partition 3:8:8 -end
11098 ▶ ▶ $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix\n11099 ▶ ▶ -input_csv_file_long_code_genma_n64_k7_cw2.csv\n11100 ▶ ▶ -box_width 25 -bit_depth 8 -partition 3:8:8 -end

11101
11102
11103
11104
11105 RM6words:
11106 ▶ -mkdir RM6
11107 ▶ #$L orbs orbiter.out -v 2 -draw_matrix -input_csv_file_long_code_genma_n64_k7_cw0.csv -box_width 25 -bit_depth 8 -partition 4:8:8 -end
11108 ▶ $(ORBITER_PATH)orbiter.out -v 2 -loopL 0 128 1\n11109 ▶ ▶ -draw_matrix -input_csv_file\n11110 ▶ ▶ ▶ long_code_genma_n64_k7_cw0.csv\n11111 ▶ ▶ -box_width 25 -bit_depth 8 -partition 4:8:8 -end\n11112 ▶ ▶ -mv long_code_genma_n64_k7_cw0.csv %L draw.bmp\n11113 ▶ ▶ ▶ RM6/RM_6_1_cw0.csv %L draw.bmp\n11114 ▶ ▶ -end_loop
11115
11116
11117
11118 RM6_convert:
11119 ▶ -mkdir RM6_convert\n11120 ▶ convert RM6/RM_6_1_cw0.png -frame 8:RM6_convert/000.png
11121 ▶ convert RM6/RM_6_1_cw1.png -frame 8:RM6_convert/001.png
11122 ▶ convert RM6/RM_6_1_cw2.png -frame 8:RM6_convert/002.png
11123 ▶ convert RM6/RM_6_1_cw3.png -frame 8:RM6_convert/003.png
11124 ▶ convert RM6/RM_6_1_cw4.png -frame 8:RM6_convert/004.png
11125 ▶ convert RM6/RM_6_1_cw5.png -frame 8:RM6_convert/005.png
11126 ▶ convert RM6/RM_6_1_cw6.png -frame 8:RM6_convert/006.png
11127 ▶ convert RM6/RM_6_1_cw7.png -frame 8:RM6_convert/007.png
11128 ▶ convert RM6/RM_6_1_cw8.png -frame 8:RM6_convert/008.png
11129 ▶ convert RM6/RM_6_1_cw9.png -frame 8:RM6_convert/009.png
11130 ▶ convert RM6/RM_6_1_cw10.png -frame 8:RM6_convert/010.png
11131 ▶ convert RM6/RM_6_1_cw11.png -frame 8:RM6_convert/011.png
11132 ▶ convert RM6/RM_6_1_cw12.png -frame 8:RM6_convert/012.png
11133 ▶ convert RM6/RM_6_1_cw13.png -frame 8:RM6_convert/013.png
11134 ▶ convert RM6/RM_6_1_cw14.png -frame 8:RM6_convert/014.png
11135 ▶ convert RM6/RM_6_1_cw15.png -frame 8:RM6_convert/015.png

693
# Section 10.6: Coding Theory -- BCH Codes

SECTION_CODING_THEORY_BCH_CODES:

draw.cyclotomic_mod_21_q8:

define F

with F

do finite_field_activity

make BCH_code

pdflatex BCH_codes_q8_n21_d3.tex

open BCH_codes_q8_n21_d3.pdf

generator_polynomial_is

697
\begin{verbatim}
11300 \triangleright \triangleright -with-F-do-finite_field_activity--make_BCH_code_21-4-end
11301
11302 \#generator_polynomial.is:X^\{5\}+6X^\{4\}+7X^\{3\}+2X+3
11303
11304
11305 F_8_BCH_code_d5:
11306 \triangleright $(\text{ORBITER\_PATH})\text{or}biter.out-v.3\:\backslash
11307 \triangleright -define-F--finite_field-\texttt{q}8--override_polynomial_11-end\backslash
11308 \triangleright -with-F-do-finite_field_activity--make_BCH_code_21-5-end
11309 \triangleright \text{pdf}latex-BCH_codes_q8_n21_d5.tex
11310 \triangleright open-BCH_codes_q8_n21_d5.pdf
11311
11312 \#generator_polynomial.is:X^\{7\}+3X^\{6\}+3X^\{5\}+2X^\{4\}+X^\{3\}+2X^\{2\}+X+2
11313
11314 F_8_BCH_code_d6:
11315 \triangleright $(\text{ORBITER\_PATH})\text{or}biter.out-v.3\:\backslash
11316 \triangleright -define-F--finite_field-\texttt{q}8--override_polynomial_11-end\backslash
11317 \triangleright -with-F-do-finite_field_activity--make_BCH_code_21-6-end
11318
11319
11320 \#generator_polynomial.is:X^\{9\}+5X^\{8\}+5X^\{6\}+4X^\{3\}+5X+4
11321
11322 F_8_BCH_code_d7:
11323 \triangleright $(\text{ORBITER\_PATH})\text{or}biter.out-v.3\:\backslash
11324 \triangleright -define-F--finite_field-\texttt{q}8--override_polynomial_11-end\backslash
11325 \triangleright -with-F-do-finite_field_activity--make_BCH_code_21-7-end
11326 \triangleright \text{pdf}latex-BCH_codes_q8_n21_d7.tex
11327 \triangleright open-BCH_codes_q8_n21_d7.pdf
11328
11329 \#generator_polynomial.is:X^\{10\}+X^\{9\}+2X^\{8\}+5X^\{7\}+2X^\{6\}+4X^\{4\}+6X^\{3\}+5X^\{2\}+6X+6
11330
11331 F_8_BCH_code_d8:
11332 \triangleright $(\text{ORBITER\_PATH})\text{or}biter.out-v.3\:\backslash
11333 \triangleright -define-F--finite_field-\texttt{q}8--override_polynomial_11-end\backslash
11334 \triangleright -with-F-do-finite_field_activity--make_BCH_code_21-8-end
11335
11336 \#generator_polynomial.is:X^\{12\}+2X^\{10\}+6X^\{9\}+5X^\{8\}+7X^\{7\}+6X^\{6\}+2X^\{5\}+7X^\{4\}+5X^\{3\}+5X^\{2\}+6
11337 \#k=9
11338 \#after_reduction:[63,27]_2-r=36
11339
11340
11341
11342
11343 F2_BCH_code_n21:
\end{verbatim}
$\text{ELEMENTARY SYMMETRIC}_8$:

-define-e7-formula="e7","e7","e7","e7","e7","e7","e7","e7";

-define-e8-collection="e1,e2,e3,e4,e5,e6,e7,e8";

-with-F-do-finite_field_activity:

-evaluate-E8-"x0=8,x1=64,x2=205,x3=143,x4=70,x5=217,x6=130,x7=23"-end

BCH15:

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-15-2-3}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-15-2-5}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-15-2-7}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-15-2-9}$;

BCH11:

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-11-2-3}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-11-2-5}$;

BCH13:

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-13-2-3}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-13-2-5}$;

BCH7:

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-7-2-3}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-7-2-5}$;

BCH21:

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-21-2-3}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-21-2-5}$;

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH_dual-21-2-7}$;

BCH255:

#$\text{(ORBITER PATH)orbiter.out}-\text{BCH-255-2-4}$

#$\text{(ORBITER PATH)orbiter.out}-\text{v-2-draw_matrix}$

-define-\text{\textasciitilde}\text{\textasciitilde}input_csv_file-BCH_255_4.csv;

-define-\text{\textasciitilde}\text{\textasciitilde}box_width-40-bit_depth-24;

-define-\text{\textasciitilde}\text{\textasciitilde}partition-10-"239"."255"-end
BCH273:

```
$ (ORBITER_PATH) orbiter.out -BCH:273:2.4
```

```
draw_mod_31:
$ (ORBITER_PATH) orbiter.out -v:2:\
  -draw_options -embedded -end\n  -draw_mod_n-31 mod_31 -draw_mod_n_power_cycle:2
  pdflatex mod_31 draw.tex
  open mod_31 draw.pdf
```

```
PR127:
$ (ORBITER_PATH) orbiter.out -v:5 -primitive_root:127
```

```
draw_mod_127_power:
$ (ORBITER_PATH) orbiter.out -v:2:\
  -draw_options -scale:0.4 -embedded -end\n  -draw_mod_n-127 mod_127 -draw_mod_n_power_cycle:3: 
  pdflatex mod_127 draw.tex
  open mod_127 draw.pdf
```

```
draw_mod_251:
$ (ORBITER_PATH) orbiter.out -v:2:\
  -draw_options -nodes_empty -radius:10 -embedded -end\n  -draw_mod_n-251 mod_251:\n  pdflatex mod_251 draw.tex
  open mod_251 draw.pdf
```

```
#-draw_mod_n_inverse
```

```
draw_mod_255_cyclotomic_1:
$ (ORBITER_PATH) orbiter.out -v:2:\
  -draw_options -nodes_empty -radius:10:\n  -line_width:0.4 -embedded -end:\n  -draw_mod_n-n:255 -file:mod_255_cyclotomic_1:\n  -cyclotomic_sets:2:"1" -end
  pdflatex mod_255_cyclotomic_1 draw.tex
  open mod_255_cyclotomic_1 draw.pdf
```

```
```
11485 draw_mod_255_cyclotomic_3:
11486 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11487 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11488 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11489 \texttt{-draw}\_options\texttt{-nodes}\_empty\texttt{-radius}\_10\
11490 \texttt{-line}\_width\_0.4\texttt{-embedded}\_end\texttt{-draw}\_mod\_n\_n\_255\texttt{-file}\_mod\_255\_\texttt{cyclotomic}\_3\
11491 \texttt{-draw}\_mod\_n\_n\_255\texttt{-file}\_mod\_255\_\texttt{cyclotomic}\_3\
11492 \texttt{pdflatex}\texttt{-mod}\_255\_\texttt{cyclotomic}\_3\_\texttt{draw}\_tex\
11493 \texttt{open}\texttt{-mod}\_255\_\texttt{cyclotomic}\_3\_\texttt{draw}\_pdf\
11494 draw_mod_255_cyclotomic_1_and_3:
11495 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11496 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11497 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11498 \texttt{-draw}\_options\texttt{-nodes}\_empty\texttt{-radius}\_10\
11499 \texttt{-line}\_width\_0.4\texttt{-embedded}\_end\texttt{-draw}\_mod\_n\_n\_63\texttt{-file}\_mod\_63\_\texttt{ cyclotomic}\_3\
11500 \texttt{-draw}\_mod\_n\_n\_63\texttt{-file}\_mod\_63\_\texttt{ cyclotomic}\_3\
11501 \texttt{pdflatex}\texttt{-mod}\_63\_\texttt{cyclotomic}\_1\_\texttt{and}\_3\_\texttt{draw}.\texttt{tex}\
11502 \texttt{open}\texttt{-mod}\_63\_\texttt{cyclotomic}\_1\_\texttt{and}\_3\_\texttt{draw}.\texttt{pdf}\
11503 draw_mod_63_4_cyclotomic_3_6:
11504 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11505 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11506 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.2}\
11507 \texttt{-draw}\_options\texttt{-radius}\_20\
11508 \texttt{-line}\_width\_0.1\texttt{-embedded}\_end\texttt{-draw}\_mod\_n\_n\_63\texttt{-file}\_mod\_63\_\texttt{cyclotomic}\_3\
11509 \texttt{-draw}\_mod\_n\_n\_63\texttt{-file}\_mod\_63\_\texttt{ cyclotomic}\_3\
11510 \texttt{-cyclotomic}\_sets\_4\texttt{-"3,6"}\
11511 \texttt{-cyclotomic}\_sets\_thickness\_50\
11512 \texttt{pdflatex}\texttt{-mod}\_63\_\texttt{cyclotomic}\_3\_\texttt{6}\_\texttt{draw}.\texttt{tex}\
11513 \texttt{open}\texttt{-mod}\_63\_\texttt{cyclotomic}\_3\_\texttt{6}\_\texttt{draw}.\texttt{pdf}\
11514 BCH_F_64:
11515 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.3}\
11516 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.3}\
11517 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.3}\
11518 \texttt{-define}\_F\_finite\_field\_\texttt{-q}\_64\_\texttt{-end}\
11519 \texttt{-with}\_F\_\texttt{-do}\_\texttt{-finite}\_\texttt{field}\_\texttt{activity}\_\texttt{-cheat}\_\texttt{sheet}\_GF\_\texttt{-end}\
11520 \texttt{pdflatex}\_GF\_64.\texttt{tex}\
11521 BCH_elementary_symmetric_functions_3:
11522 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out}\_\texttt{-make}\_\texttt{elementary}\_\texttt{symmetric}\_\texttt{functions}\_\texttt{3}\_\texttt{3}\
11523 BCH_63_4_evaluate_elementary_symmetric_functions_1:
11524 \$\{\textbf{ORBITER} PATH\}\texttt{orbiter.out -v.3}\_\texttt{-define}\_F\_\texttt{-finite}\_\texttt{field}\_\texttt{-q}\_64\_\texttt{-end}\
11525 \texttt{-define}\_e1\_\texttt{-formula}\_\"e1\"\_\texttt{-e1\"}$\{\texttt{ELEMENTARY}\_\texttt{SYMMETRIC}\_3\_\texttt{1}\_\texttt{-end}\texttt{-define}\_e1\_\texttt{-formula}\_\"e1\"\_\texttt{-e1\"}$\{\texttt{ELEMENTARY}\_\texttt{SYMMETRIC}\_3\_\texttt{2}\_\texttt{-end}\texttt{-define}\_e3\_\texttt{-formula}\_\"e3\"\_\texttt{-e3\"}$\{\texttt{ELEMENTARY}\_\texttt{SYMMETRIC}\_3\_\texttt{3}\_\texttt{-end}\texttt{-define}\_E3\_\texttt{-collection}\_\"e1,e2,e3\"}\
11531 \texttt{-with}\_F\_\texttt{-do}\_\texttt{-finite}\_\texttt{field}\_\texttt{activity}\_\texttt{-}}
The values of the formulae are:

0: 57
1: 0
2: 1

poly: 1,0,2,1

BCH_63_4_evaluate_elementary_symmetric_functions_2:

BCH_21_poly_mult_mod_F4:

BCH_21_poly_division_a:
BCH\_21\_poly\_division\_b:
\$\{{\text{ORBITER\_PATH}}\}\text{orbiter.out}-v.2\$
\define F \text{-finite_field-}q.4\text{-end}\$
\with F \text{-do}\$
\define F \text{-finite_field_activity}\$
\define F \text{-polynomial_division}\$
"1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1"
"1,0,2,1"\text{-end}
BCH\_21\_poly\_division\_ab:
\$\{{\text{ORBITER\_PATH}}\}\text{orbiter.out}-v.2\$
\define F \text{-finite_field-}q.4\text{-end}\$
\with F \text{-do}\$
\define F \text{-finite_field_activity}\$
\define F \text{-polynomial_division}\$
"1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1"
"1,0,3,1"\text{-end}
BCH\_21\_generator\_matrix:
\$\{{\text{ORBITER\_PATH}}\}\text{orbiter.out}-v.2\$
\define F \text{-finite_field-}q.4\text{-end}\$
\with F \text{-do}\$
\define F \text{-generator_matrix_cyclic_code.21-}1,0,1,0,1,1,1\text{-end}
BCH\_21\_15\_weight\_enumerator:
\$\{{\text{ORBITER\_PATH}}\}\text{orbiter.out}-v.2\$
\define F \text{-finite_field-}q.4\text{-end}\$
\define v \text{-vector-field.F-format.15}\$
\dense $(BCH\_21\_15\_GENERATOR\_MATRIX)\$
\end$
\with F \text{-do}\$
\define F \text{-finite_field_activity-}weight\_enumerator-v.\text{-end}
#too\_slow!
BCH\_21\_15\_dual:
11626 \>$\$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}-v\cdot2\$
11627 \>$\define F\cdot\text{finite\_field}\cdot q\cdot4\cdot\text{-end}\$
11628 \>$\define v\cdot\text{-vector}\cdot\text{-field}\cdot F\cdot\text{-format}\cdot 15\cdot\text{-end}\$
11629 \>$\>\>\define F\cdot\text{BCH}\_21\_15\_\text{GENERATOR\_MATRIX}\cdot\text{-end}\$
11630 \>$\define F\cdot\text{-do}\cdot\text{-finite\_field\_activity}\$
11631 \>$\define F\cdot\text{-nullspace\_v}\$
11632 \>$\define F\cdot\text{-normalize\_from\_the\_right}\$
11633 \>$\>\>\text{-end}\$
11634
11635
11636 \text{BCH}\_21\_6\_\text{weight\_enumerator}:  
11637 \>$\$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}-v\cdot2\$
11638 \>$\define F\cdot\text{finite\_field}\cdot q\cdot4\cdot\text{-end}\$
11639 \>$\define v\cdot\text{-vector}\cdot\text{-field}\cdot F\cdot\text{-format}\cdot 6\cdot\text{-field}\cdot F\$
11640 \>$\>\>\define F\cdot\text{BCH}\_21\_6\_\text{GENERATOR\_MATRIX}\cdot\text{-end}\$
11641 \>$\define F\cdot\text{-do}\cdot\text{-finite\_field\_activity}\cdot\text{-weight\_enumerator}\cdot v\cdot\text{-end}\$
11642
11643
11644
11645 \#1y^21\cdot63x^8y^{13}\cdot294x^{12}y^9\cdot756x^{14}y^7\cdot1890x^{16}y^5\cdot1092x^{18}
11646
11647 \#(1,0,0,0,0,0,63,0,0,0,294,0,756,0,1890,0,1092,0,0,0)
11648
11649
11650
11651 \text{BCH}\_21\_6\_4\_\text{macwilliams}:  
11652 \>$\$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}-v\cdot2\cdot\text{-make\_macwilliams\_system}\cdot 21\cdot6\cdot4$
11653 \>$\text{pdflatex}\cdot\text{MacWilliams}\_n21\_k6\_q4\cdot\text{tex}$
11654 \>$\text{open}\cdot\text{MacWilliams}\_n21\_k6\_q4\cdot\text{pdf}$
11655
11656
11657
11658 \#ww:=\{1,0,0,84,252,1575,10080,58032,319662,1411116,5133744,15282792,  
11659 
11660
11661 \text{BCH}\_21\_15\_4\_\text{field\_reduction}:  
11662 \>$\$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}-v\cdot2\$
11663 \>$\define F\cdot\text{finite\_field}\cdot q\cdot4\cdot\text{-end}\$
11664 \>$\define F\cdot\text{-do}\$
11665 \>$\>\>\text{-finite\_field\_activity}\$
11666 \>$\>\>\text{-field\_reduction}\cdot\text{BCH}\_21\_15\_4\cdot21\cdot 21\cdot\$(\text{BCH}\_21\_15)\cdot\text{-end}\$
11667 \>$\$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}-v\cdot2\$
11668 \>$\>\>\text{-draw\_matrix}\cdot\text{-input\_csv\_file}\cdot\text{BCH}\_21\_15\_4\cdot\text{csv}$
11669 \>$\>\>\>\text{-box\_width}\cdot 20\cdot\text{-bit\_depth}\cdot 24$
11670 ▶ ▶ -partition=4:"30"."42".-end
11671 ▶ pdflatex field_reduction_Q4_q2_15_21.tex
11672 ▶ open-BCH_21_15_4_draw.bmp
11673 ▶ #open-field_reduction_Q4_q2_15_21.pdf
11674
11675 #poly.of.degree:12:1,0,1,0,1,0,0,0,1,0,0,0,1
11676
11677 BCH_21_poly_division_c:
11678 ▶ $(ORBITER_PATH)orbiter.out.-v.2`
11679 ▶ ▶ -define:F=.finite_field.-q.2.-end`
11680 ▶ ▶ -with:F.-do`
11681 ▶ ▶ -finite_field_activity`
11682 ▶ ▶ -polynomial_division="1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
11683 ,0,0,0,0,0,0,0,0,0,1"."1,0,1,0,1,0,0,0,1,0,0,0,1".-end
11684
11685 F16.roots_5:
11686 ▶ $(ORBITER_PATH)orbiter.out.-v.3`
11687 ▶ ▶ -define:F=.finite_field.-q.2.-end`
11688 ▶ ▶ -with:F.-do.-finite_field_activity.-nth_roots_5.-end
11689
11690
11691
11692 F64.roots_21:
11693 ▶ $(ORBITER_PATH)orbiter.out.-v.3`
11694 ▶ ▶ -define:F=.finite_field.-q.2.-end`
11695 ▶ ▶ -with:F.-do.-finite_field_activity.-nth_roots_21.-end
11696
11697
11698
11699 F256.roots_771:
11700 ▶ $(ORBITER_PATH)orbiter.out.-v.3`
11701 ▶ ▶ -define:F=.finite_field.-q.256.-end`
11702 ▶ ▶ -with:F.-do.-finite_field_activity.-nth_roots_771.-end
11703
11704
11705
11706 F256.BCH_code_d16:
11707 ▶ $(ORBITER_PATH)orbiter.out.-v.3`
11708 ▶ ▶ -define:F=.finite_field.-q.256.-end`
11709 ▶ ▶ -with:F.-do.-finite_field_activity.-make.BCH_code.771.16.-end
11710 ▶ pdflatex BCH_codes_q256_n771_d16.tex
11711 ▶ open-BCH_codes_q256_n771_d16.pdf
11712
11713 #generator.polynomial.is:X^-{30}+253X^-{29}+174X^-{28}+109X^-{27}+97X^-{26}+144X^-{25}+112X^-{24}+212X^-{23}+192X^-{22}+169X^-{21}+24X^-{20}+150X^-{19}+110X^-{18}+248X^-{17}+3X^-{16}+193X^-{15}+194X^-{14}+205X^-{13}+9X^-{12}+
56X^{11} + 95X^{10} + 199X^9 + 108X^8 + 58X^7 + 160X^6 + 148X^5 + 138X^4 + 24X^3 + 210X^2 + 26X + 1

F256_BCH_code_d16_division:
   $(ORBITER_PATH)orbiter.out -v 2 \
   -define F -finite_field -q 256 -end \
   -define A -vector -field F -sparse 772 "1,771,1,0" -end \
   -define B -vector -field F -dense $(POLY_Q256_DEG30_DENSE) -end \
   -with F -do \
   -finite_field_activity \
   -polynomial_division A B -end

F256_BCH_write_code_for_division_d16:
   $(ORBITER_PATH)orbiter.out -v 2 \
   -define F -finite_field -q 256 -end \
   -define A -vector -field F -sparse 772 "1,771,1,0" -end \
   -define B -vector -field F -dense $(POLY_Q256_DEG30_DENSE) -end \
   -with F -do \
   -finite_field_activity \
   -write_code_for_division-check_q256_n771_r30.cpp A B -end

   g++ check_q256_n771_r30.cpp -o check_q256_n771_r30.out \
   ./check_q256_n771_r30.out

F256_BCH_code_d16_error:
   $(ORBITER_PATH)orbiter.out -v 2 \
   -define F -finite_field -q 256 -end \
   -define A -vector -field F -sparse 771 "2,30,3,31,55,770" -end \
   -define B -vector -field F -dense $(POLY_Q256_DEG30_DENSE) -end \
   -with F -do \
   -finite_field_activity \
   -polynomial_division A B -end

SECTION CODING THEORY REED SOLOMON CODES:

#ToDo:
11759  F_7_BCH_code.n6:
11760  $(ORBITER_PATH)orbiter.out-v.3:\
11761  >  -define F-finite_field-q7-end:
11762  >  -with F-do-finite_field_activity-make_BCH_code.7.3-end
11763
11764
11765
11766  RREF_RS_6_4_7_weight Enumerator:
11767  $(ORBITER_PATH)orbiter.out-v.2:\n11768  >  -define F-finite_field-q7-end:
11769  >  -define v-vector-format.4-field F:\
11770  >  -compact $(CODE_RS_6_4_7)\:
11771  >  -end:\
11772  >  -with F-do:\
11773  >  -finite_field_activity:\
11774  >  -weight Enumerator-v\:
11775  >  -end
11776
11777
11778
11779
11780
11781
11782
11783  Code_RS_11:
11784  $(ORBITER_PATH)orbiter.out-v.2:\
11785  >  -define F-finite_field-q11-end:\
11786  >  -define v-vector-format.8-field F:\
11787  >  -compact $(CODE_RS_11_RREF)\:
11788  >  -end:\
11789  >  -with F-do:\
11790  >  -finite_field Activity-RREF.v-end
11791  pdfLaTeX RREF_example_q11_8_10.tex
11792  #gs-sDEVICE=png16-dFIXEDMEDIA-dDEVICEWIDTHPOINTS=500-dDEVICEHEIGHTPOINTS=450-
11793  #r240-oRREF_example_q11_8_10_page%02d.png\]
11794  #o RREF_example_q11_8_10.pdf
11795  open RREF_example_q11_8_10.pdf
11796
11797
11798  Code_RS_11_weight Enumerator:
11799  $(ORBITER_PATH)orbiter.out-v.2:\
11800  >  -define F-finite_field-q11-end:\
11801  >  -define v-vector-format.8-field F:\
11802  >  -compact $(CODE_RS_11_RREF)\:
11803  >  -end:\
11804  >  -with F-do:\

708
#1*y^(10) + 1200*x^3*y^7 + 16800*x^4*y^6 + 209160*x^5*y^5 + 1734600*x^6*y^4 + 9918000*x^7*y^3 + 37189800*x^8*y^2 + 82644700*x^9*y + 82644620*x^(10)
11851  
11852  
11853  
11854  
11855  
11856 CODE_21_15_4_store:
11857  
11858  
11859  
11860  
11861  
11862  
11863  
11864  
11865  
11866 CODE_21_15_4_weight Enumerator:
11867  
11868  
11869  
11870  
11871  
11872  
11873  
11874  
11875  
11876  
11877  
11878  
11879 Reed_solomon_F8_work:
11880  
11881  
11882  
11883  
11884  
11885  
11886  
11887 # Section 10.8: Coding Theory -- Bounds
11888  
11889 SECTION_CODING_THEORY_BOUNDS:
11890  
11891 bounds for d given n6.k4.q7:
11892  
11893  
11894  
11895 bounds for d given n15.k6.q2:
11896  
11897  

coding_theory_bounds_q2:

GV
n15
k6
d5:

GV
n15
k6
d5: # [15,6].code.created

bounds_for_d_given_n12_k4_q13:

GV
n15
k6
d5_weight Enumerator:
\[ y^{15} + 27x^6y^9 + 24x^8y^7 + 9x^5 + 3x^{12}y^3 \]

\[ d = 6 \]

\[ \text{code n15.k6.d6.a we:} \]
\[ \text{#1y}\{15\} + 27x^6y^9 + 24x^8y^7 + 9x^5 + 3x^{12}y^3 \]

\[ \text{code n15.k6.d6.RREF:} \]
\[ \text{#1y}\{15\} + 27x^6y^9 + 24x^8y^7 + 9x^5 + 3x^{12}y^3 \]

\[ \text{code n15.k6.d6.check.RREF:} \]
\[ \text{#1y}\{15\} + 27x^6y^9 + 24x^8y^7 + 9x^5 + 3x^{12}y^3 \]
Section 10.9: Coding Theory - Classification

SECTION CODING THEORY CLASSIFICATION:

# Code classification:

codes_8_4_4:

$\text{(ORBITER_PATH)}\text{orbiter.out}$ -v 3

- $\text{orбирter_path:$(ORBITER_PATH)}$
- $\text{-define-G}$
- $\text{-linear_group:-PGL.4.2.-end}$
- $\text{-with-G-do}$
- $\text{-group_theoretic_activity}$
- $\text{-poset_classification_control}$
- $\text{-problem_label:codes_8_4_4}$
- $\text{-draw_poset}$
- $\text{-draw_options:-embedded:-radius-250}$
- $\text{-line_width:1.0:-spanning_tree:-end}$
- $\text{-report:-end}$
- $\text{-end}$

linear_codes_3_8

-end

pdflatex codes_8_4_4_poset_lvl_8.tex
open-codes_8_4_4_poset_lvl_8.pdf
pdflatex codes_8_4_4_poset.tex
open-codes_8_4_4_poset.pdf
codes_8_4_4_draw:

draw_layered_graph

codes_8_4_4_poset_lvl_8.layered_graph
-radius 250 -embedded -line_width 1.0
-y_stretch 1.0 -scale 0.5
-end

pdflatex codes_8_4_4_poset_lvl_8_draw.tex
open-codes_8_4_4_poset_lvl_8_draw.pdf
codes_14_4_9_3:
12039 $(\text{ORBITER PATH})\text{orbiter.out}$-\text{v-6}\$
12040 \text{define-G}$
12041 \text{-linear_group}-\text{PGL-10:3}-\text{-end}$
12042 \text{-with-G-do}$
12043 \text{-group_theoretic_activity}$
12044 \text{-poset_classification_control}$
12045 \text{-problem_label-codes_14_4_9_3}$
12046 \text{-draw_poset}$
12047 \text{-draw_options}$
12048 \text{-embedded-radius-250}$
12049 \text{-end}$
12050 \text{-end}$
12051 \text{-linear_codes-9-14}$
12052 \text{-end}$
12053 \text{pdflatex-codes_14_4_9_3_poset_lvl_13.tex}$
12054 \text{open-codes_14_4_9_3_poset_lvl_13.pdf}$
12055
12056
12057 \text{codes_15_6_6_2}$:
12058 $(\text{ORBITER PATH})\text{orbiter.out}$-\text{v-6}\$
12059 \text{define-G}$
12060 \text{-linear_group}-\text{PGL-9:2}-\text{-end}$
12061 \text{-with-G-do}$
12062 \text{-group_theoretic_activity}$
12063 \text{-poset_classification_control}$
12064 \text{-problem_label-codes_15_6_6_2}$
12065 \text{-draw_poset}$
12066 \text{-draw_options}$
12067 \text{-embedded-radius-250}$
12068 \text{-end}$
12069 \text{-end}$
12070 \text{-linear_codes-6-15}$
12071 \text{-end}$
12072 \text{pdflatex-codes_15_6_6_2_poset_lvl_15.tex}$
12073 \text{open-codes_15_6_6_2_poset_lvl_15.pdf}$
12074
12075
12076
12077 \text{codes_16_5_9_3}$:
12078 $(\text{ORBITER PATH})\text{orbiter.out}$-\text{v-6}\$
12079 \text{-codes_classify-n-16-k-5-q-3-d-9-w-W-lex}$
12080 \text{-draw_poset}$
12081 \text{-end}$
12082
12083
12084 \text{#-5/31/2020: 28 min 22 sec on Mac}$
12085

12086
12087
12088
12089
12090
12091
12092
12093
12094
12095
12096
12097
12098
12099
12100
12101
12102
12103
12104
12105
12106
12107
12108
12109
12110
12111
12112
12113
12114
12115
12116
12117
12118
12119
12120
12121
12122
12123
12124
12125
12126
12127
12128
12129
12130
12131
12132
12133
12086 #0::1-orbits
12087 #1::1-orbits
12088 #2::1-orbits
12089 #3::1-orbits
12090 #4::1-orbits
12091 #5::1-orbits
12092 #6::1-orbits
12093 #7::1-orbits
12094 #8::1-orbits
12095 #9::2-orbits
12096 #10::3-orbits
12097 #11::4-orbits
12098 #12::5-orbits
12099 #13::5-orbits
12100 #14::4-orbits
12101 #15::3-orbits
12102 #16::1-orbits
12103 #total::36
12104
12105
12106 codes_d4:
12107   $(ORBITER_PATH)orbiter.out-v:3:\n12108   -define-G-lineargroup=\PGL\4\2-end:\n12109   -with-G-do:\n12110   -group_theoretic_activity:\n12111   -poset_classification_control=w:\n12112   -problem\_label\_codes_r4_d4\-\draw\_poset:\n12113   -embedded\-end\-\linear\_codes\4\cdot100:\n12114   -end:\n12115   -end\n12116
12117
12118 codes_24_12_8:
12119   $(ORBITER_PATH)orbiter.out-v:6:\n12120   -orbiter_path=$(ORBITER_PATH)\n12121   -define-G\n12122   -linear_group=\PGL\12\2-end\n12123   -with-G-do\n12124   -group_theoretic_activity\n12125   -poset\_classification\_control\n12126   -\-problem\_label\_codes\_24\_12\_8\n12127   -\-\draw\_poset\n12128   -\-\draw\_options\-\embedded\-radius\_250\n12129   -\-\line\_width\_1.0\-\spanning\_tree\-end\n12130   -\-\report\-end\n12131   -\-\end\n12132   -\-linear\_codes\_8\cdot24\n
715
#codes.24.12_8_poset_lvl.24.layered_graph

codes.24.12_8_draw:

\$\langle ORBITER\ PATH\rangle orbiter.out-v.3.\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{draw_layered_graph}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{codes.24.12_8_poset_lvl.24.layered_graph}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-radius.100,-spanning_tree,-embedded}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-line_width.0.5,-x_stretch.1.4}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-scale.0.25,-nodes_empty}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-end}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{pdf_latex}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{open-codes.24.12_8_poset_lvl.24_draw.pdf}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{glynn_arc}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-v.5}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-orbiter_path}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-define-G}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-linear_group-PGL.5.9}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-with-G-do}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-group_theoretic_activity}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-poset_classification_control}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-problem_label-glynn_arc}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-draw_options:-embedded,-radius.250}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-line_width.1.0,-spanning_tree,-end}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-draw_poset}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-report,-end}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-end}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-linear_codes.6.10}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-end}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{pdf_latex-glynn_arc_poset.tex}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{open-glynn_arc_poset.pdf}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{five_points_in_general}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-v.5}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-orbiter_path}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-define-G}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-linear_group-PGL.4.2}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-with-G-do}\$

\$\langle ORBITER\ PATH\rangle \rightarrow \text{-group_theoretic_activity}\$
\textit{poset classification control}
\textit{-problem_label:five_points_in_general}
\textit{-draw_options:/embedded/-radius:250
\textit{-line_width:1.0/-spanning_tree/-end
\textit{-draw_poset
\textit{-report/-end
\textit{-end
\textit{-linear_codes:4-5
\textit{-end
\textit{pdflatex:five_points_in_general_poset.tex
\textit{open:five_points_in_general_poset.pdf
\textit{codes_q13_12_4:
\textit{$(ORBITER\ PATH)\ orbiter.out\ -v:6$
\textit{-orbiter_path:$(ORBITER\ PATH)
\textit{-define:G
\textit{-linear_group:-PGL:4-13/-end
\textit{-with:G:-do
\textit{-group_theoretic_activity
\textit{-poset_classification_control
\textit{-problem_label:codes_q13
\textit{-report/-end
\textit{-end
\textit{-linear_codes:6-12
\textit{-end
\textit{pdflatex:codes_q13_poset.tex
\textit{open:codes_q13_poset.pdf
\textit{codes_q13_12_4:
\textit{Sym_4_conj_classes:
\textit{Section\ 11.1::Combinatorics
\textit{Chapter\ 11::Combinatorics
\textit{Section\ 11.1::Combinatorics
\textit{SECTION\_COMBINATORICS::
\textit{Sym_4_conj_classes::}
# $(ORBITER_PATH)orbiter.out-v.2-conjugacy_classes_Sym_n-4

Sym_10_conj_classes:

Sym_10_conj_classes:

Sym_15_conj_classes:

Sym_15_conj_classes:

Char_Sym_4:

Char_Sym_5:

Char_Sym_6:

Char_Sym_6:

all_subsets_10.3:

all_subsets_10.3:

rank_k_subsets_test:

rank_k_subsets_test:

Walsh_matrix_4:

Walsh_matrix_4:

Dedekind_10_10:

Dedekind_10_10:

Dedekind_30_2:

Dedekind_30_2:

Walsh_01_4.csv

Walsh_01_4.csv

# pdflatex GF_2.tex

# open GF_2.pdf

718
Dedekind_100_2:
$\$(ORBITER\_PATH)\$orbiter.out\_v.3\_Dedekind\_numbers.2\_100\_2.2$

elementary_symmetric_functions_4:
$\$(ORBITER\_PATH)\$orbiter.out\_make_elementary_symmetric_functions.4.4$

elementary_symmetric_functions_8:
$\$(ORBITER\_PATH)\$orbiter.out\_make_elementary_symmetric_functions.8.8$

#large-sets:

GENERATORS_H5="1,2,3,4,0,6,7,8,9,5,10,11,12"
#(0,1,2,3,4)(5,6,7,8,9)

GENERATORS_C13="11,0,10,12,5,3,7,4,2,8,6,9,1"
#(0,11,9,8,2,10,6,7,4,5,3,12,1)

LARGE_SET_S0="0,1,2,3,4,5,6,7,8,9,10,11,12"
#-identity

LARGE_SET_S1="6,8,9,2,7,10,1,11,0,3,5,4,12"
#(0,6,1,8)(2,9,3)(4,7,11)(5,10),&\$

LARGE_SET_S2="2,0,1,6,3,4,11,5,7,8,10,9,12"
#(0,2,1)(3,6,11,9,8,7,5,4),\$

LARGE_SET_S3="12,5,6,11,3,7,10,8,9,1,4,2,0"
#(0,12)(1,5,7,8,9)(2,6,10,4,3,11),&\$

LARGE_SET_S4="5,8,10,3,11,0,2,1,12,4,6,7,9"
#(0,5)(1,8,12,9,4,11,7)(2,10,6),\$
file_S:
> echo -ROW,CO"n0,"$(LARGE_SET_S0)"n1,"$(LARGE_SET_S1)"n2,"$(LARGE_SET_S2)"n3,"$(LARGE_SET_S3)"n4,"$(LARGE_SET_S4)"n5,"$(LARGE_SET_S5)"n6,"$(LARGE_SET_S6)"n7,"$(LARGE_SET_S7)"n8,"$(LARGE_SET_S8)"n9,"$(LARGE_SET_S9)"n10,"$(LARGE_SET_S10)"nEND"n" > S.csv

Large_set_H5:
> $(ORBITER_PATH)orbiter.out -v 10\  
> -define G=permutation_group=symmetric_group=13\  
> -subgroup_by_generators H5=5-1:$GENERATORS_H5\  
> -with-G=do\  
> -group_theoretic_activity\  
> -report\  
> -end\  
> -with-G=do\  
> -group_theoretic_activity\  
> -save_elements_csv="H5_elts.csv"\  
> -end

> pdflatex Perm13_Subgroup_H5_5_report.tex
> open Perm13_Subgroup_H5_5_report.pdf

Large_set_C13:
Large set mult $C_{13} \times S$:

Large set mult $C_{13} \times S \times H_5$: 
Large set multi-C13xSxH5 apply:

$(ORBITER\ PATH)orbiter.out \ -v\ 10$

-define G permutation_group -symmetric_group 13 -end

-with G -do

-define elements_csv to set

-apply elements_csv to set

C13xSxH5.csv

C13xSxH5.csv

C13xSxH5_images.csv

"0,1,2,3"

-end

domino portrait:

$(ORBITER\ PATH)orbiter.out \ -v\ 3\ -domino_portrait 7-4 anton 28x32 -end

domino portrait input:

$(ORBITER\ PATH)orbiter.out \ -v\ 2$

-define all one r vector -repeat 1 28 -end

-define all one c vector -repeat 1 32 -end

-draw_matrix-

-grayscale-

-invert_colors-

-input_csv_file anton 28x32 m.csv

-box_width 20 -bit_depth 8-

-partition 3

-all_one_c all_one_r

-end

open anton 28x32 m draw.bmp

SECTION_DIOPHANT:

part10:

$(ORBITER\ PATH)orbiter.out \ -v\ 4$

-define A vector dense "10,9,8,7,6,5,4,3,2,1" -end

-define D diophant

-label part10

-coefficient_matrix A
octic_monomials:
\$(ORBITER\ PATH)orbiter.out-v.4\$
\$\define A -vector -dense "1,1,1,1" -end\$
\$\define D -diophant\$
\$\label -octic_monomials\$
\$\coefficient_matrix -A\$
\$\RHS "8,8,1"\$
\$\x_min\_global -0 -x_max\_global -8\$
\$\end\$
\$\with -D -do\$
\$\diophant_activity -solve\_mckay\$
\$\end\$
\$\sort -r -octic_monomials.sol > octic_monomials_sorted.txt\$

\#Found 165 solutions with 210 backtrack steps
\#165=binomial(11,3)

\$\define A -vector -format 7 -dense $(TEST\ SYSTEM)$ -end\$
\$\define D -diophant\$
\$\label -test\ system\$
\$\coefficient_matrix -A\$
\$\RHS $(TEST\ RHS)$\$
\$\x_min\_global -0 -x_max\_global -1\$
\$\end\$
\$\with -D -do\$
\$\diophant_activity -solve\_mckay\$
\$\end\$
12506 \> \> \> \> -coefficient_matrix.A.shell\n12507 \> \> \> \> -RHS.(TEST.RHS).shell\n12508 \> \> \> \> -x_min_global.0.-x_max_global.1.shell\n12509 \> \> \> -end.shell\n12510 \> \> \> -with.D.-do.shell\n12511 \> \> \> \> -diophant_activity.-solve.mckay.shell\n12512 \> \> \> -end.shell\n12513
12514 DLX_test:\n12515 \> $(ORBITER_PATH)orbiter.out.v.4.shell\n12516 \> \> -define.A.-vector.-format.7.-dense.$(TEST_SYSTEM).-end.shell\n12517 \> \> -define.D.shell\n12518 \> \> -diophant.-label.test_system.shell\n12519 \> \> \> -coefficient_matrix.A.shell\n12520 \> \> \> -RHS.$(TEST.RHS).shell\n12521 \> \> \> -x_min_global.0.-x_max_global.1.shell\n12522 \> \> \> -end.shell\n12523 \> \> \> -with.D.-do.shell\n12524 \> \> \> \> -diophant_activity.-solve.DLX.shell\n12525 \> \> \> -end.shell\n12526
12527 #DLX_test.sol
12528 #1:solution.in.6:backtrack.steps
12529
12530
12531
12532
12533 ############################################################################################################################
12534 #Section.11.3:-Combinatorial.Linear.Spaces
12535
12536 SECTION_COMBINATORIAL_LINEAR_SPACES:
12537
12538
12539 linsp6:
12540 \> $(ORBITER_PATH)orbiter.out.v.4.shell\n12541 \> \> -define.A.-vector.-format.1.-dense."15,10,6,3,1".-end.shell\n12542 \> \> -define.D.-diophant.-label.linsp6.shell\n12543 \> \> -coefficient_matrix.A.shell\n12544 \> \> -RHS."15,15,1".shell\n12545 \> \> -x_min_global.0.shell\n12546 \> \> -x_max_global.15.shell\n12547 \> \> -end.shell\n12548 \> \> -with.D.-do.shell\n12549 \> \> \> -diophant_activity.-solve.mckay.shell\n12550 \> \> -end.shell\n12551 \> #Found.15:solutions.with.22:backtrack.steps
linsp7:

$(ORBITER\_PATH)orbiter.out -v 4 \\
define A -vector -format 1 -dense "21,15,10,6,3,1" -end \\
define D -diophant -label linsp7 \\
define A -vector -format 1 -dense "21,15,10,6,3,1" -end \\
define D -diophant -label linsp7 \\
coefficient matrix A \\
RHS "21,21,1" \\
x_min_global 0 \\
x_max_global 21 \\
end \\
with D -do \\
diophant_activity -solve mckay \\
end \\

# 32 solutions in 45 backtrack steps

linsp30 pt_types:

$(ORBITER\_PATH)orbiter.out -v 4 \\
define A -vector -format 1 -dense "6,4,3" -end \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
define D -diophant \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
define D -diophant \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
define D -diophant \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
RHS "29,29,1" -x_bounds "0,1,0,27,0,24" \\
end \\
with D -do \\
diophant_activity -solve mckay \\
end \\

linsp30 pt_distribution:

$(ORBITER\_PATH)orbiter.out -v 4 \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
define D -diophant \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
define D -diophant \\
define A -vector -format 6 -dense \\
"1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21" \\
end \\
RHS "30,30,1,7,7,1,135,135,1,96,96,1,0,351,2,0,276,2" \\

725
# Section 11.4: Combinatorial Linear Spaces

```bash
geo_passch:
  $(ORBITER_PATH)orbiter.out -v 8
  -define Test_lines_set -loop 1.7.1 -end
  -define Geo -geometry_builder
  -V 6 -B 4 -TDO 2 -fuse 1
  -fname GEO_passch
  -test Test_lines
  -end

geo_petersen:
  $(ORBITER_PATH)orbiter.out -v 8
  -define Test_lines_set -loop 3.11.1 -end
  -define Geo -geometry_builder
  -V 10 -B 15 -TDO 3 -fuse 1
  -fname GEO_petersen -girth 5
  -search_tree
  -test Test_lines
  -end

geo_7.3:
  $(ORBITER_PATH)orbiter.out -v 2
  -define Test_lines_set -loop 3.8.1 -end
  -define Geo -geometry_builder
  -V 7 -B 7 -TDO 3
  -fuse 1 -fname GEO_7.3
  -test Test_lines
  -end
```
geo_7_3_no_square_test:
  $(ORBITER_PATH)orbiter.out -v 2 \n  -define Test_lines -set -loop 3 -end \n  -define Geo geometry_builder \n  -V 7 -B 7 -TDO 3 \n  -f fuse 1 -fname GEO 7_3_nst \n  -test Test_lines \n  -no_square_test \n  -end

geo_7_3_no_square_test_draw:
  $(ORBITER_PATH)orbiter.out -v 10 \n  -draw incidence structure description \n  -width 60 -with 10 6 -end \n  -define C combinatorial_objects \n  -file of incidence geometries 7_3_nst.inc 7_7 21 \n  -end \n  -with C -do \n  -combinatorial object activity \n  -draw incidence matrices \n  -7_3_nst \n  -end

pdflatex 7_3_nst_incma.tex
open 7_3_nst_incma.pdf

geo_7_3_orderly:
  $(ORBITER_PATH)orbiter.out -v 200 \n  -define Test_lines -set -loop 3 -end \n  -define Geo geometry_builder \n  -V 7 -B 7 -TDO 3 \n  -f fuse 1 -fname GEO 7_3 \n  -test Test_lines \n  -search tree \n  -orderly \n  -end

geo_7_3_orderly_draw:
  $(ORBITER_PATH)orbiter.out -v 20 \n  -draw options -embedded -radius 50 \n  -xin 10000 -yin 10000 \n  -xout 100000 -yout 100000 \n  -nodes empty \n  -scale 0.5 -line width 0.3 \n  -end

727
12694 ▶ ▶ -tree_draw·file·7_3_tree.txt·end
12695 ▶ pdflatex·7_3_tree_draw.tex
12696 ▶ open·7_3_tree_draw.pdf
12697
12698 geo_7_3_orderly_mem_debug:
12699 ▶ $(ORBITER_PATH)orbiter.out·-v·20·\n12700 ▶ ▶ -memory_debug·2·\n12701 ▶ ▶ ▶ -define·Test_lines·-set·-loop·3·8·1·-end·\n12702 ▶ ▶ -define·Geo·-geometry_builder·\n12703 ▶ ▶ ▶ -V·7·-B·7·-TDO·3·\n12704 ▶ ▶ ▶ -fuse·1·-fname_GEO·7_3·\n12705 ▶ ▶ ▶ -test·Test_lines·\n12706 ▶ ▶ ▶ -search_tree·\n12707 ▶ ▶ ▶ -orderly·\n12708 ▶ ▶ -end
12709
12710 geo_8_3:
12711 ▶ $(ORBITER_PATH)orbiter.out·-v·2·\n12712 ▶ ▶ -define·Test_lines·-set·-loop·3·9·1·-end·\n12713 ▶ ▶ -define·Geo·-geometry_builder·\n12714 ▶ ▶ ▶ -V·8·-B·8·-TDO·3·\n12715 ▶ ▶ ▶ -fuse·1·-fname_GEO·8_3·\n12716 ▶ ▶ ▶ -test·Test_lines·\n12717 ▶ ▶ -end
12718
12719 #-print_at_line·4
12720 #·1·geo·:·0·-11·-18·-29·-30·-38·-44·-54·
12721 #·ago·=48
12722
12723
12724
12725
12726
12727 geo_9_3:
12728 ▶ $(ORBITER_PATH)orbiter.out·-v·2·\n12729 ▶ ▶ -define·Test_lines·-set·-loop·3·10·1·-end·\n12730 ▶ ▶ -define·Geo·-geometry_builder·\n12731 ▶ ▶ ▶ -V·9·-B·9·-TDO·3·\n12732 ▶ ▶ ▶ -fuse·1·-fname_GEO·9_3·\n12733 ▶ ▶ ▶ -test·Test_lines·\n12734 ▶ ▶ -end
12735
12736
12737 geo_10_3:
12738 ▶ $(ORBITER_PATH)orbiter.out·-v·2·\n12739 ▶ ▶ -define·Test_lines·-set·-loop·4·11·1·-end·\n12740 ▶ ▶ -define·Geo·-geometry_builder·\n
728
12741   -V.10-B.10-TDO.3-fuse.1\n12742   -fname_GEO.10.3\n12743   -test-Test_lines\n12744   -end
12745
12746
12747
12748  #.10:geos
12749  #.8/26/2021:.0:sec:on:Mac
12750
12751
12752 geo_10.3_inc_draw:
12753   $(ORBITER_PATH)orbiter.out-v.10:\n12754   -draw.incidence_structure_description\n12755   -width.60-with.10.6.-end\n12756   -define-C-combinatorial_objects\n12757   -file_of_incidence_geometries\n12758   -10.3.inc.10.10.30\n12759   -end\n12760   -with.C-do\n12761   -combinatorial_object_activity\n12762   -draw_incidence_matrices\n12763   -10.3.inc\n12764   -end
12765   pdflatex.10.3_inc_incma.tex
12766   open.10.3_inc_incma.pdf
12767
12768
12769 geo_10.3_orderly:
12770   $(ORBITER_PATH)orbiter.out-v.20:\n12771   -define-Test_lines-set-loop.4.11.1.-end\n12772   -define-Geo-geometry_builder\n12773   -V.10-B.10-TDO.3-fuse.1\n12774   -fname_GEO.10.3\n12775   -test-Test_lines\n12776   -orderly\n12777   -end
12778
12779 geo_10.3_orderly_mem_debug:
12780   $(ORBITER_PATH)orbiter.out-v.2\n12781   -memory_debug.2\n12782   -define-Test_lines-set-loop.4.11.1.-end\n12783   -define-Geo-geometry_builder\n12784   -V.10-B.10-TDO.3-fuse.1\n12785   -fname_GEO.10.3\n12786   -test-Test_lines\n12787   -orderly\n
geo_10_3.tree:
  $(ORBITER_PATH)orbiter.out -v.20 \\
  define Test_lines -set -loop 0.11.1 -end \\
  define GEO -geometry_builder \\
  define -v.20 -B.10 -TDD.3 -fuse.1 \\
  define -v.20 -fname GEO-10.3 \\
  define -v.20 -search_tree \\
  define -v.20 -test Test_lines \\
  define -v.20 -end \\
  $(ORBITER_PATH)orbiter.out -v.20 \\
  draw_options -embedded -radius 20 \\
  paperheight 220 \\
  paperwidth 330 \\
  xin 10000 -yin 10000 \\
  xout 1000000 -yout 500000 \\
  scale 2 -line_width 0.3 \\
  nodes_empty \\
  end \\
  tree_draw \\
  file 10_3_tree.txt \\
  end \\
  pdflatex 10_3_tree_draw.tex \\
  open 10_3_tree_draw.pdf \\
  geo_10_3.tree_path:
  $(ORBITER_PATH)orbiter.out -v.20 \\
  define Test_lines -set -loop 0.11.1 -end \\
  define GEO -geometry_builder \\
  define -v.20 -B.10 -TDD.3 -fuse.1 \\
  define -v.20 -fname GEO-10.3 \\
  define -v.20 -search_tree \\
  define -v.20 -test Test_lines \\
  define -v.20 -end \\
  $(ORBITER_PATH)orbiter.out -v.20 \\
  draw_options -embedded -radius 20 \\
  paperheight 220 \\
  paperwidth 330 \\
  xin 10000 -yin 10000 \\
  xout 1000000 -yout 500000 \\
  scale 2 -line_width 0.3 \\
  end
12878 \> \> -end
12879 \> pdflatex-Desargues_path_canAnc_incma.tex
12880 \> open-Desargues_path_canAnc_incma.pdf
12881
12882
12883
12884 geo_11_3:
12885 \> $(ORBITER_PATH)orbiter.out-v.2\n12886 \> \> -define-Test_lines-set-loop-4-12-1-end\n12887 \> \> -define-Geo-geometry_builder\n12888 \> \> \> -V-11-B-11-TDO-3\n12889 \> \> \> -fuse-1-fname_GEO-11_3\n12890 \> \> \> -test-Test_lines\n12891 \> \> -end
12892
12893 #31-geos
12894 #8/26/2021:0-sec-on-Mac
12895
12896 geo_12_3:
12897 \> $(ORBITER_PATH)orbiter.out-v.2\n12898 \> \> -define-Test_lines-set-loop-4-13-1-end\n12899 \> \> -define-Geo-geometry_builder\n12900 \> \> \> -V-12-B-12-TDO-3\n12901 \> \> \> -fuse-1-fname_GEO-12_3\n12902 \> \> \> -test-Test_lines\n12903 \> \> -end
12904
12905 #229-geos
12906 #User-time:0.45-of-a-second,dt=45tps=.100
12907 #nb_calls_to_densenauty=24586
12908
12909
12910 geo_12_3_orderly:
12911 \> $(ORBITER_PATH)orbiter.out-v.2\n12912 \> \> -define-Test_lines-set-loop-4-13-1-end\n12913 \> \> -define-Geo-geometry_builder\n12914 \> \> \> -V-12-B-12-TDO-3\n12915 \> \> \> -fuse-1-fname_GEO-12_3\n12916 \> \> \> -test-Test_lines\n12917 \> \> \> -f_orderly\n12918 \> \> -end
12919
12920
12921
12922 geo_13_3:
12923 \> $(ORBITER_PATH)orbiter.out-v.2\n12924 \> \> -define-Test_lines-set-loop-4-14-1-end\n12925

732
define Geo-geometry_builder:
  V-13-B-13-TDO-3:
  fuse-1-fname_GEO-13-3:
  test-Test_lines:
  end

geo_13_3_orderly:
  $(ORBITER_PATH)orbiter.out-v.2:
  define Test_lines-set-loop-4-14-1-end:
  define Geo-geometry_builder:
  V-13-B-13-TDO-3:
  fuse-1-fname_GEO-13-3:
  test-Test_lines:
  f_orderly:
  end

geo_14_3:
  $(ORBITER_PATH)orbiter.out-v.2:
  define Test_lines-set-loop-4-15-1-end:
  define Geo-geometry_builder:
  V-14-B-14-TDO-3:
  fuse-1-fname_GEO-14-3:
  test-Test_lines:
  end

 geo_14_3_orderly:
  $(ORBITER_PATH)orbiter.out-v.2:
  define Test_lines-set-loop-4-15-1-end:
  define Geo-geometry_builder:
  V-14-B-14-TDO-3:
  fuse-1-fname_GEO-14-3:
  test-Test_lines:
  f_orderly:
  end
12971 #User.time::0:50
12972
12973
12974 15_3.inc:
12975 > $(ORBITER_PATH)orbiter.out-v.2
12976 > > -define:Test_lines-set:-loop:4:16:1:-end:
12977 > > -define:Geo:-geometry_builder:
12978 > > > -V.15-B.15-TDO:3:
12979 > > > -fuse:1:-fname:GEO:15_3:
12980 > > > -test:Test_lines:
12981 > > > -end
12982
12983 #245342.geos,8064,720,192^2,128,72,48^6,32,30^2,24^2,20^2,18,16^10,
12984 15^2,12^11,10^3,8^34,6^59,5^5,4^180,3^69,2^3709,1^241240
12985 #8-min:11-sec-on-Mac-
12986 #running-out-of-memory
12987
12988 geo_15_3_g4:
12989 > $(ORBITER_PATH)orbiter.out-v.2
12990 > > -define:Test_lines-set:-loop:4:16:1:-end:
12991 > > -define:Geo:-geometry_builder:
12992 > > > -V.15-B.15-TDO:3:
12993 > > > -fuse:1:-fname:GEO:15_3_g4:
12994 > > > -girth:4:
12995 > > > -search_tree:
12996 > > > -test:Test_lines:
12997 > > > -end
12998 > > $(ORBITER_PATH)orbiter.out-v.2
12999 > > -draw_options:-embedded:-radius:50:
13000 > > > -nodes_empty:
13001 > > > -scale:0.5:-line_width:0.3:-end:
13002 > > -tree_draw:-file:15_3_g4_tree.txt:-end
13003 > pdflatex:15_3_g4_tree_draw.tex
13004 > open:15_3_g4_tree_draw.pdf
13005
13006 #The-unique-Cremona-Richmond-configuration-with-group-of-order-720
13007 #User.time::0-of-a-second,:dt=0-tps=:.100
13008 #nb_calls_to_densenauty=23
13009
13010 #-sideways.
13011
13012
13013
13014
13015 geo_17_3_g4_orderly:
13016 > $(ORBITER_PATH)orbiter.out-v.2
13017  ▶  ▶  -memory_debug.2\n13018  ▶  ▶  -define:Test_lines-set-loop-4.18.1-end\n13019  ▶  ▶  -define:Geo-geometry_builder\n13020  ▶  ▶  ▶  -V.17--B.17--TDO.3\n13021  ▶  ▶  ▶  ▶  -fuse.1--fname GEO.17.3 g4\n13022  ▶  ▶  ▶  ▶  -girth.4\n13023  ▶  ▶  ▶  ▶  -test:Test_lines\n13024  ▶  ▶  ▶  ▶  -orderly\n13025  ▶  ▶  ▶  ▶  -end
13026
13027  #.1: sol
13028
13029  geo_18.3 g4:
13030  ▶  $(ORBITER_PATH)orbiter.out-v.2\n13031  ▶  ▶  -define:Test_lines-set-loop-4.19.1-end\n13032  ▶  ▶  -define:Geo-geometry_builder\n13033  ▶  ▶  ▶  -V.18--B.18--TDO.3\n13034  ▶  ▶  ▶  ▶  -fuse.1--fname GEO.18.3 g4\n13035  ▶  ▶  ▶  ▶  -girth.4\n13036  ▶  ▶  ▶  ▶  -search_tree\n13037  ▶  ▶  ▶  ▶  -test:Test_lines\n13038  ▶  ▶  ▶  ▶  -end
13039
13040  #.4: sol, 1: sec
13041
13042
13043  geo_19.3 g4:
13044  ▶  $(ORBITER_PATH)orbiter.out-v.2\n13045  ▶  ▶  -define:Test_lines-set-loop-4.20.1-end\n13046  ▶  ▶  -define:Geo-geometry_builder\n13047  ▶  ▶  ▶  -V.19--B.19--TDO.3\n13048  ▶  ▶  ▶  ▶  -fuse.1--fname GEO.19.3 g4\n13049  ▶  ▶  ▶  ▶  -girth.4\n13050  ▶  ▶  ▶  ▶  -test:Test_lines\n13051  ▶  ▶  ▶  ▶  -end
13052
13053  #.14: sol, 10: sec on Mac
13054
13055  geo_20.3 g4:
13056  ▶  $(ORBITER_PATH)orbiter.out-v.2\n13057  ▶  ▶  -define:Test_lines-set-loop-4.21.1-end\n13058  ▶  ▶  -define:Geo-geometry_builder\n13059  ▶  ▶  ▶  -V.20--B.20--TDO.3\n13060  ▶  ▶  ▶  ▶  -fuse.1--fname GEO.20.3 g4\n13061  ▶  ▶  ▶  ▶  -girth.4\n13062  ▶  ▶  ▶  ▶  -test:Test_lines\n13063  ▶  ▶  ▶  ▶  -end
13064
13065 # 162: sol, User.time: 2:5 on Mac
13066
13067 geo_21.3_g4:
13068 $ (ORBITER_PATH) orbiter.out -v.2 \
13069 $ define-Test_lines -set-loop-4:22:1-end \
13070 $ define-Geo-geometry_builder \
13071 $ V.21-B.21-TDO.3 \$
13072 $ fuse-1-fname_GEO.21.3_g4 \
13073 $ girth.4 \$
13074 $ test-Test_lines \$
13075 $ end
13076
13077
13078 geo_15.4:
13079 $ (ORBITER_PATH) orbiter.out -v.2 \
13080 $ define-Test_lines -set-loop-4:16:1-end \
13081 $ define-Geo-geometry_builder \
13082 $ V.15-B.15-TDO.4 \$
13083 $ fuse-1-fname_GEO.15.4 \$
13084 $ search_tree \
13085 $ test-Test_lines \$
13086 $ end
13087 $ (ORBITER_PATH) orbiter.out -v.2 \
13088 $ draw_options-embedded-radius.50 \$
13089 $ nodes_empty \$
13090 $ scale.0.5-line_width.0.3-end \$
13091 $ tree_draw-file.15.4_tree.txt-end \
13092 $ pdflatex.15.4_tree draw.tex \
13093 $ open.15.4_tree draw.pdf \
13094
13095 # 4 objects
13096 # ago=360, 30, 24, 15
13097 # User.time: 0.15 of a second, dt=15 tps = 100
13098 # nb_calls_to_densenauty = 6767
13099
13100
13101
13102 geo_16.4_g4:
13103 $ (ORBITER_PATH) orbiter.out -v.2 \
13104 $ define-Test_lines -set-loop-4:17:1-end \
13105 $ define-Geo-geometry_builder \
13106 $ V.16-B.16-TDO.4 \$
13107 $ fuse-1-fname_GEO.16.4_g4 \$
13108 $ girth.4 \$
13109 $ test-Test_lines \$
13110 $ end

736
40_4_g4.inc:

```
$(ORBITER_PATH)orbiter.out-v.2\n  -define Test_lines -set-loop 0.41.1.-end\n  -define Geo-geometry_builder\n  -V-40-440-4TD0.4\n  -girth4-\n  -search_tree\n  -define Geo-geometry\n  -V 40 4 4 36\n  -girth 6\n  -end
```

draw_options-embedded-radius 50-\n-xin 10000-yin 10000-\n-xout 1000000-yout 1000000-\n-nodes_empty-\n-scale 0.5-line_width 0.3-end-\n-tree_draw-file 40_4_g4_tree.txt-end-\npdflatex 40_4_g4_tree_draw.tex\nopen 40_4_g4_tree_draw.pdf

#2_geos,-ago=51840^2
#User:time: 0.18 of a second, dt=18tps= 100
#nb_calls_to_densenaity=1065

geo_63_3_g6:
```
$(ORBITER_PATH)orbiter.out-v.2\n  -define Test_lines -set-loop 4.64.1.-end\n  -define Geo-geometry_builder\n  -V 63-63-63-TDO.3\n  -girth64-\n  -test Test_lines\n  -end
```

geo_LSQ6:
```
$(ORBITER_PATH)orbiter.out-v.2\n  -define Test_lines -set-loop 1.19.1.-end\n  -define Geo-geometry_builder\n  -V 6,6,6,-B,1,1,1,36-TDO-
design PG_2_3:

design PG_2_4:

design PG_2_3_table_create:
13205 \>$\text{\(\text{ORBITER\_PATH/}\text{orbiter.out-v.2/}\)}$
13206 \>$\text{\(-define\_D\_design-q.3\_family-PG2\_q\_end\)}$
13207 \>$\text{\(-define\_Sym13\_permutation\_group-symmetric\_group13\_end\)}$
13208 \>$\text{\(-define\_T\_design\_table\_D\_PG2\_13\_Sym13\_end\)}$
13209
13210 \>$\text{\#written\_file-PG2\_13\_design\_table.csv}$
13211 \>$\text{\#1108800\_designs}$
13212 \>$\text{\#User\_time::7:30}$
13213
13214 \>$\text{design\_PG2\_3\_group\_5:}$
13215 \>$\text{\$\text{\(\text{ORBITER\_PATH/}\text{orbiter.out-v.2/}\)}\}}$
13216 \>$\text{\(-define\_D\_design-q.3\_family-PG2\_q\_end\)}$
13217 \>$\text{\(-define\_T\_design\_table\_D\_PG2\_13\_Sym13\_end\)}$
13218 \>$\text{\(-define\_LSW\_large\_set\_with\_symmetry\_assumption\_T\)}$
13219 \>$\text{\(-H\_5\_\$(\text{\(\text{GENERATORS\_H5/}\))}\)}$
13220 \>$\text{\(-N\_1200\_\$(\text{\(\text{GENERATORS\_N5/}\))}\)}$
13221 \>$\text{\(-prefix\_H5\)}$
13222 \>$\text{\(-selected\_orbit\_length\_5\)}$
13223 \>$\text{\(-end\)}$
13224 \>$\text{\(-with\_LSW\_do\)}$
13225 \>$\text{\(-large\_set\_with\_symmetry\_assumption\_activity\)}$
13226 \>$\text{\(-normalizer\_on\_orbits\_of\_a\_given\_length\_5\)}$
13227 \>$\text{\(-end\)}$
13228
13229 \>$\text{\#H5\_N\_orbit\_reps.csv}$
13230 \>$\text{\#678\_orbits}$
13231 \>$\text{\#User\_time::2:39}$
13232
13233 \>$\text{design\_PG2\_3\_group\_5\_sol\_0:}$
13234 \>$\text{\$\text{\(\text{ORBITER\_PATH/}\text{orbiter.out-v.2/}\)}\}}$
13235 \>$\text{\(-define\_D\_design-q.3\_family-PG2\_q\_end\)}$
13236 \>$\text{\(-define\_T\_design\_table\_D\_PG2\_13\_Sym13\_end\)}$
13237 \>$\text{\(-define\_LSW\_large\_set\_with\_symmetry\_assumption\_T\)}$
13238 \>$\text{\(-H\_5\_\$(\text{\(\text{GENERATORS\_H5/}\))}\)}$
13239 \>$\text{\(-N\_1200\_\$(\text{\(\text{GENERATORS\_N5/}\))}\)}$
13240 \>$\text{\(-prefix\_H5\)}$
13241 \>$\text{\(-selected\_orbit\_length\_5\)}$
13242 \>$\text{\(-end\)}$
13243 \>$\text{\(-with\_LSW\_do\)}$
13244 \>$\text{\(-large\_set\_with\_symmetry\_assumption\_activity\)}$
13245 \>$\text{\(-read\_solution\_file\_5\_case\_0\_sol.txt\)}$
13246 \>$\text{\(-end\)}$
13247 \>$\text{\(-end\)}$
13248
13249
13250
13251 \>$\text{KM\_cyclic\_7:}$
13252 \> $(\text{ORBITER\_PATH})\text{orbiter.out- \text{\$-}v.3}$\\ 
13253 \> \> -define-gens-vector-dense:"1,2,3,4,5,6,0"-end\\ 
13254 \> \> -define-G-permutation\_group-symmetric\_group-7\\ 
13255 \> \> \> -subgroup\_by\_generators:"C7":7\_gens\\ 
13256 \> \> -end\\ 
13257 \> \> -with-G-do\\ 
13258 \> \> -group\_theoretic\_activity\\ 
13259 \> \> \> -poset\_classification\_control\\ 
13260 \> \> \> \> -problem\_label:KM-C7-W-depth-3\\ 
13261 \> \> \> \> -Kramer_Mesner\_matrix-2\_3\\ 
13262 \> \> \> \> -draw\_poset\\ 
13263 \> \> \> \> -draw\_options-embedded-sideways-radius-50\\ 
13264 \> \> \> \> \> -scale-0.5-line\_width-0.3-end\ 
13265 \> \> \> -end\ 
13266 \> \> \> -orbits\_on\_subsets-3\
13267 \> \> -end
13268 \> $(\text{ORBITER\_PATH})\text{orbiter.out- \text{\$-}v.4}$\ 
13269 \> \> -define-A-vector-file:C7\_KM-2.3.csv-end\ 
13270 \> \> -define-D-diophant:\ 
13271 \> \> -label:"C7\_KM\_2.3\_system"\ 
13272 \> \> -coefficient\_matrix:A\ 
13273 \> \> -RHS\_constant:"1,1,1"\ 
13274 \> \> -x\_min\_global:0:-x\_max\_global:1\ 
13275 \> \> -end\ 
13276 \> \> -with-D-do\ 
13277 \> \> \> -diophant\_activity-solve.mckay\ 
13278 \> \> -end\ 
13279 
13280 
13281 
13282 \# to create simple 7-designs on 33 points with block size 8 and lambda = 10 invariant under PGGL(2,32):
13283 
13284 KM\_PGGL\_2\_32:
13285 \> $(\text{ORBITER\_PATH})\text{orbiter.out- \text{\$-}v.3}$\ 
13286 \> \> -define-G-linear\_group-PGGL\_2\_32-end\ 
13287 \> \> -with-G-do\ 
13288 \> \> -group\_theoretic\_activity\ 
13289 \> \> \> -poset\_classification\_control\ 
13290 \> \> \> \> -problem\_label:KM\_PGGL\_2\_32-W-depth-8\ 
13291 \> \> \> \> -Kramer_Mesner\_matrix-7\_8\ 
13292 \> \> \> \> -draw\_poset\ 
13293 \> \> \> \> -draw\_options-embedded-sideways-radius-50\ 
13294 \> \> \> \> \> -scale-0.5-line\_width-0.3-end\ 
13295 \> \> \> \> -end\ 
13296 \> \> \> \> -orbits\_on\_subsets-8\ 
13297 \> \> -end

740
$(\text{ORBITER\_PATH})$ orbiter.out -v.2 -draw_matrix -v.4

-define A -vector -file KM_PSL_3_5_KM_8_10.csv -end

-define D -diophant

-label "KM_PSL_3_5_KM_8_10\_system"

-coefficient\_matrix A

-RHS\_constant "93,93,1"

-x\_min\_global 0 -x\_max\_global 0.1

-end

-with D do

-diophant\_activity -solve mckay

-end

-end

-end

-orbits on subsets 10

-end

$(\text{ORBITER\_PATH})$ orbiter.out -v.3

-define G -linear\_group -PSL 3.5 -end

-with G do

-group\_theoretic\_activity

-poset\_classification\_control

-problem\_label KM_PSL 3.5 -W -depth 10

-Kramer\_Mesner\_matrix 8:10

-draw\_poset

-draw\_options -embedded -sideways

-radius 50 -scale 0.5 -line\_width 0.3

-end

-end

-end

-define A -vector -file KM_PSL_3_5_KM_8_10.csv -end

-define D -diophant

-label "KM_PSL_3_5_KM_8_10\_system"

-coefficient\_matrix A

-RHS\_constant "93,93,1"

-x\_min\_global 0 -x\_max\_global 0.1

-end

-end
design large set rank k subsets a:  
$(\text{ORBITER PATH})\text{orbiter.out}$ -v -2 -rank k subset 13:4 $(\text{PLANE}_1)$

design large set rank k subsets b:
$(\text{ORBITER PATH})\text{orbiter.out}$ -v -2 -rank k subset 13:4 $(\text{PLANE}_2)$

design large set rank k subsets c:
$(\text{ORBITER PATH})\text{orbiter.out}$ -v -2 -rank k subset 13:4 $(\text{PLANE}_3)$

design large set rank k subsets d:
$(\text{ORBITER PATH})\text{orbiter.out}$ -v -2 -rank k subset 13:4 $(\text{PLANE}_4)$
SECTION DESIGN THEORY LARGE_SETS:

LS_AG_2_3_design_table_create:

#creates AG_2_3_design_table.csv

#and AG_2_3.makefile

#is the first design in AG_2_3_design_table.csv

#poset.orbit_node::init_root_node storing strong generators for a group of order 362880

#stabilizer.order 432

#840-designs

#User.time: 0.13 of a second, dt=13 tps = 100
AG_2.3_on_designs:

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
> -define gens -vector file AG_2.3_gens.csv -end
```

```bash
> -define G -permutation_group
```

```bash
> -bsgs AG_2.3 -AG_2.3 -840 362880 -0,1,2,3,4,5,6,7 -gens -end
```

```bash
> -with G -do
```

```bash
> -group_theoretic_activity
```

```bash
> -orbits on points
```

```bash
> -stabilizer of orbit rep 0
```

```bash
> -end
```

```bash
#Written file AG_2.3_stab_orb_0.makefile of size 239
```

```bash
```

```bash
#the stabilizer of the first design:
```

```bash
```

```bash
AG_2.3_stab_orb_0:
```

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
> -define gens -vector file AG_2.3_stab_orb_0_gens.csv -end
```

```bash
> -define G -permutation_group
```

```bash
> -bsgs AG_2.3_stab_orb_0 AG_2.3_stab_orb_0 -840 432 0,1,2,3,4,5,6,7,8 -gens
```

```bash
> -end
```

```bash
> -define Gr -modified group -from G
```

```bash
> -restricted action $(LARGE_SET AG_2.3 NEIGHBOR_SET)
```

```bash
> -end
```

```bash
> -with Gr -do
```

```bash
> -group_theoretic_activity
```

```bash
> -export orbiter
```

```bash
> -end
```

```bash
```

```bash
AG_2.3_stab_orb_0_Perm840_res192:
```

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
> -define gens -vector file Perm840_res192_gens.csv -end
```

```bash
> -define G -permutation_group
```

```bash
> -bsgs Perm840_res192 Perm840 {\rm res192} -192 432 0,1,2,3,4,5,6,7,8 -gens
```

```bash
> -end
```

```bash
> -with G -do
```

```bash
> -group_theoretic_activity
```

```bash
> -report
```

```bash
> -end
```

```bash
pdflatex Perm840_res192_report.tex
```

```bash
open Perm840_res192_report.pdf
```
LS_AG_2_3.disjoint_sets.graph_and_cliques:

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
define Gamma -graph
```

```bash
disjoint_sets_graph
```

```bash
AG_2_3_design_table.csv
```

```bash
end
```

```bash
with Gamma -do
```

```bash
graph_theoretic_activity
```

```bash
save
```

```bash
end
```

```bash
with Gamma -do
```

```bash
graph_theoretic_activity
```

```bash
find_cliques -target_size:7 -end
```

```bash
end
```

```bash
print_symbols
```

#AG_2_3_design_table.disjoint_sets.colored_graph

#User time: 0.66 of a second, dt = 66 tps = .100

#AG_2_3_design_table.disjoint_sets_sol.txt

#AG_2_3_design_table.disjoint_sets_sol.csv

15360 solutions

LS_AG_2_3.disjoint_sets.split:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
define Gamma -graph -load
```

```bash
AG_2_3_design_table.disjoint_sets.colored_graph
```

```bash
end
```

```bash
with Gamma -do
```

```bash
graph_theoretic_activity
```

```bash
split_by_clique "0" "0"
```

```bash
end
```

#AG_2_3_design_table.disjoint_sets_0.graph

#AG_2_3_design_table.disjoint_sets_0.subset.txt

15317

LS_AG_2_3.export.solutions:

```bash
$(ORBITER_PATH)orbiter.out -v 20
```

```bash
define D -design -list_of_blocks: 9 3
```

```bash
$(AG_2_3_BLOCKS) -end
```

#AG_2_3_design_table.disjoint_sets.colored.graph

#AG_2_3_design_table.disjoint_sets_sol.txt

#AG_2_3_design_table.disjoint_sets_sol.csv

#15360 solutions

#AG_2_3_design_table.disjoint_sets_0.graph

#AG_2_3_design_table.disjoint_sets_0.subset.txt

export solutions:
section 1.1.7: design theory - delandtsheer-doyen

section design theory delandtsheer_doyen:

dd_pp4:

-define $(ORBITER_PATH) orbiter.out -v 6 \n-define T -define_table_D "AG_2_3" Sym9 \n-with D -do \n-extract_solutions_by_index "AG_2_3" Sym9 \n AG_2_3_design_table_disjoint_sets_sol.csv \nsolutions.csv \ns -end

dd_pp4_system:

-define $(ORBITER_PATH) orbiter.out -v 4 \n-diophant -label PP4 \n-problem_of_steiner_type_10_PPP4_pair_covering.csv \n-has_sum 1 \n-end \n-with D -do \n-diophant_activity -solve mckay \n-end

dd_cc:

-define $(ORBITER_PATH) orbiter.out -v 6 \n-define D -diophant -label PP4 \n-problem_of_steiner_type_10_PPP4_pair_covering.csv \n-has_sum 1 \n-end \n-with D -do \n-diophant_activity -solve mckay \n-end

...
13571 $\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_COLBOURN\_COLBOURN\_7\_13\_MASK1}\}\$
13572 -end
13573
13574 \#target\_level: 6
13575 \#k2: 15
13576 \#number\_of\_k\_orbits\_at\_target\_level: 1774964
13577
13578 \#creates DD\_CC\_7\_13\_pair\_covering.csv
13579
13580 DD\_CC\_system:
13581 $(ORBITER\_PATH)\text{orbiter.out}\text{-v\_4}\$
13582 -define D\_diophant\_label DD\_CC\_7\_13$
13583 -problem\_of\_Steiner\_type 45 DD\_CC\_7\_13\_pair\_covering.csv$
13584 -has_sum 3$
13585 -end$
13586 -with D\_do$
13587 -diophant\_activity\_solve\_mckay$
13588 -end
13589
13590
13591
13592
13593 \#no\_solution
13594
13595
13596
13597 \#18603 = 27*53*13
13598
13599 DD\_M1\_G1:
13600 $(ORBITER\_PATH)\text{orbiter.out}\text{-v\_4}\$
13601 -Delandtsheer\_Doyen$
13602 -\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_27\_53}\}\$
13603 -\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_27\_53\_GROUP1}\}\$
13604 -\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_27\_53\_MASK1}\}\$
13605 -end$
13606
13607 DD\_M1\_G1\_S:
13608 $(ORBITER\_PATH)\text{orbiter.out}\text{-v\_4}\$
13609 -Delandtsheer\_Doyen$
13610 -\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_27\_53}\}\$
13611 -\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_27\_53\_GROUP1}\}\$
13612 -\{\text{DELANDTSHEER\_DOYEN\_PROBLEM\_27\_53\_MASK1}\}\$
13613 -singletons$
13614 -end$
13615
13616
13617 DD\_PG\_2\_4\_M1\_G1:
Section 11.8: Tactical Decompositions

maxarc16_4 start:
maxarc16_4 convert stack tdo:
maxarc16_4 refine:
maxarc16_4 print:

Chapter 12: Finite Geometry

Section Tactical Decompositions:

maxarc16_4 start:
maxarc16_4 convert stack tdo:
maxarc16_4 refine:
maxarc16_4 print:
Section 12.1: Spreads

SECTION_SPREADS:

desarguesian_spread_in_PG_3_2:
$\$(\text{ORBITER_PATH})\text{orbiter.out} -v.3\$
\>$\text{-define}\ FQ\text{-finite_field}\text{-q.4}\text{-end}\$
\>$\text{-define}\ Fq\text{-finite_field}\text{-q.2}\text{-end}\$
\>$\text{-with}\ FQ\text{-and}\ Fq\text{-do}\text{-finite_field_activity}\$
\>$\text{-cheat_sheet_desarguesian_spread.2}\text{-end}$
\>pdflatex\text{Desarguesian_Spread.3.2.tex}$
\>open\text{Desarguesian_Spread.3.2.pdf}$

desarguesian_spread_in_PG_3_4:
$\$(\text{ORBITER_PATH})\text{orbiter.out} -v.3\$
\>$\text{-define}\ FQ\text{-finite_field}\text{-q.16}\text{-end}\$
\>$\text{-define}\ Fq\text{-finite_field}\text{-q.4}\text{-end}\$
\>$\text{-with}\ FQ\text{-and}\ Fq\text{-do}\text{-finite_field_activity}\$
\>$\text{-cheat_sheet_desarguesian_spread.2}\text{-end}$
\>pdflatex\text{Desarguesian_Spread.3.4.tex}$
\>open\text{Desarguesian_Spread.3.4.pdf}$

desarguesian_spread_in_PG_3_5:
$\$(\text{ORBITER_PATH})\text{orbiter.out} -v.3\$
\>$\text{-define}\ FQ\text{-finite_field}\text{-q.25}\text{-end}\$
\>$\text{-define}\ Fq\text{-finite_field}\text{-q.5}\text{-end}\$
\>$\text{-with}\ FQ\text{-and}\ Fq\text{-do}\text{-finite_field_activity}\$
\>$\text{-cheat_sheet_desarguesian_spread.2}\text{-end}$
\>pdflatex\text{Desarguesian_Spread.3.5.tex}$
\>open\text{Desarguesian_Spread.3.5.pdf}$

749
spreads4:
- mdir SPREADS_4
- rm live_points.txt
- $(ORBITER_PATH)orbiter.out -v 10
- define F -finite_field -q 2 -end
- define P -projective_space 3 F -end
- with P -do
- projective_space_activity
- spread classify 2
- problem_label spreads 2 2 -depth 5
- draw poset
- end

spreads16_4:
- $(ORBITER_PATH)orbiter.out -v 6
- orbiter_path $(ORBITER_PATH)
- define F -finite_field -q 4 -end
- define P -projective_space 3 F -end
- with P -do
- projective_space_activity
- spread classify 2
- problem_label spreads 4 2 -depth 17
- draw poset
- draw_options -radius 20
- nodes_empty -line_width 0.2 -embedded
- end
- report
- end

# pdflatex spreads 4 2 poset detailed lvl 17.tex
# open spreads 4 2 poset detailed lvl 17.pdf

# 17
# 17-0.25-50.75-90.107-122-140.144-157-179.204-213-238-268.334-345-1200-agaaaaaad
# aaaaaaaabaaaaaaaaaaaaaaaaaaaaaagaaaaaaaaaaaaaaaaaaafaaaaaabaaaabaaaaaahaaaaadamejgpaaacajljdmabfncjmbneaa
# 17-0.25-50.75-90.107-140-157-179-204-213-238-265-282-299-316-356-81600-agaaaaaar
# haaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaagaaaaaafaaaaaaaafaaaaaabaaaabaaaaaaadamcaiaaacaiahaaacaiajlpahabahelaalabecljdaaabmpbnaeaa
# amanajmpnna
# aaaaaaaaaaaaabaaaaaaaacaaaaaaaaaaaaaaaaaaagaaaaaaaaaafaaaaaaaabaaaabaaaaaaadamcaiaaaacaiahaaacaiajlpahabahelaalabecljdaaabmpbnaeaa
# amanajmpnna
# -1.3-1126-in 0 of a second, dt=0-tps=.100
# (81600,1200,576)-average is 27792+ 0 / 3

750
# Section 12.2: Translation planes

SECTION_TRANSLATION_PLANES:

# ToDo:

TP_9_0:

```bash
$ (ORBITER_PATH)orbiter.out -v.3:
```

```
> define F finite_field -q.3 -end:
```

```
> define PGL4 linear_group -PGL.4 F -end:
```

```
> define PGL5 linear_group -PGL.5 F -end:
```

```
> with PGL4 and PGL5 -do:
```

```
> group_theoretic_activity:
```

```
> > Andre_Bruck_Bose_construction 0 "TP9-0":
```

```bash
> end
```

```
> $(ORBITER_PATH)orbiter.out -v.2 -draw_matrix:
```

```
> > input_csv_file TP9-0 incma.csv:
```

```
> > box_width 6 -bit_depth 8 -partition 6 91 91 -end
```

```
> open TP9-0 incma_draw.bmp
```

TP_9_1:

```
$ (ORBITER_PATH)orbiter.out -v.3:
```

```
> define F finite_field -q.3 -end:
```

```
> define PGL4 linear_group -PGL.4 F -end:
```

```
> define PGL5 linear_group -PGL.5 F -end:
```

```
> with PGL4 and PGL5 -do:
```

```
> group_theoretic_activity:
```

```
> > Andre_Bruck_Bose_construction 0 "TP9-1":
```

```bash
> end
```

```
> $(ORBITER_PATH)orbiter.out -v.2 -draw_matrix:
```

```
> > input_csv_file TP9-1 incma.csv:
```

```
> > box_width 6 -bit_depth 8 -partition 6 91 91 -end
```

```
> open TP9-1 incma_draw.bmp
```

```
> pdfflatex TP9-1 report.tex
```

```
> open TP9-1 report.pdf
```

```bash
#ToDo:

TP_16_4:

- $(ORBITER_PATH)orbiter.out --v.3
- define-F--finite_field--q.4--end
- define-PGGL4--linear_group--PGGL.4:F--end
- define-PGGL5--linear_group--PGGL.5:F--end
- with-PGGL4--and-PGGL5--do
- group_theoretic_activity
- Andre_Brack_Bose_construction:0:"TP16-4-HALL"
- end

- $(ORBITER_PATH)orbiter.out --v.2--draw_matrix
- input_csv_file:TP16-4-HALL_incma.csv
- box_width:6--bit_depth:8
- end

- open:TP16-4-HALL_incma_draw.bmp
- pdflatex:TP16-4-HALL_report.tex
- open:TP16-4-HALL_report.pdf

TP_16_2:

- $(ORBITER_PATH)orbiter.out --v.3
- define-F--finite_field--q.2--end
- define-PGGL8--linear_group--PGGL.8:F--end
- define-PGGL9--linear_group--PGGL.9:F--end
- with-PGGL8--and-PGGL9--do
- group_theoretic_activity
- Andre_Brack_Bose_construction:0:"TP16_0_1008"
- end

- "1008","1008","1728","216","360","288","3600","244800"
SECTION PACKINGS:

spread_table_PG_3_4:

- mkdir SPREAD_TABLES_4

$(ORBITER\ PATH)\ orbiter.\ out\ -v\ -6\ \\
-define F\ -finite_field\ -q\ -4\ -end\ \\
-define P\ -projective_space\ -3\ F\ -end\ \\
-define T\ -spread_table\ -P\ -2\ \"0,1,2\"\ \\ "SPREAD\ TABLES\ 4/\ \\

# 5096448 spreads

# 1020 self-dual spreads

# User time: 56:38 on Mac

spread_table_PG_3_5_regular:

- mkdir SPREAD_TABLES_5_REG

$(ORBITER\ PATH)\ orbiter.\ out\ -v\ -6\ \\
-define F\ -finite_field\ -q\ -5\ -end\ \\
-define P\ -projective_space\ -3\ F\ -end\ \\
-define T\ -spread_table\ -P\ -2\ \"12\"\ "SPREAD\ TABLES\ 5\ REG/\ \\

# 12/9/2020: 34 sec on Mac

# 12 is the index of the regular spread in the classification of spreads

# 155000 spreads

PG_3_5_desarguesian_spread:

$(ORBITER\ PATH)\ orbiter.\ out\ -v\ -3\ \\
-define FQ\ -finite_field\ -q\ -25\ -end\ \\
-define Fq\ -finite_field\ -q\ -5\ -end\ \\
-with FQ\ and Fq\ do\ \\
-finite_field\ activity\ \\
-define cheat_sheet.desarguesian_spread\ -2\ \\
-end

pdflatex Desarguesian_Spread_3_5.tex

open Desarguesian_Spread_3_5.pdf

# Spread elements by rank: (0, 805, 36, 108, 72, 144, 581, 509, 686, 415, 639, 758, 285, 722, 332, 343, 202, 592, 473, 238, 675, 379, 166, 545, 249, 451)
13892
13893
13894 PG_3.5_element_of_order_31:
13895 $$(\text{ORBITER\_PATH})\text{orbiter.out-v.6.-define-G}$$
13896 $$\text{-linear_group-GL.3-5-end}$$
13897 $$\text{-with-G-do}$$
13898 $$\text{-group_theoretic_activity}$$
13899 $$\text{-raise_to_the_power}"0,1,0,0,1,0,3,0,4".31\$$
13900 $$\text{-end}$$
13901 pdflatex GL_3.5_power.tex
13902 open GL_3.5_power.pdf
13903
13904 PG_3.5_element_of_order_31_normalizer:
13905 $$(\text{ORBITER\_PATH})\text{orbiter.out-v.6.-define-G}$$
13906 $$\text{-linear_group-PGL.4-5-end}$$
13907 $$\text{-with-G-do}$$
13908 $$\text{-group_theoretic_activity}$$
13909 $$\text{-normalizer_of_cyclic_subgroup"31"}$$
13910 $$\text{"2,0,0,0,0,1,0,0,0,1,0,3,0,4"}$$
13911 $$\text{-end}$$
13912 mv normalizer_of_31_in_PGL_4.5.tex normalizer_of_31_AB_in_PGL_4.5.tex
13913 pdflatex normalizer_of_31_AB_in_PGL_4.5.tex
13914 open normalizer_of_31_AB_in_PGL_4.5.pdf
13915
13916
13917
13918 PG_3.5_element_of_order_31_GL_normalizer:
13919 $$(\text{ORBITER\_PATH})\text{orbiter.out-v.6.-define-G}$$
13920 $$\text{-linear_group-GL.4-5-end}$$
13921 $$\text{-with-G-do}$$
13922 $$\text{-group_theoretic_activity}$$
13923 $$\text{-normalizer_of_cyclic_subgroup"124"}$$
13924 $$\text{"2,0,0,0,0,1,0,0,0,0,1,0,3,0,4"}$$
13925 $$\text{-end}$$
13926 mv normalizer_of_31_in_PGL_4.5.tex normalizer_of_31_AB_in_PGL_4.5.tex
13927 pdflatex normalizer_of_124_in_GL_4.5.tex
13928 open normalizer_of_124_in_GL_4.5.pdf
13929
13930
13931
13932 PG_3.5_element_of_order_31_ME_normalizer:
13933 $$(\text{ORBITER\_PATH})\text{orbiter.out-v.6.-define-G}$$
13934 $$\text{-linear_group-PGL.4-5-end}$$
13935 $$\text{-with-G-do}$$
13936 $$\text{-group_theoretic_activity}$$
13937 $$\text{-normalizer_of_cyclic_subgroup"31"}$$
13938 $$\text{"1,0,0,0,0,3,4,3,0,3,3,4,0,3,2,3"}$$

754
normalizer has order $1488 = 4 \times 372 = 4 \times 4 \times 3 \times 31$

PG $3_5$ assume $31$ graph:

```
$(ORBITER_PATH) orbiter.out -v 5 \
define F -finite_field -q 5 -end \
define P -projective_space -3 F -end \
define T -spread_table -P 2 -12 -"SPREAD_TABLES.5_REG/" \
define P -packing_with_symmetry_assumption -T \
define H -H "31" "$(PGL_4_5_SUBGROUP_31_ME)" -end \
define N -N "31" "$(PGL_4_5_SUBGROUP_31_ME_NORMALIZER)" -end \
end \
define PW -packing_choose_fixed_points -PW 0 -end \
define L -packing_long_orbits -PW F \
end \
end \
print_symbols
```

```
pdflatex H31_reduced_spread_orbits_orbits_report.tex 
opn H31_reduced_spread_orbits_orbits_report.pdf
```

```
pdflatex H31_line_orbits_orbits_report.tex
open H31_line_orbits_orbits_report.pdf
```

```
pdflatex H31_line_orbits_orbits_report.tex
open H31_line_orbits_orbits_report.pdf
```

```
pdflatex H31_point_orbits_orbits_report.tex
open H31_point_orbits_orbits_report.pdf
```

```
pdflatex N31_line_orbits_orbits_report.tex
open N31_line_orbits_orbits_report.pdf
```

```
pdflatex H31_point_orbits_orbits_report.tex
open H31_point_orbits_orbits_report.pdf
```

```
pdflatex H31_spread_orbits_orbits_report.tex
```

```
open H31_spread_orbits_orbits_report.pdf
```

```
H31_line_orbits_orbits.bin
H31_line_orbits_orbits.report.tex
H31_line_orbits_orbits.report.pdf
H31_line_orbits_orbits.report.bin
H31_good_orbits
H31_spread_orbits_orbit_types_report.tex
H31_spread_orbits_orbit_types_report.pdf
H31_spread_types_reduced_orbit_types_report.tex
H31_spread_types_reduced_orbit_types_report.pdf
H31_reduced_spread_orbits_orbits.bin
H31_fpc0_lo.graph
PG_3.5_assume_31_fpc0_lo_cliques:

$ (ORBITER\_PATH)\text{orbiter.out}\ -v\ -2 $

-define G -graph -load H31_fpc0_lo.graph -end \\n-define G -do \\n  -graph_theoretic_activity \\n  -find_cliques -target_size 1 -end -end \\n-print_symbols

#H31_fpc0_lo_sol.txt
#H31_fpc0_lo_sol.csv

Todo: problem when computing the orbits of the normalizer:

PG_3.5_assume_31_read_again:

$ (ORBITER\_PATH)\text{orbiter.out}\ -v\ -5 $ \\n-define F -finite_field -q 5 -end \\n-define P -projective_space 3 F -end \\n-define T -spread_table P:2:"12":"SPREAD\_TABLES.5\_REG/".$ \\n-define PW -packing with symmetry assumption T:\ \n-define H.\"H31\" $(PGL.4.5\_SUBGROUP.31\_ME) -end \\n-define N.\"H31\" $(PGL.4.5\_SUBGROUP.31\_ME) -end \\n-end \\n-define PW -packing choose fixed points PW 0 -end \\n-define L -packing long orbits PW \\n-orbit_length 31 -clique_size 1 \\n-read_solutions \\n-end \\
writes H31.packings.csv

The two packings (ignore the first number):

#0,444,43313,154402,46682,108254,75363,27729,32139,5244,60442,142811,111115,94209
,120678,89533,13798,103994,129953,82168,136838,19253,23017,145985,134996,54705,36
267,55066,117542,96699,69154,74036

#1,616,42728,152655,48576,105431,79607,28634,32817,9799,62356,141176,110085,92557
,122136,86312,13975,101942,126869,81478,139352,18028,24325,147284,130370,52074,36
843,55602,118454,95973,69642,74036

576
PG_3_5_packing_0_dualize:
-define-F-finite_field-q-5-end-
-define-P-projective_space-3-F-end-
-define-T-spread_table-P-2:"12".SPREAD_TABLES_5_REG/
-with-T-do-
-spread_table_activity-
-dualize Packing $(PENTILLA_WILLIAMS_PRINCE_REG Packing_0):
-end.

PG_3_5_assume_3:
-rm-N3B_ME_fxp_cliques.csv
$(ORBITER_PATH)orbiter.out-v-5-
-define-F-finite_field-q-5-end-
-define-P-projective_space-3-F-end-
-define-T-spread_table-P-2:"12".SPREAD_TABLES_5_REG/
-Packing with symmetry assumption T:
-h "H3B_ME" $(PGL_4_5_SUBGROUP_3B_ME)-end-
-N "N3B_ME" $(PGL_4_5_SUBGROUP_3B_ME_NORMALIZER)-end-
-end-
-end-
-Packing choose fixed points PW-1-
-W problem label N3B_ME_fxp_cliques-
-preferred choice 0 0 2-end-
-print symbols

PG_3_5_3B_create_graph_on_long_orbits:
$(ORBITER_PATH)orbiter.out-v-5-
-define-F-finite_field-q-5-end-
-define-P-projective_space-3-F-end-
-define-T-spread_table-P-2:"12".SPREAD_TABLES_5_REG/
-Packing with symmetry assumption T:
-h "H3B_ME" $(PGL_4_5_SUBGROUP_3B_ME)-end-
-N "N3B_ME" $(PGL_4_5_SUBGROUP_3B_ME_NORMALIZER)-end-
-end-
-end-
-Packing choose fixed points PW-
-1-W problem label N3B_ME_fxp_cliques-end-
-L packing long orbits PW-
-orbit length 3-
-clique size 10-
-list of cases from file N3B_ME_fxp_cliques.csv-
-create graphs-
-end-
-print symbols

# 16120 vertices

757
# creates H3B_ME_fpc0_lo.graph

PG_3.5_assume_3B_fpc0_lo_cliques:

```bash
$ (ORBITER_PATH) orbiter.out -v.2
```

```bash
define G -graph -load H3B_ME_fpc0_lo.graph -end
with G -do

graph_theoretic_activity -find_cliques

target_size 10 -end -end

print_symbols
```

# 768 solutions.

# User.time: 8:16

PG_3.5_assume_3B_long_read:

```bash
$ (ORBITER_PATH) orbiter.out -v.5
```

```bash
define F -finite_field -q 5 -end
define P -projective_space 3 F -end
define T -spread_table P 2 "$12" "$SPREAD_TABLES_5_REG/"
define PW -packing with_symmetry_assumption T

H "H3B_ME" $(PGL_4_5_SUBGROUP_3B_ME) -end
N "N3B_ME" $(PGL_4_5_SUBGROUP_3B_ME_NORMALIZER) -end

define PW -packing_choose_fixed_points PW

define L -projecting_problem_label N3B_ME_fixp_cliques -end

define L -packing_long_orbits PW

orbit_length 3 -clique_size 10

list_of_cases_from_file N3B_ME_fixp_cliques.csv

read_solutions

end

print_symbols

# total number of packings = 768

written_file N3B_ME_fixp_cliques_count.csv of size 38

packing_long_orbits::list_of_cases_from_file before save_packings_by_case

packing_long_orbits::save_packings_by_case

written_file H3B_ME_packings.csv of size 150540

# written file H3B_ME_packings.csv of size 150540

PG_3.5_packing0_print:

```bash
$ (ORBITER_PATH) orbiter.out -v.5
```

```bash
define F -finite_field -q 5 -end
define P -projective_space 3 F -end
define T -spread_table P 2 "$12" "$SPREAD_TABLES_5_REG/"
define PW -packing_with_symmetry_assumption T
```

# 758
Let's focus on the text portion:

```
14123  #define PWF-packing.choose_fixed_points_PW-
14124  0-W-problem_label-N3B_ME_fixp_cliques -end-
14125  -with-PWF-do-packing_fixed_points_activity-
14126  -print_packing $(PG_3.5_PACKING_0_WITH_AGO3_FIXP444)-end-
14127  -end
14128  -end
14129  -end

14130
14131
14132
14133
14134
14135  #H-in-the-action-on-point-has-6-orbits-of-length-1-and-50-orbits-of-length-3
14136  #the-fixed-points-make-up-a-line. It is the first of the two special lines.
14137  #line-orbits-2-and-3-make-up-the-second-of-the-special-line.
14138
14139  #H-in-the-action-on-lines-has-270-orbits.
14140  #There are 2-orbits-of-length-1-and-268-orbits-of-length-3
14141  #the-two-orbits-of-length-one-are-the-special-lines.
14142  #the-first-line-is-fixed-pointwise.
14143
14144
14145  #spread-orbits-of-length-1:
14146  #0: Orbit-0:
14147  #0=\{0,36,72,108,144,157,193,229,265,301,314,350,386,422,458,466,502,538,574,610,623,659,695,731,767,805\}
14148
14149  ORBIT_OF_LENGTH_1="0,36,72,108,144,157,193,229,265,301,314,350,386,422,458,466,502,538,574,610,623,659,695,731,767,805"
14150
14151  #define L-packing_long_orbits-PWF-orbit_length-31-clique_size-1-create_graphs-end-
14152
14153
14154
14155
14156  PG_3.5_H3_orbit_of_length_1:
14157  $(ORBITER_PATH)orbiter.out-\-v.5-
14158  -define-F-finite_field-q-5-end-
14159  -define-P-projective_space-3-F-end-
14160  -define-T-spread_table-P-2:"12\".SPREAD_TABLES.5_REG/\-
14161  -with-T-do-
14162  -spread_table_activity-
14163  -find_spread_and_dualize $(ORBIT_OF_LENGTH_1)-end-
14164
14165
14166  #The given spread has index 0 in the spread table
```

# The dual spread has index 1 in the spread table

H31_ORBIT_OF_SPREAD_0="0, 44137, 153432, 45323, 109781, 77407, 29412, 32522, 6582, 61911, 144009, 112494, 91257, 123677, 88268, 13372, 100509, 125783, 80312, 135206, 17508, 22283, 146811, 132608, 53487, 36011, 55803, 116998, 99446, 69752, 73292"

H31_packings.csv

H3B_ME_packings.csv

PG_3_5_packings_compare:

$(ORBITER_PATH) orbiter.out -v 5

$define F - finite_field - q 5 - end

$define P - projective_space - 3 F - end

$define T - spread_table P 2 "12" "SPREAD_TABLES_5_REG/"

$define PW - packing with symmetry_assumption T

$define H "H31_ME" $(PGL_4_5_SUBGROUP_31_ME) - end

$define N "N31_ME" $(PGL_4_5_SUBGROUP_31_ME_NORMALIZER) - end

$define PW - packing choose fixed_points

$define H "H31_ME" $(PGL_4_5_SUBGROUP_31_ME_NORMALIZER) - end

$define PW - packing choose fixed_points

compare_files_of_packings H31_packings.csv H3B_ME_packings.csv

$ORBITER_PATH orbiter.out -v 5 - csv_file_sort_each_row H31_packings.csv

$ORBITER_PATH orbiter.out -v 5 - csv_file_sort_each_row H3B_ME_packings.csv

# H31_packings_sorted.csv

PG_3_5_packings_sort_each_row:

$ORBITER_PATH orbiter.out -v 5 - csv_file_sort_each_row H31_packings.csv

$ORBITER_PATH orbiter.out -v 5 - csv_file_sort_each_row H3B_ME_packings.csv

PG_3_5_assume_31_classify:

$(ORBITER_PATH) orbiter.out -v 2

$define C - combinatorial_objects

$file_of_packings_through_spread_table H31_packings.csv

$define H "H31_ME" $(PGL_4_5_SUBGROUP_31_ME_NORMALIZER) - end

$define F - finite_field - q 5 - end
define F - finite_field - q 5 - end
$\text{combinatorial_object_activity}$

define P - projective_space 3 - F - end

with C - do

combinatorial_object_activity

projective_space 3

finite_field q 5

end

with P - do

projective_space_activity

define F - finite_field - q 5 - end

define P - projective_space 3 - F - end

with P - do

projective_space_activity

define F - finite_field - q 5 - end

define P - projective_space 3 - F - end

with C - do

combinatorial_object_activity

projective_space 3

finite_field q 5

end

with P - do

projective_space_activity

define F - finite_field - q 5 - end

define P - projective_space 3 - F - end
14258 $(\text{ORBITER\_PATH})\text{orbirer.out-v.2}\$
14259 \triangleright \triangleright \text{-define-F-finite_field-q.5-end}$\end
14260 \triangleright \triangleright \text{-define-0-orthogonal_space-0.5:F:-end}$\end
14261 \triangleright \triangleright \text{-with-0-do-orthogonal_space_activity}$\end
14262 \triangleright \triangleright \text{-create_BLT_set-catalogue-1-end}$\end
14263 \triangleright \triangleright \text{-end}$
14264 \text{pdflatex catalogue_q5.isol.tex}$
14265 \text{open-catalogue_q5.isol.pdf}$
14266
14267 \text{BLT\_5_\text{Linear}}$
14268 \triangleright \triangleright $(\text{ORBITER\_PATH})\text{orbirer.out-v.2}\$
14269 \triangleright \triangleright \text{-define-F-finite_field-q.5-end}$\end
14270 \triangleright \triangleright \text{-define-0-orthogonal_space-0.5:F:-end}$\end
14271 \triangleright \triangleright \text{-with-0-do-orthogonal_space_activity}$\end
14272 \triangleright \triangleright \text{-create_BLT_set-family:\text{"Linear"}-end}$\end
14273 \triangleright \triangleright \text{-end}$
14274 \text{pdflatex BLT\text{Linear}q5.tex}$
14275 \text{open-BLT\text{Linear}q5.pdf}$
14276
14277 \text{BLT\_9\_K1}$
14278 \triangleright \triangleright $(\text{ORBITER\_PATH})\text{orbirer.out-v.2}\$
14279 \triangleright \triangleright \text{-define-F-finite_field-q.9-end}$\end
14280 \triangleright \triangleright \text{-define-0-orthogonal_space-0.5:F:-end}$\end
14281 \triangleright \triangleright \text{-with-0-do-orthogonal_space_activity}$\end
14282 \triangleright \triangleright \text{-create_BLT_set-family:\text{"K1"}-end}$\end
14283 \triangleright \triangleright \text{-end}$
14284 \text{pdflatex BLT\text{K1}q9.tex}$
14285 \text{open-BLT\text{K1}q9.pdf}$
14286
14287
14288
14289
14290 \text{BLT\_11\_0}$
14291 \triangleright \triangleright $(\text{ORBITER\_PATH})\text{orbirer.out-v.2}\$
14292 \triangleright \triangleright \text{-define-F-finite_field-q.11-end}$\end
14293 \triangleright \triangleright \text{-define-0-orthogonal_space-0.5:F:-end}$\end
14294 \triangleright \triangleright \text{-with-0-do-orthogonal_space_activity}$\end
14295 \triangleright \triangleright \text{-create_BLT_set-catalogue-0-end}$\end
14296 \triangleright \triangleright \text{-end}$
14297 \text{#pdflatex 0.1.6.2_report.tex}$
14298 \text{#open 0.1.6.2_report.pdf}$
14299
14300
14301 \text{BLT\_11\_\text{Fisher}}$
14302 \triangleright \triangleright $(\text{ORBITER\_PATH})\text{orbirer.out-v.2}\$
14303 \triangleright \triangleright \text{-define-F-finite_field-q.11-end}$\end
14304 \triangleright \triangleright \text{-define-0-orthogonal_space-0.5:F:-end}$\end
BLT_11_Mondello:

BLT_13_FTWKB:

# for K2, q must be congruent to 2 or 3 mod 5

BLT_13_K2:

BLT_13_starter_5:

763
Chapter 13 - Graph Theory

Section 13.1: Creating Graphs

BLT_13_deep_14:
$ (ORBITER_PATH) orbiter.out -v 2 \ 
.define F -finite_field -q 13 -end \ 
.define 0 -orthogonal_space 0 5 F -end \ 
.with 0 -do -orthogonal_space_activity \ 
.BLT_set_starter 14 \ 
.problem_label BLT_q13 -W -depth 14 -end \ 
.end

BLT_13_graphs:
$ (ORBITER_PATH) orbiter.out -v 2 \ 
.define F -finite_field -q 13 -end \ 
.define 0 -orthogonal_space 0 5 F -end \ 
.with 0 -do -orthogonal_space_activity \ 
.BLT_set_graphs 5 0 1 \ 
.end

BLT_13_cliques:
$ (ORBITER_PATH) orbiter.out -v 2 \ 
.loop L 0 38 1 \ 
.define G -graph -load BLT_q13_graph 5 %L.bin -end \ 
.with G -do \ 
.graph_theoretic_activity \ 
.find_cliques -rainbow -target_size 9 -end \ 
.end_loop

#BLT_13_lvl 5
SECTION_CREATING_GRAPHS:

Cycle_13:

triangle_graph:

Chain_232:

Paley_13_graph:

triheiral_pair_graph:
small_graph:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.2\$
  \> -define\text{-graph-edges_as_pairs}5.0,1,0,2,0,3,0,4,1,3,1,4,2,4.\text{-end}

petersen:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.2\$
  \> -define\text{-graph-Johnson-5.2.0.\text{-end}}

Johnson_{6.2.0}:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.2\$
  \> -define\text{-graph-Johnson-6.2.0.\text{-end}}

Hamming_{graph.3}:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.2\$
  \> -define\text{-graph-Hamming.3.2.\text{-end}}

Hamming_{graph.7}:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.2\$
  \> -define\text{-graph-Hamming.7.2.\text{-end}}

#needs\text{-halljanko315.csv}
#from https://www.win.tue.nl/~aeb/drg/graphs/HJ315.html
#There is a unique distance-regular graph Gamma with intersection array {10,8,8,2
\>;1,1,4,5}. It was constructed in Cohen (1981), and uniqueness (given the intersection array) was proved in Cohen & Tits (1985).

HJ_{graph}:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.6$\$
  \> -define\text{-graph-\text{\color{red}}}
  \> -load\text{\color{red}}\text{-csv_no_border-\text{\color{red}}}
  \> halljanko315.csv\$
  \> -end

HJ315_{orbital_graph.3}:
  \>$\text{(ORBITER_PATH)}\text{orbiter.out}-v.2$\$
  \> -define\text{-gens-\text{\color{red}}-vector-\text{\color{red}}-file-\text{\color{red}}}
  \> halljanko315.gens.csv\$-\text{-end}$\$

766
-define G -permutation_group:
-bsgs-halljanko315:"File\_halljanko315."
-define Gamma -graph:
-orbital_graph G.3:
-end:

halljanko315.csv
-distance 2
-end:

Cayley Z11 1mod3:
-define F -finite_field -q:11 -end:
-define S -vector -dense:
"1,1,1,4,1,7,1,10"-end:
-define G -linear_group -AGL.1-F:
-subgroup_by_generators:"Z11":11.^1,1":
-end:

Cayley graph G S:

Cayley Sym4.coxeter:
-define S -vector -dense:"1,0,2,3,0,2,1,3,0,1,3,2"-end:
-define G -permutation_group -symmetric_group 4:
-end:

Cayley graph G S:

Cayley Sym4.star:
-define S -vector -dense:"1,0,2,3,2,1,0,3,3,1,2,0"-end:
-define G -permutation_group -symmetric_group 4:
-end:

Cayley graph G S:
SECTION GRAPH_THEORETIC_ACTIVITIES:

triangle_graph_properties:
  echo $(TRIANGLE_GRAPH) > triangle_graph.csv
  $(ORBITER_PATH) orbiter.out -v 6
  -define G -graph "
  -load Csv_no_border "
  triangle_graph.csv "
  -end "
  -with G -do "
  -graph_theoretic_activity -properties "
  -end

Cycle_13_draw:
  $(ORBITER_PATH) orbiter.out -v 2
  -define Gamma -graph -cycle 13 -end 
  -with Gamma -do 
  -graph_theoretic_activity -export_csv -end 
  -with Gamma -do 
  -graph_theoretic_activity -export_graphviz -end 
  $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix 
  -input_csv_file Cycle_13.csv 
  -box_width 20 -bit_depth 8 -partition 4 13 13 -end 
  dot -Tpng Cycle_13.gv > Cycle_13.png
  #twopi -Tpng Cycle_13.gv > Cycle_13.png 
  #open Cycle_13_draw.bmp 
  #pdflatex Cycle_13_report.tex 
  #open Cycle_13_report.pdf

Cycle_9_eigenvalues:
  $(ORBITER_PATH) orbiter.out -v 2
  -define Gamma -graph "
  -cycle 9 "
  -end "

768
Paley_13.draw:
$\$(ORBITER_PATH)orbiter.out \-v.2\$
$\$(ORBITER_PATH)orbiter.out \-v.2\$
  $\$-define Gamma-\$-graph-Paley.13-\$-end\$
  $\$-with Gamma-\$-do\$
  $\$-graph_theoretic_activity-\$-eigenvalues-\$-end\$
  $\$-with Gamma-\$-do\$
  $\$-graph_theoretic_activity-\$-export_csv-\$-end\$
  $\$-with Gamma-\$-do\$
  $\$-graph_theoretic_activity-\$-export_graphviz-\$-end\$
$\$(ORBITER_PATH)orbiter.out \-v.2\$
  $\$-input_csv_file-Paley.13.csv\$
  $\$-box_width-20-\$-bit_depth-8-\$-partition-4-\$-13-\$-13-\$-end\$
  $\$-dot-Tpng-Paley.13.gv\$-Paley.13.png\$
  $\$-open-Paley.13_draw.bmp\$
Paley_13.eigenvalues:
$\$(ORBITER_PATH)orbiter.out \-v.2\$
$\$(ORBITER_PATH)orbiter.out \-v.2\$
  $\$-define Gamma-\$-graph\$
  $\$-Paley.13\$
  $\$-end\$
  $\$-with Gamma-\$-do\$
  $\$-graph_theoretic_activity-\$-eigenvalues-\$-end\$
  $\$-with Gamma-\$-do\$
  $\$-define Gamma-\$-graph\$
  $\$-Cayley graph G-S-\$-end\$
  $\$-with Gamma-\$-do\$
  $\$-graph_theoretic_activity-\$-draw-\$-end\$
pdflatex Cayley_graph_AGL.11_draw.tex
Cayley\_Sym4\_coxeter eigenvalues and draw:
\$(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot2\$
\(\text{-draw_options}=-\text{xin}2000000--\text{yin}2000000\cdot\)
\(\text{-radius}20000--\text{embedded}--\text{nodes_empty}--\text{end}\cdot\)
\(\text{-define}\cdot\text{S}--\text{vector}--\text{dense}\cdot\)
\(\text{-define}\cdot\text{G}--\text{permutation group}--\text{symmetric group}4\cdot\)
\(\text{-end}\cdot\)
\(\text{-define}\cdot\text{Gamma}--\text{graph}\cdot\)
\(\text{-Cayley graph G S}\cdot\)
\(\text{-graph theoretic activity}--\text{eigenvalues}--\text{end}\cdot\)
\(\text{-with Gamma do}\cdot\)
\(\text{-graph theoretic activity}--\text{draw end}\cdot\)
pdflatex\_Cayley\_graph\_Perm4\_draw.tex
open\_Cayley\_graph\_Perm4\_draw.pdf
#pdflatex\_Cayley\_graph\_Perm4\_eigenvalues.tex
#open\_Cayley\_graph\_Perm4\_eigenvalues.pdf

Cayley\_Sym5\_coxeter draw:
\$(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot2\$
\(\text{-draw_options}=-\text{xin}1000000--\text{yin}1000000\cdot\)
\(\text{-radius}10000--\text{embedded}--\text{nodes_empty}--\text{end}\cdot\)
\(\text{-define}\cdot\text{S}--\text{vector}--\text{dense}\cdot\)
\(\text{-define}\cdot\text{G}--\text{permutation group}--\text{symmetric group}5\cdot\)
\(\text{-end}\cdot\)
\(\text{-define}\cdot\text{Gamma}--\text{graph}\cdot\)
\(\text{-Cayley graph G S}\cdot\)
\(\text{-end}\cdot\)
\(\text{-with Gamma do}\cdot\)
\(\text{-graph theoretic activity}--\text{draw end}\cdot\)
pdflatex\_Cayley\_graph\_Perm5\_draw.tex
open\_Cayley\_graph\_Perm5\_draw.pdf

Cayley\_Sym4\_star eigenvalues and draw:
\$(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot2\$
\(\text{-draw_options}=-\text{xin}1000000--\text{yin}1000000--\text{embedded}--\text{end}\cdot\)
\(\text{-define}\cdot\text{S}--\text{vector}--\text{dense}--1,0,2,3,\cdot0,2,1,3,\cdot0,1,3,2,\cdot0,1,2,4,\cdot0,1,2,4,3\cdot\text{end}\cdot\)
\(\text{-define}\cdot\text{G}--\text{permutation group}--\text{symmetric group}4\cdot\)
\(\text{-end}\cdot\)
\(\text{-define}\cdot\text{Gamma}--\text{graph}\cdot\)
\begin{verbatim}
14679 \> \> -Cayley_graph-G:S/\n14680 \> \> -end/\n14681 \> \> -with-Gamma--do/\n14682 \> \> -graph_theoretic_activity--eigenvalues--end/\n14683 \> \> -with-Gamma--do/\n14684 \> \> -graph_theoretic_activity--draw--end\n14685 pdfLaTeX-Cayley_graph_Perm4.draw.tex\n14686 open-Cayley_graph_Perm4.draw.pdf\n14687 pdfLaTeX-Cayley_graph_Perm4.eigenvalues.tex\n14688 open-Cayley_graph_Perm4.eigenvalues.pdf
14689
14690 14691 small_graph_draw:\n14692 \> $(ORBITER_PATH)orbiter.out--v.2/\n14693 \> \> -define-G--graph--edges_as_pairs.5/\n14694 \> \> \> "0,1,0,2,0,3,0,4,1,3,1,4,2,4"/\n14695 \> \> -end/\n14696 \> \> -with-G--do/\n14697 \> \> -graph_theoretic_activity--export_csv--end/\n14698 \> \> -with-G--do/\n14699 \> \> -graph_theoretic_activity--export_graphviz--end/\n14700 \> \> -with-G--do/\n14701 \> \> -graph_theoretic_activity--save--end\n14702 \> $(ORBITER_PATH)orbiter.out--v.2--draw_matrix/\n14703 \> \> -input_csv_file-graph_v5_e7.csv/\n14704 \> \> -box_width.40--bit_depth.24/\n14705 \> \> -partition.4:"1,1,1,1":"1,1,1,1"--end\n14706 \> dot--Tpng-graph_v5_e7.gv>graph_v5_e7.png\n14707\n14708 \> #creates-graph_v5_e7.csv\n14709 \> #creates-graph_v5_e7.colored_graph\n14710\n14711 14712 petersen_draw:\n14713 \> $(ORBITER_PATH)orbiter.out--v.2/\n14714 \> \> -define-G--graph--Johnson-5:2:0--end/\n14715 \> \> -with-G--do/\n14716 \> \> -graph_theoretic_activity--export_csv--end/\n14717 \> \> -with-G--do/\n14718 \> \> -graph_theoretic_activity--export_graphviz--end/\n14719 \> \> -with-G--do/\n14720 \> \> -graph_theoretic_activity--save--end\n14721 \> $(ORBITER_PATH)orbiter.out--v.2--draw_matrix/\n14722 \> \> -input_csv_file-Johnson_5_2_0.csv/\n14723 \> \> -box_width.40--bit_depth.24--partition.4:"10":"10"--end\n14724 \> dot--Tpng-Johnson_5_2_0.gv>Johnson_5_2_0.png
14725\n\end{verbatim}
```
14726
14727 Johnson_6_2_0_draw:
14728   $(ORBITER_PATH)orbiter.out-v.2-
14729   -define G-graph-Johnson_6_2_0-end-
14730   -with G-do-
14731   -graph_theoretic_activity-export_csv-end-
14732   -with G-do-
14733   -graph_theoretic_activity-export_graphviz-end-
14734   -with G-do-
14735   -graph_theoretic_activity-save-end
14736   $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
14737   -input_csv_file:Johnson_6_2_0.csv-
14738   -box_width:40-bit_depth:24-partition:4:"10","10"-end
14739   dot-Tpng:Johnson_6_2_0.gv>Johnson_6_2_0.png
14740
14741
14742
14743 Hamming_graph_3_draw:
14744   $(ORBITER_PATH)orbiter.out-v.2-
14745   -define G-graph-Hamming_3_2-end-
14746   -with G-do-
14747   -graph_theoretic_activity-export_csv-end-
14748   -with G-do-
14749   -graph_theoretic_activity-export_graphviz-end-
14750   -with G-do-
14751   -graph_theoretic_activity-save-end
14752   $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
14753   -input_csv_file:Hamming_3_2.csv-
14754   -box_width:40-bit_depth:24-
14755   -partition:4:"1,1,1,1,1,1,1,1","1,1,1,1,1,1,1,1"-end
14756   dot-Tpng:Hamming_3_2.gv>Hamming_3_2.png
14757
14758
14759 Hamming_graph_7_draw:
14760   $(ORBITER_PATH)orbiter.out-v.2-
14761   -define G-graph-Hamming_7_2-end-
14762   -with G-do-
14763   -graph_theoretic_activity-export_csv-end-
14764   -with G-do-
14765   -graph_theoretic_activity-export_graphviz-end-
14766   -with G-do-
14767   -graph_theoretic_activity-save-end
14768   $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
14769   -input_csv_file:Hamming_7_2.csv-
14771   dot-Tpng:Hamming_7_2.gv>Hamming_7_2.png
14772
```
Chain_232.properties:

```bash
$(ORBITER_PATH)orbiter.out-v.2\n
define-P1-vector-dense:2,3,2-end\n
define-P2-vector-dense:2,3,2-end\n
define-Gamma-graph\n
chain_graph-P1-P2\n
-end\n
define-Gamma-do\n
with-Gamma-do\n
with-Gamma-do\n
with-Gamma-do\n
end

end
```

Chain_232.eigen:

```bash
$(ORBITER_PATH)orbiter.out-v.2\n
define-P1-vector-dense:2,3,2-end\n
define-P2-vector-dense:2,3,2-end\n
define-Gamma-graph\n
chain_graph-P1-P2\n
-end\n
define-Gamma-do\n
with-Gamma-do\n
with-Gamma-do\n
end

with-Gamma-do\n
with-Gamma-do\n
with-Gamma-do\n
end
```

HJ.properties:

```bash
$(ORBITER_PATH)orbiter.out-v.6\n
define-G-graph\n
load_csv_no_border\n
halljanko315.csv\n
-end\n
define-G-do\n
with-G-do\n
with-G-do\n
end
```

#need the file halljanko315.csv

773
HJ_d2.properties:
$(ORBITER_PATH)orbiter.out -v 6 \n$define-G-graph \n$load_csv_no_border \nhalljanko315.csv \n$distance 2 \n$with-G-do \n-graph_theoretic_activity \n-properties \n-end

PGO_5_2_collinearity_graph:0_5_2.incidence_matrix.csv:
$(ORBITER_PATH)orbiter.out -v 3 \n$define-Inc-vector-file 0_5_2.incidence_matrix.csv-end \n$define-Gamma-graph-collinearity_graph-Inc-end \n$with-Gamma-do \n-graph_theoretic_activity \n-properties \n-end

trihedral_pair_graph_draw:
$(ORBITER_PATH)orbiter.out -v 2 -define-Gamma \n-graph-trihedral_pair_disjointness_graph-end \n$with-Gamma-do \n-graph_theoretic_activity-export_csv-end \n$(ORBITER_PATH)orbiter.out -v 2 -draw_matrix \n-input_csv_file-trihedral_pair_disjointness.csv \n-box_width 20 -depth 8-end \n-open-trihedral_pair_disjointness.draw.bmp

trihedral_pair_graph_draw:0-trihedral_pair_disjointness.csv
-open-trihedral_pair_disjointness.draw.bmp

#Section 13.3: Graph Theory: Classification
SECTION_GRAPH_THEORY_CLASSIFICATION:

tournament_classify_4:
$$(ORBITER\_PATH)orbiter.out\ -v\ -2\\$

> -define-GC\ -graph\_classification\\$
> -n\ -4\\$
> -poset\_classification\_control\\$
> -problem\_label\_tou\_nt\_4\ -depth\ -6\ -draw\_poset\\$
> -draw\_options\ -radius\ 250\ -embedded\ -end\\$
> -end\\$
> -end\\$
> -end\\$
> -end\\$
> -end\\$
> -end\\$
> -end\\$
> -with-GC\ -do\\$
> -graph\_classification\_activity\\$
> -draw\_options\ -embedded\ -radius\ -400\\$
> -line\_width\ -2\ -scale\ -0.15\ -end\\$
> -draw\_graphs\_at\_level\ -6\\$
> -end\\$
> -print\_symbols\\$

depth 4:

> pdf\_latex\_graphs\_v4\_rep\_6_0.tex
> pdf\_latex\_graphs\_v4\_rep\_6_1.tex
Section 13.4: Graph Theory: Clique finding

```
graph_classify_8_r3:
$\text{\textbackslash ORBITER\_PATH}/\text{\textbackslash orbiters.out}\text{\textbackslash -v.3}\
-define GC -graph_classification:\
  -n 8 -regular 3\
-define GC -graph_classification_control:\
  -problem_label graphs_v8_r3 -depth 12 -draw_poset:\
  -draw_options -radius 250\
  -line_width 0.2 -embedded -end:\
  -end:\
-define GC -do\
  -graph_classification_activity:\
  -draw_options -embedded -radius 400\
  -line_width 2 -end:\
  -draw_graphs_at_level 12\
-define GC -end\
-define GC -print_symbols
```

```
pdflatex grafhs_v8_r3_poset_lvl_12.tex
```

```
open-graehs_v8_r3_poset_lvl_12.pdf
```

```
#pdflatex grafhs_v8_r3_rep_12_0.tex
```

```
#open-graehs_v8_r3_rep_12_0.pdf
```

```
#pdflatex grafhs_v8_r3_rep_12_1.tex
```

```
#open-graehs_v8_r3_rep_12_1.pdf
```

```
#pdflatex grafhs_v8_r3_rep_12_2.tex
```

```
#open-graehs_v8_r3_rep_12_2.pdf
```

```
#pdflatex grafhs_v8_r3_rep_12_3.tex
```

```
#open-graehs_v8_r3_rep_12_3.pdf
```

```
#pdflatex grafhs_v8_r3_rep_12_4.tex
```

```
#open-graehs_v8_r3_rep_12_4.pdf
```

```
#pdflatex grafhs_v8_r3_rep_12_5.tex
```

```
#open-graehs_v8_r3_rep_12_5.pdf
```

776
SECTION_GRAPH_THEORY_CLIQUES_FINDING:

small_graph_cliques:

$\text{(ORBITER\_PATH)\texttt{orbiter.out}} \text{-v.10}\$

-define-G-graph-load-graph_v5_e7.colored_graph-end\$

-\text{with-G-do}\$

-graph_theoretic_activity\$

-find_cliques\text{-target\_size-3}\$

-end

#\text{nb\_sol=} 3

small_graph_cliques_Sajeeb:

$\text{(ORBITER\_PATH)\texttt{orbiter.out}} \text{-v.2}\$

-define-G-graph-load-graph_v5_e7.colored_graph-end\$

-\text{with-G-do}\$

-graph_theoretic_activity\$

-find_cliques\text{-Sajeeb\text{-target\_size-3}}\$

-end

#\text{nb\_sol=} 3

Paley_13_aut:

$\text{(ORBITER\_PATH)\texttt{orbiter.out}} \text{-v.2}\$

-define-Gamma-graph-Paley\_13-end\$

-\text{with-Gamma-do}\$

-graph_theoretic_activity\$

-automorphism_group\$

-end

#\text{writes-Paley\_13\_group.makefile}

User-time: 0 of a second, dt=0 tps=.100

#\text{nb\_calls\_to\_densenauty=1}

Paley_13:

$\text{(ORBITER\_PATH)\texttt{orbiter.out}} \text{-v.2}\$

-define-gens-vector-file-Paley\_13\_gens.csv-end\$

-\text{define-G-permutation\_group}\$

-bsgs-Paley\_13\"Paley\_13\"-13-78\"0,1\"-3\_gens-end\$

Paley_13\_cliques\_classify:

$\text{(ORBITER\_PATH)\texttt{orbiter.out}} \text{-v.4}\$

-define-gens-vector-file-Paley\_13\_gens.csv-end\$

#777
-define G -permutation_group \ 
-bsgs Paley_13 "Paley\_13".13.78."0,1".3.gens -end \ 
 DEFINE Gamma -graph Paley.13 -end \ 
-with G -do \ 
group_theoretic_activity \ 
poset_classification_control \ 
-problem_label Paley13_cliques \ 
clique_test Gamma \ 
depth 5 \ 
-orbits_on_subsets 5 \ 
-report \ 
-end \ 

Paley_13_cliques: \ 
$\$(ORBITER\_PATH)\ orbs\_out -v.10 \ 
 DEFINE Gamma -graph Paley.13 -end \ 
-with Gamma -do \ 
-graph_theoretic_activity \ 
-find cliques -target_size 3 \ 
-end \ 

PG_5_2_cliques: 0_5_2_incidence_matrix.csv \ 
$\$(ORBITER\_PATH)\ orbs\_out -v.3 \ 
-find Inc -vector file 0_5_2_incidence_matrix.csv -end \ 
-find Gamma -graph collinearity_graph Inc -end \ 
-with Gamma -do \ 
-graph_theoretic_activity \ 
-find cliques -target_size 3 -end \ 
-end \ 

HJ_d2_c5: \ 
$\$(ORBITER\_PATH)\ orbs\_out -v.6 \ 
-find Gamma -graph \ 
-load_csv_no_border \ 
-halljanko315.csv \ 
-distance 2 \ 

User time: 0.01 of a second, dt=1 tps=.100
15055 #graph_theoretic_activity::perform_activity:Gr->label=halljanko315 nb_sol=.26208
15056 0
15057 #graph_theoretic_activity::perform_activity:Gr->label=GroupPerm315 Orbital3 nb_sol=.1008
15058 #GroupPerm315 Orbital3 sol.csv
15059
15060 HJ64_cliques5:
15061
15062 $\text{(ORBITER\_PATH)orbiter.out\,-v}\,6$
15063 $\text{-define\,-Gamma\,-graph}$
15064 $\text{-load}$
15065 $\text{-Group\_Perm315\_Orbital\_3\,colored\_graph}$
15066 $\text{-end}$
15067 $\text{-with\,-Gamma\,-do}$
15068 $\text{-graph\,-theoretic\,-activity}$
15069 $\text{-find\,-cliques\,-target\,-size\,5}$
15070 $\text{-end}$
15071 $\text{-end}$
15072 $\text{HJ64\,cliques5}$
15073 $\text{classify:}$
15074 $\text{-define\,-Gamma\,-graph}$
15075 $\text{-load}$
15076 $\text{-Group\_Perm315\_Orbital\_3\,colored\_graph}$
15077 $\text{-end}$
15078 $\text{-with\,-Gamma\,-do}$
15079 $\text{-graph\,-theoretic\,-activity}$
15080 $\text{-find\,-cliques\,-target\,-size\,5}$
15081 $\text{-end}$
15082 $\text{-end}$
15083 $\text{HJ64\,cliques5\,classify:}$
15084 $\text{-define\,-Gamma\,-graph}$
15085 $\text{-load}$
15086 $\text{-Group\_Perm315\_Orbital\_3\,colored\_graph}$
15087 $\text{-end}$
15088 $\text{-file\,-halljanko315\_gens\,csv}$
15089 $\text{-end}$
15090 $\text{-define\,-G\,-permutation\,-group}$
15091 $\text{-bsgs\,-halljanko315\,"File\,halljanko315\,"}$
15092 $\text{-315\,-1209600\,0\,1\,42\,95\,-6\,-gens\,-end}$
15093 $\text{-with\,-G\,-do}$
15094 $\text{-group\,-theoretic\,-activity}$
15095 $\text{-poset\,-classification\,-control}$
15096 $\text{-problem\,-label\,HJ64\,cliques}$
15097 $\text{-clique\,-test\,-Gamma}$
15098
15099

779
Combinatorial Objects

SECTION COMBINATORIAL OBJECTS:

Hirschfeld q4 from set:
$ORBITER\_PATH$orbiter.out--v.4\n DEFINE H -set--here\n $(HIRSCHFELD\_SURFACE\_Q4\_SET\_OF\_POINTS)\n -end\n -define C -combinatorial_objects\n -set_of_points H\n -end

hyperoval 16 create:
$ORBITER\_PATH$orbiter.out--v.2\n DEFINE C -combinatorial_objects\n -set_of_points $(HYPEROVAL\_16\_16320)\n -set_of_points $(HYPEROVAL\_16\_144)\n
15147 ▷ ▷ -end\n15148
15149
15150 EC_read::elliptic_curve_b1_c3_q11.txt
15151 ▷ $(ORBITER_PATH)orbiter.out-v.4\n15152 ▷ ▷ -define-C:-combinatorial_objects:\n15153 ▷ ▷ ▷ -file_of_points::elliptic_curve_b1_c3_q11.txt\n15154 ▷ ▷ -end
15155
15156
15157
15158 PG_3.5_assume_31_read:
15159 ▷ $(ORBITER_PATH)orbiter.out-v.2\n15160 ▷ ▷ -define-C:-combinatorial_objects:\n15161 ▷ ▷ ▷ -file_of_packings_through_spread_table:\n15162 ▷ ▷ ▷ ▷ H31_packings.csv\n15163 ▷ ▷ ▷ ▷ SPREAD_TABLES_5_REG/spread_25_spreads.csv\n15164 ▷ ▷ ▷ ▷ 5\n15165 ▷ ▷ -end
15166
15167
15168
15169 LS_AG_2.3_read:
15170 ▷ $(ORBITER_PATH)orbiter.out-v.2\n15171 ▷ ▷ -define-C:-combinatorial_objects:\n15172 ▷ ▷ ▷ -file_of_designs:\n15173 ▷ ▷ ▷ ▷ solutions.csv:9:84:3:12\n15174 ▷ ▷ -end
15175
15176
15177
15178 geo_7.3_read:
15179 ▷ $(ORBITER_PATH)orbiter.out-v.10\n15180 ▷ ▷ -draw_incidence_structure_description:\n15181 ▷ ▷ ▷ -width:60:-with:10:6:-end\n15182 ▷ ▷ -define-C:-combinatorial_objects:\n15183 ▷ ▷ ▷ -file_of_incidence_geometries:\n15184 ▷ ▷ ▷ ▷ 7.3.inc:7:7:21\n15185 ▷ ▷ -end
15186
15187
15188
15189 Desargues_path_lex_least_read:
15190 ▷ echo:$($DESARGUES_PATH_LEX_LEAST):>Desargues_path_lex_least.inc
15191 ▷ $(ORBITER_PATH)orbiter.out-v.10\n15192 ▷ ▷ -draw_incidence_structure_description:\n15193 ▷ ▷ ▷ -width:60:-with:10:6:-end\n
-define C=combinatorial_objects:\
-define -file_of_incidence_geometries_by_row_ranks:\
Desargues_path_lex_least.inc:10:10:3:\
-end

geoPasch_read:
$(ORBITER_PATH)orbiter.out-v.10:\
-define C=combinatorial_objects:\
-file_of_incidence_geometries:\
pasch.inc:6:4:12:\
-end

geoPasch_given:
$(ORBITER_PATH)orbiter.out-v.10:\
-define C=combinatorial_objects:\
-incidence_geometry:\
"0,1,4,6,8,11,13,14,17,19,22,23":\
6:4:12:\
-end

SECTION OVERVIEW CANONICAL FORMS:
SECTION OBJECTS_IN_PROJECTIVE_SPACE:

EC_canon: elliptic_curve_b1_c3_q11.txt

$(ORBITER_PATH)orbiter.out -v 4

· define C - combinatorial_objects

· file_of_points_ellipsis_curve_b1_c3_q11.txt

· end

· define F - finite_field - q 11 - end

· define P - projective_space - 2 F - end

· with C - do

· -combinatorial_object_activity

· -canonical_form_PG_P

· -classification_prefix_EC

· -label_EC

· -save_ago

· -max_TDO_depth 4

· -end

· -report

· -prefix_EC

· -export_flag_orbits

· -show_TDO

· -show_TDA

· -dont_show_incidence_matrices

· -export_group

· -end

· -end

pdflatex EC_classification.tex

open EC_classification.pdf

$(ORBITER_PATH)orbiter.out -v 2 -draw_matrix

· -input_csv_file_EC_object0_TDA_flag_orbits.csv

· -secondary_input_csv_file_EC_object0_TDA.csv

· -box_width 20 - bit_depth 24

· -end

open EC_object0_TDA_flag_orbits_draw.bmp

Hirschfeld_q4.c: Hirschfeld_surface_q4.txt

$(ORBITER_PATH)orbiter.out -v 6

· define C - combinatorial_objects

· file_of_points_Hirschfeld_surface_q4.txt

· -end

· -define F - finite_field - q 4 - end

· -define P - projective_space - 3 F - end
Dickson_sets_stabilizer:
$$(ORBITER_PATH)orbiter.out-v.3$$

with C-do-
-combinatorial_object_activity-
canonical_form_PG_P-
classification_prefix-Hirschfeld_surface_q4-
save_ag-
max_TDO_depth-10-
end-
report-
-prefix-Hirschfeld_surface_q4-
export_flag_orbits-
show_TDO-
show_TDA-
dont_show_incidence_matrices-
export_group-
end-
end

Hirschfeld_q4_set_c:
$(ORBITER_PATH)orbiter.out-v.4
define-H-set--here-
$(HIRSCHFELD_SURFACE_Q4_SET_OF_POINTS)-
end-
define-C-combinatorial_objects-
set_of_points-H-
end-
define-F-finite_field-q.4-end-
define-P-projective_space-3-F-end-
with-C-do-
combinatorial_object_activity-
canonical_form_PG_P-
classification_prefix-Hirschfeld_surface_q4-
save_ag-
end

Dickson_sets_stabilizer:
$$(ORBITER_PATH)orbiter.out-v.3$$

define-C-combinatorial_objects-
-set_of_points"0,1,2,5,6"
-defineF-finite_field-q2-end
-defineP-projective_space-3F-end
-withC-do-
-combinatorial_object_activity-
-canonical_form_PG_P-
-classification_prefix-Dickson_sets-
-save_ago-
-end-
-report-
-end
pdlatex-Dickson_sets_classification.tex
open-Dickson_sets_classification.pdf

Endrass_7c::Endrass_F7.txt
$p(ORBITER\ PATH)orbiter.out-v.2-
defineC-combinatorial_objects-
-file_of_points-Endrass_F7.txt-
-end-
defineF-finite_field-q7-end-
defineP-projective_space-3F-end-
-withC-do-
-combinatorial_object_activity-
-canonical_form_PG_P-
-classification_prefix-Endrass_F7-
save_ago-
-end-
-report-
-end
pdlatex-Endrass_F7_classification.tex
open-Endrass_F7_classification.pdf
# group-order is 32

```bash
hyperoval_16_c:

$ (ORBITER_PATH) orbiter.out -v 2 \
-define C -combinatorial_objects \
-set_of_points $(HYPEROVAL_16_16320) \
-set_of_points $(HYPEROVAL_16_144) \
-end \
-define F -finite_field -q 16 -end \
-end \
-with C -do \
-combinatorial_object_activity \
-cannotical_form PG P \
-classification_prefix hyperoval_q16 \
-label hyperoval_q16 \
save ago \
save transversal \
-max TDO depth 10 \
-end \
-report \
-prefix hyperoval_q16 \
-export_flag_orbits \
-show TDO \
-show TDA \
dont_show incidence_matrices \
-export group \
-end \
pdflatex hyperoval_q16_classification.tex
open-hyperoval_q16_classification.pdf

$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix \
-input_csv_file hyperoval_q16_object0_TDA_flag_orbits.csv \
-secondary_input_csv_file hyperoval_q16_object0_TDA.csv \
-box_width 4 -bit_depth 24 \
-end \
opend hyperoval_q16_object0_TDA_flag_orbits_draw.bmp

$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix \
-input_csv_file hyperoval_q16_object1_TDA_flag_orbits.csv \
-secondary_input_csv_file hyperoval_q16_object1_TDA.csv \
-box_width 4 -bit_depth 24 \
-end \
opend hyperoval_q16_object1_TDA_flag_orbits_draw.bmp
```

786
cubic_curves PG 2 8 canonic: $(\text{ORBITER PATH})/\text{or}\text{biter.out}\ -v\cdot 6\ \$

```bash
define \text{C} - \text{combinatorial objects}: 
  set_of_points: "2,3,28,46,51,61,40,71". 
-define \text{F} - \text{finite field}\ - q\cdot 8 - end; 
-define \text{P} - \text{projective space}\ - 2\cdot \text{F} - end; 
-combinatorial_object_activity: 
-do 
  -canonical_form \text{PG}\cdot \text{P}; 
  -classification_prefix cc 8; 
  -save \text{ago};
  -max_TD0_depth 10; 
  -report; 
-end 
define F - finite field\ - q\cdot 7 - end; 
define P - projective space\ - 3\cdot \text{F} - end; 
with C - do; 
-combinatorial_object_activity: 
-canonical_form \text{PG}\cdot \text{P}; 
-classification_prefix_surface 15\cdot \text{lines} q7; 
-save \text{ago};
-save_transversal; 
-end 
end 
desolve cc 8_classification.tex 
open cc 8_classification.pdf 

F_{\alpha \beta \gamma \delta} classify q7 nauty: F_{\alpha \beta \gamma \delta} q7 points.txt 
$\text{ORBITER PATH})/\text{or}\text{biter.out}\ -v\cdot 6\ \$
define \text{C} - \text{combinatorial objects}: 
  file_of_points: 
  F_{\alpha \beta \gamma \delta} q7 points.txt. 
-end 
define F - finite field\ - q\cdot 7 - end; 
define P - projective space\ - 3\cdot \text{F} - end; 
with C - do; 
-combinatorial_object_activity: 
-canonical_form \text{PG}\cdot \text{P}; 
-classification_prefix_surface 15\cdot \text{lines} q7; 
-save \text{ago};
-save_transversal; 
-end 
end 
pdflatex cc 8_classification.tex 
open cc 8_classification.pdf 

#4:38

#User:time:4:12 on Mac 

#-6-orbits
ovoid_q8.canon: ovoid_q8.txt

$(ORBITER_PATH)orbiter.out -v 6

define C - combinatorial_objects

define F - finite_field - q 8 - end

define P - projective_space - 3 F - end

with C - do

combinatorial_object_activity

canonical_form PG P

classification_prefix ovoid

label ovoid

save ago

max TDO depth 4

end

report

prefix ovoid

show TDO

show TDA

don't show incidence_matrices

export group

end

dep

dep

ovoid_ST_q8.canon: ovoid_ST_q8.txt

$(ORBITER_PATH)orbiter.out -v 6

define C - combinatorial_objects

define F - finite_field - q 8 - end

define P - projective_space - 3 F - end
15522 \>
15523 \ >\ with-C-do\ \
15524 \ >\ combinatorial_object_activity\ \
15525 \ >\ >\ canonical_form_PGP\ \
15526 \ >\ >\ >\ classification_prefix-void\:ST\:\ 
15527 \ >\ >\ >\ -label-void\:ST\:\ 
15528 \ >\ >\ >\ -save Ago\:\ 
15529 \ >\ >\ >\ -max\_TD0_depth\:4\:\ 
15530 \ >\ >\ >\ -end\:\ 
15531 \ >\ >\ >\ -report\:\ 
15532 \ >\ >\ >\ -prefix-void\:ST\:\ 
15533 \ >\ >\ >\ -show\_TD0\:\ 
15534 \ >\ >\ >\ -show\_TDA\:\ 
15535 \ >\ >\ >\ -dont\_show\_incidence\_matrices\:\ 
15536 \ >\ >\ >\ -export\_group\:\ 
15537 \ >\ >\ >\ -end\:\ 
15538 \ >\ >\ -end\ 
15539 \ >\ open-void\:ST\_classification.pdf
15540
15541 # group\_order\:87360\:=\:3\:*\:29120
15542 SUZUKI\:8\_GENERATORS="\"\ 
15543 1,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,1,\ 
15544 1,0,0,0,0,6,0,0,0,0,2,0,0,0,0,3,0,\ 
15545 1,0,0,0,1,1,1,0,0,0,1,0,1,0,0,1,0,\ 
15546 1,0,0,0,3,6,2,2,5,0,2,0,3,0,6,3,2,\ 
15547 0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,1,2"\ 
15548 Suzuki\:8:
15549 $(\text{ORBITER\_PATH})\text{orbiter.out}\:=-v\:6:\ 
15550 \ >\ $\{(\text{ORBITER\_PATH})\text{orbiter.out}\:=-v\:6\:\ 
15551 \ >\ \ >\ -define\_F=-finite\_field\:-q\:8\:-end\:\ 
15552 \ >\ \ >\ -define\_gens=-vector\:-field\_F\:\ 
15553 \ >\ \ >\ \ >\ -compact\:$\{(\text{SUZUKI\:8\_GENERATORS})\:-end\:\ 
15554 \ >\ \ >\ \ >\ -define\_G=linear\_group\=-PGGL\:4\:8\:\ 
15555 \ >\ \ >\ \ >\ -subgroup\_by\_generators\:"\text{Sz8}"\:"87360"\:5\_gens\:\ 
15556 \ >\ \ >\ \ >\ -end\:\ 
15557 \ >\ \ >\ \ >\ -with\:G\:-do\:\ 
15558 \ >\ \ >\ \ >\ -group\_theoretic\_activity\:\ 
15559 \ >\ \ >\ \ >\ -report\:\ 
15560 \ >\ \ >\ \ >\ -end\ 
15561 \ >\ pdflatex-PGGL\:4\:8\_Subgroup\_Sz8\:87360\_report.tex
15562 \ >\ open-PGGL\:4\:8\_Subgroup\_Sz8\:87360\_report.pdf
15563
15564
15565 #####################################################################
15566 # Section 15.3: Incidence Geometries
15567
15568
SECTION_INCIDENCE_GEOMETRIES:

geo_7_3_c:
$\$(ORBITER_PATH)\$\{\texttt{orbiter.out-v.10}\
\$\$(ORBITER_PATH)\$\{\texttt{-draw incidence structure description}\
\$\$(ORBITER_PATH)\$\{\texttt{-width 60 -with 10 6 -end}\
\$\$(ORBITER_PATH)\$\{\texttt{-define C -combinatorial objects}\
\$\$(ORBITER_PATH)\$\{\texttt{-file of incidence geometries 7_3.inc 7_7.21}\
\$\$(ORBITER_PATH)\$\{\texttt{-with C -do}\
\$\$(ORBITER_PATH)\$\{\texttt{-combinatorial object activity}\
\$\$(ORBITER_PATH)\$\{\texttt{-canonical form}\
\$\$(ORBITER_PATH)\$\{\texttt{-classification prefix 7_3}\
\$\$(ORBITER_PATH)\$\{\texttt{-label 7_3}\
\$\$(ORBITER_PATH)\$\{\texttt{-save ago}\
\$\$(ORBITER_PATH)\$\{\texttt{-save transversal}\
\$\$(ORBITER_PATH)\$\{\texttt{-end}\
\$\$(ORBITER_PATH)\$\{\texttt{-report}\
\$\$(ORBITER_PATH)\$\{\texttt{-prefix 7_3}\
\$\$(ORBITER_PATH)\$\{\texttt{-export flag orbits}\
\$\$(ORBITER_PATH)\$\{\texttt{-show incidence matrices}\
\$\$(ORBITER_PATH)\$\{\texttt{-export group}\
\$\$(ORBITER_PATH)\$\{\texttt{-end}\
\$\$(ORBITER_PATH)\$\{\texttt{-end}\
\$\$(ORBITER_PATH)\$\{\texttt{pdflatex 7_3_classification.tex}\
\$\$(ORBITER_PATH)\$\{\texttt{open 7_3_classification.pdf}\
\$\$(ORBITER_PATH)\$\{\texttt{\textbf{\$\$(ORBITER_PATH)\$\{\texttt{orbiter.out-v.2 -draw matrix}\
\$\$(ORBITER_PATH)\$\{\texttt{-input csv file 7_3 object0 TDA flag orbits.csv}\
\$\$(ORBITER_PATH)\$\{\texttt{-secondary input csv file 7_3 object0 TDA.csv}\
\$\$(ORBITER_PATH)\$\{\texttt{-box width 32 -bit depth 24}\
\$\$(ORBITER_PATH)\$\{\texttt{-end}\
\$\$(ORBITER_PATH)\$\{\texttt{\textbf{\$\$(ORBITER_PATH)\$\{\texttt{orbiter.out-v.2 -draw matrix}\
\$\$(ORBITER_PATH)\$\{\texttt{-input csv file 7_3 object0 INP flag orbits.csv}\
\$\$(ORBITER_PATH)\$\{\texttt{-secondary input csv file 7_3 object0 INP.csv}\
\$\$(ORBITER_PATH)\$\{\texttt{-box width 32 -bit depth 24}\
\$\$(ORBITER_PATH)\$\{\texttt{-end}\
\$\$(ORBITER_PATH)\$\{\texttt{open 7_3 object0 INP flag orbits draw bmp}\
\$\$(ORBITER_PATH)\$\{\texttt{790}
15616  >  >  > -width=60-\with_{10}-6-\end\
15617  >  >  -define Test_lines-\set-loop 4\cdot 11.1-\end\
15618  >  >  -define C-\combinatorial_objects\
15619  >  >  > -file_of_incidence_geometries 10\_3.inc 10\_10 30\
15620  >  >  -end\
15621  >  >  -with_c\-do\
15622  >  >  -combinatorial_object_activity\
15623  >  >  > -canonical_form\
15624  >  >  >  > -classification_prefix 10\_3\
15625  >  >  >  > -label 10\_3\
15626  >  >  >  > -save ago\
15627  >  >  >  > -save_transversal\
15628  >  >  >  -end\
15629  >  >  >  -report\
15630  >  >  >  > -prefix 10\_3\
15631  >  >  >  > -export flag orbits\
15632  >  >  >  > -show incidence matrices\
15633  >  >  >  > -export group\
15634  >  >  >  -end\
15635  >  >  -end\
15636  >  pdflatex 10\_3\_classification.tex
15637  >  open 10\_3\_classification.pdf
15638  >  $(\text{ORBITER\_PATH})\text{orbiter.out-\text{-v.2-\text{-draw_matrix}}}$
15639  >  >  > -input_csv_file 10\_3\_object7\_TDA\_flag_orbits.csv\
15640  >  >  > -secondary_input_csv_file 10\_3\_object7\_TDA.csv\
15641  >  >  > -box_width 16-\bit_depth 24\
15642  >  >  > -end\
15643  >  $(\text{ORBITER\_PATH})\text{orbiter.out-\text{-v.2-\text{-draw_matrix}}}$
15644  >  >  > -input_csv_file 10\_3\_object7\_INP\_flag_orbits.csv\
15645  >  >  > -secondary_input_csv_file 10\_3\_object7\_INP.csv\
15646  >  >  > -box_width 16-\bit_depth 24\
15647  >  >  > -end\
15648  
15649  
15650  
15651  
15652  
15653  \text{geo} 10\_3\_c\_lex least:
15654  >  $(\text{ORBITER\_PATH})\text{orbiter.out-\text{-v.10-}}$
15655  >  >  -draw incidence structure description\
15656  >  >  > -width=60-\with_{10}-6-\end\
15657  >  >  > -define Test_lines-\set-loop 4\cdot 11\_1-\end\
15658  >  >  -define Geo-\geometry builder\
15659  >  >  > -V.10-B.10-\text{TDO.3-\text{-fuse.1}}\
15660  >  >  > -fname GEO\_10\_3\
15661  >  >  > -test Test_lines\
15662  >  >  -end\n
791
15710 \> \> -end\.
15711 \> \> -with\.C-\-do-\.
15712 \> \> -combinatorial\_object\_activity-\.
15713 \> \> \> -canonical\_form-\.
15714 \> \> \> \> -classification\_prefix\:14\_3-\.
15715 \> \> \> \> -label\:14\_3-\.
15716 \> \> \> \> -save\_ago-\.
15717 \> \> \> \> -save\_transversal-\.
15718 \> \> \> \> -end-\.
15719 \> \> -end
15720
15721
15722 #\> \> \> -report-\.
15723 #\> \> \> \> -prefix\:14\_3-\.
15724 #\> \> \> \> -export\_flag\_orbits-\.
15725 #\> \> \> \> -show\_incidence\_matrices-\.
15726 #\> \> \> \> -export\_group-\.
15727 #\> \> \> \> -end-\.
15728
15729
15730 geo\:15\_3\_c:
15731 \> $(\text{ORBITER\_PATH})\text{orbiter.out-\-v\:2-}\.
15732 \> \> -draw\_incidence\_structure\_description-\.
15733 \> \> \> -width\:50\-\-with\:10\:5-\-end-\.
15734 \> \> \> -define\:C\-combinatorial\_objects-\.
15735 \> \> \> \> -file\_of\_incidence\_geometries\:15\_3\_inc\:15\:15\:45-\.
15736 \> \> \> -end-\.
15737 \> \> \> -with\:C\-do-\.
15738 \> \> \> -combinatorial\_object\_activity-\.
15739 \> \> \> \> -canonical\_form-\.
15740 \> \> \> \> -classification\_prefix\:10\_3-\.
15741 \> \> \> \> -label\:10\_3-\.
15742 \> \> \> \> -save\_ago-\.
15743 \> \> \> -end
15744 \> pdflatex\:15\_3\_classification\_tex
15745 \> open\:15\_3\_classification\_pdf
15746
15747 TFC\:24\_3\_c:
15748 \> echo $(\text{FILE\_24\_3\_TFC\_INC})\>24\_3\_TFC\_inc
15749 \> $(\text{ORBITER\_PATH})\text{orbiter.out-\-v\:6-}\.
15750 \> \> -define\:C\-combinatorial\_objects-\.
15751 \> \> \> -file\_of\_incidence\_geometries\:24\_3\_TFC\_inc\:24\:24\:72-\.
15752 \> \> -end-\.
15753 \> \> -with\:C\-do-\.
15754 \> \> -combinatorial\_object\_activity-\.
15755 \> \> \> -canonical\_form-\.
15756 \> \> \> \> -classification\_prefix\:24\_3\_TFC-\.

793
15757 \triangleright \triangleright -label 24_3\_TFC\n15758 \triangleright \triangleright -save\_ago\n15759 \triangleright \triangleright -end\n15760 \triangleright \triangleright -report\n15761 \triangleright \triangleright \triangleright -prefix 24_3\_TFC\n15762 \triangleright \triangleright \triangleright -export\_flag\_orbits\n15763 \triangleright \triangleright \triangleright -show\_TDO\n15764 \triangleright \triangleright \triangleright -show\_TDA\n15765 \triangleright \triangleright \triangleright -show\_incidence\_matrices\n15766 \triangleright \triangleright -end\n15767 \triangleright -end\n15768 pdflatex 24_3\_TFC\_classification.tex
15769 open 24_3\_TFC\_classification.pdf
15770 $(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_2\_draw\_matrix$
15771 \triangleright -input\_csv\_file 24_3\_TFC\_object2\_TDA\_flag\_orbits.csv\n15772 \triangleright -secondary\_input\_csv\_file 24_3\_TFC\_object2\_TDA.csv\n15773 \triangleright -box\_width 40 -bit\_depth 24\n15774 \triangleright -end\n15775 open 24_3\_TFC\_object2\_TDA\_flag\_orbits\_draw.bmp
15776 15777
15778 geo 40_4\_g4.c:
15779 $(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_2\n15780 \triangleright -draw\_incidence\_structure\_description\n15781 \triangleright \triangleright -width 50 -with\_10\_5 -end\n15782 \triangleright -define C -combinatorial\_objects\n15783 \triangleright \triangleright -file\_of\_incidence\_geometries 40_4\_g4.inc 40_4\_g4_160\n15784 \triangleright -end\n15785 \triangleright -with C -do\n15786 \triangleright -combinatorial\_object\_activity\n15787 \triangleright \triangleright -canonical\_form\n15788 \triangleright \triangleright \triangleright -classification\_prefix 40_4\_g4\n15789 \triangleright \triangleright \triangleright -label 40_4\_g4\n15790 \triangleright \triangleright \triangleright -save\_ago\n15791 \triangleright \triangleright \triangleright -end\n15792 \triangleright \triangleright -report\n15793 \triangleright \triangleright \triangleright -prefix 40_4\_g4\n15794 \triangleright \triangleright \triangleright -export\_flag\_orbits\n15795 \triangleright \triangleright \triangleright -show\_TDO\n15796 \triangleright \triangleright \triangleright -show\_TDA\n15797 \triangleright \triangleright \triangleright -show\_incidence\_matrices\n15798 \triangleright \triangleright -end\n15799 \triangleright -end
15800 pdflatex 40_4\_g4\_classification.tex
15801 open 40_4\_g4\_classification.pdf
15802 15803 geo 17_3\_g4.c:
\verb|$\{ORBITER\_PATH\}orbiter.out\ -v.2:\$
\verb|$\{ORBITER\_PATH\}draw.incidence_structure_description\$
\verb|-width.50\ -with.10.5\ -end\$
\verb|$\{ORBITER\_PATH\}define.C\ -combinatorial\_objects\$
\verb|--file_of_incidence_geometries.17.3\_g4.inc.17.17.51\$
\verb|$\{ORBITER\_PATH\}width.50\ -with.10.5\ -end\$
\verb|$\{ORBITER\_PATH\}define.C\ -combinatorial\_object\_activity\$
\verb|--canonical_form\$
\verb|$\{ORBITER\_PATH\}classification_prefix.17.3\_g4\$
\verb|--label.17.3\_g4\$
\verb|$\{ORBITER\_PATH\}save.ago\$
\verb|$\{ORBITER\_PATH\}report\$
\verb|$\{ORBITER\_PATH\}prefix.17.3\_g4\$
\verb|--export_flag.orbits\$
\verb|$\{ORBITER\_PATH\}show.TDO\$
\verb|$\{ORBITER\_PATH\}show.TDA\$
\verb|$\{ORBITER\_PATH\}show.incidence\_matrices\$
\verb|$\{ORBITER\_PATH\}end\$
\verb|$\{ORBITER\_PATH\}end\$
\verb|pdf\_latex.17.3\_g4\_classification.tex\$
\verb|open.17.3\_g4\_classification.pdf\$

AG\_2\_3\_c:\ AG\_2\_3.inc

\verb|$\{ORBITER\_PATH\}orbiter.out\ -v.2:\$
\verb|$\{ORBITER\_PATH\}define.C\ -combinatorial\_objects\$
\verb|--file_of_incidence_geometries.9.12.36\$
\verb|$\{ORBITER\_PATH\}end\$
\verb|$\{ORBITER\_PATH\}end\$

AG\_2\_3\_c:\ AG\_2\_3.inc

\verb|pdflatex.17.3\_g4\_classification.tex\$
\verb|open.17.3\_g4\_classification.pdf\$

15827

15828

15829

15830

15831

15832

15833

15834

15835

15836

15837

15838

15839

15840

15841

15842

15843

15844

15845

15846

15847

15848

15849

15850

795
15851 ▷ pdflatex AG_2_3_classification.tex
15852 ▷ open AG_2_3_classification.pdf
15853 ▷ $(ORBITER\_PATH)\$orbiter.out -v 2 -draw_matrix\$
15854 ▷ ▷ -input_csv_file AG_2_3_object0_INP_flag_orbits.csv\
15855 ▷ ▷ -secondary_input_csv_file AG_2_3_object0_INP.csv\
15856 ▷ ▷ -box_width 40 -bit_depth 24\
15857 ▷ ▷ -end
15858 ▷ open AG_2_3_object0_INP_flag_orbits_draw.bmp
15859
15860
15861
15862
15863 geo_LSQ6.c:
15864 ▷ $(ORBITER\_PATH)\$orbiter.out -v 10\$
15865 ▷ ▷ -draw_incidence_structure_description\$
15866 ▷ ▷ ▷ -width 60 -with 10 6 -end\$
15867 ▷ ▷ -define C -combinatorial_objects\$
15868 ▷ ▷ ▷ -file_of_incidence_geometries\$
15869 ▷ ▷ ▷ ▷ LSQ6.inc 18 39 126\$
15870 ▷ ▷ ▷ -end\$
15871 ▷ ▷ -with C do\$
15872 ▷ ▷ -combinatorial_object_activity\$
15873 ▷ ▷ ▷ -canonical_form\$
15874 ▷ ▷ ▷ ▷ -classification_prefix LSQ6\$
15875 ▷ ▷ ▷ ▷ -label LSQ6\$
15876 ▷ ▷ ▷ ▷ -save_ago\$
15877 ▷ ▷ ▷ ▷ -save_transversal\$
15878 ▷ ▷ ▷ -end\$
15879 ▷ ▷ ▷ -report\$
15880 ▷ ▷ ▷ ▷ -prefix LSQ6\$
15881 ▷ ▷ ▷ ▷ -export_flag_orbits\$
15882 ▷ ▷ ▷ ▷ -show_incidence_matrices\$
15883 ▷ ▷ ▷ ▷ -export_group\$
15884 ▷ ▷ ▷ -end\$
15885 ▷ ▷ -end
15886 ▷ pdflatex LSQ6_classification.tex
15887 ▷ #open LSQ6_classification.pdf
15888 ▷ $(ORBITER\_PATH)\$orbiter.out -v 2 -draw_matrix\$
15889 ▷ ▷ -input_csv_file LSQ6_object0_TDA_flag_orbits.csv\$
15890 ▷ ▷ -secondary_input_csv_file LSQ6_object0_TDA_flag_orbits.csv\$
15891 ▷ ▷ -box_width 32 -bit_depth 24\$
15892 ▷ ▷ -end
15893 ▷ $(ORBITER\_PATH)\$orbiter.out -v 2 -draw_matrix\$
15894 ▷ ▷ -input_csv_file LSQ6_object0_INP_flag_orbits.csv\$
15895 ▷ ▷ -secondary_input_csv_file LSQ6_object0_INP_flag_orbits.csv\$
15896 ▷ ▷ -box_width 32 -bit_depth 24\$
15897 ▷ ▷ -end
open-LSQ6.object0_INP_flag_orbits_draw.bmp

#define F - finite_field - q 25 - end\n
#define P - projective_space - 2 F - end\n
with P - do:\n
projective_space_activity:\n
canonical_form_PG:\n
input:\n
-set_of_points"10,11,59,63,124,135,136,170,206,257,275,284,285,367,378,393,433,619,641,644"\n
-set_of_points"2,12,48,65,87,120,189,246,305,323,354,375,434,435,482,496,557,586,595"\n
-classification_prefix-quartic_25_0_0:\n
-report:\n
-end\n
d-pdflatex-quartic_25_0_0.classification.tex\n
open-quartic_25_0_0.classification.pdf\n
geo_16.c:\n
#define F - finite_field - q 10 - end\n
draw.incidence.structure.description:\n
-width 60 - with 10 - 6 - end\n
#define C - combinatorial_objects:\n
-file_of_incidence_geometries-geo_16.inc;16.20-80\n
-end\n
-with C - do:\n
-combinatorial_object_activity\n
797
**Section 15.4: Objects from Design Theory**

LS_AG_2_3_solutions_classify:

```
15937 ▶ ▶ ▶ -canonical_form\n15938 ▶ ▶ ▶ -classification_prefix.16\n15939 ▶ ▶ ▶ -label.16\n15940 ▶ ▶ ▶ -save\n15941 ▶ ▶ ▶ -save_transversal\n15942 ▶ ▶ -end\n15943 ▶ ▶ -report\n15944 ▶ ▶ ▶ -prefix.16\n15945 ▶ ▶ ▶ -export_flag.orbits\n15946 ▶ ▶ ▶ -show_incidence_matrices\n15947 ▶ ▶ ▶ -export_group\n15948 ▶ ▶ -end\n15949 ▶ -end
15950 ▶ pdflatex.16.classification.tex
15951 ▶ open.16.classification.pdf
15952 ▶ $(ORBITER\PATH)orbiter.out -v.2 -draw_matrix\n15953 ▶ -input_csv_file.16.object0_TDA_flag.orbits.csv\n15954 ▶ -secondary_input_csv_file.16.object0_TDA.csv\n15955 ▶ -box_width.16 -bit_depth.24\n15956 ▶ -end\n15957 ▶ $(ORBITER\PATH)orbiter.out -v.2 -draw_matrix\n15958 ▶ -input_csv_file.16.object0_INP_flag.orbits.csv\n15959 ▶ -secondary_input_csv_file.16.object0_INP.csv\n15960 ▶ -box_width.16 -bit_depth.24\n15961 ▶ -end
15962
15963
15964 # Section 15.4: Objects from Design Theory
15966
15967
15968 SECTION_OBJECTS_FROM_DESIGN_THEORY:
15969
15970
15971
15972 LS_AG_2_3_solutions_classify:
15973 ▶ $(ORBITER\PATH)orbiter.out -v.2\n15974 ▶ -draw_incidence_structure_description\n15975 ▶ ▶ -width.20 -width.10.2\n15976 ▶ ▶ -define.C -combinatorial_objects\n15977 ▶ ▶ -file.of.designs\n15978 ▶ ▶ -solutions.csv.9.84.3.12\n15979 ▶ ▶ -end\n15980 ▶ -with.C\n15981 ▶ -combinatorial_object_activity\n15982 ▶ ▶ -canonical_form\n15983 ▶ ▶ ▶ -save\n```
-save_transversal\n-classification_prefix:LS_AG.2.3\n-label:LS_AG.2.3\n-max_TDO.depth:10\n-end\n-report:\n-prefix:LS_AG.2.3\n-export_flag_orbits:\n-show_TDO:\n-end\n-prefix::LS_AG.2.3\n-export_flag_orbits:\n-show_TDO:\n-end

pdflatex:LS_AG.2.3_classification.tex
open:LS_AG.2.3_classification.pdf

$(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
-input_csv_file:LS_AG.2.3_object0_INP_flag_orbits.csv-
-secondary_input_csv_file:LS_AG.2.3_object0_INP.csv-
-box_width:12-bit_depth:24
-end

open:LS_AG.2.3_object0_INP_flag_orbits.draw.bmp

$(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
-input_csv_file:LS_AG.2.3_object1_INP_flag_orbits.csv-
-secondary_input_csv_file:LS_AG.2.3_object1_INP.csv-
-box_width:12-bit_depth:24
-end

open:LS_AG.2.3_object1_INP_flag_orbits.draw.bmp

design_27c:
$(ORBITER_PATH)orbiter.out-v.4-
define:C-combinatorial_objects-
-set_of_points:"2,56,30,112,253,90,440,508"
-end

#define:F-finite_field-q:27-override_polynomial:46-end

#define:P-projective_space:2-F-end

-with:C-do-

-combinatorial_object_activity-

-canonical_form:PG-P-
classification.prefix:design-
-end-

-report-

-end

pdflatex:design_classification.tex
open:design_classification.pdf
# Section 15.5: Linear Codes

SECTION CANONICAL FORMS OF LINEAR CODES:

code 3 2 aut:

\$(\text{ORBITER \text{PATH}})\text{orbiter.out-}v.20/ \\
\$\> -define\text{-finite_field-}q.2-\text{-end/} \\
\$\> -define-gemma-\text{-field-}F.-\text{-format-}2/ \\
\$\> \> -dense-$(\text{CODE}_3.K2.Q2.GENMA)/ \\
\$\> -end/ \\
\$\> -define-P.-projective_space-1.F.-\text{-end/} \\
\$\> \> -with-P.-do/ \\
\$\> \> -projective_space_activity/ \\
\$\> \> \> -canonical_form_of_code/ \\
\$\> \> \> \> "3_2".gemma.-save_ago.-label."3_2="/ \\
\$\> \> \> \> -classification_prefix."3_2="/ \\
\$\> \> \> \> -end/ \\
\$\> \> \> -end \\
\$\> pdf\text{-latex-3_2.classification.tex} \\
\$\> open-3_2.classification.pdf \\
\$\> $(\text{ORBITER \text{PATH}})\text{orbiter.out-}v.2.-\text{-draw_matrix/} \\
\$\> \> -input_csv_file-3_2.object0.TDA_flag_orbits.csv/ \\
\$\> \> -secondary_input_csv_file-3_2.object0.TDA.csv/ \\
\$\> \> -box_width-16.-bit_depth.24/ \\
\$\> \> -end \\
\$\> open-3_2.object0.TDA_flag_orbits_draw.bmp \\
\$\> open-3_2.object0.TDA_flag_orbits_draw.bmp \\

code 6 3 aut:

\$(\text{ORBITER \text{PATH}})\text{orbiter.out-}v.20/ \\
\$\> -define\text{-finite_field-}q.2-\text{-end/} \\
\$\> -define-gemma-\text{-field-}F.-\text{-format-}3/ \\
\$\> \> -compact-$(\text{CODE}_6.K3.Q2.GENMA)/ \\
\$\> \> -end/ \\
\$\> \> -define-P.-projective_space-2.F.-\text{-end/}
RM_3.1_group:
defined in a finite field F-q^2.
Genma vector field F-format 4.
Compact $(CODE_RM_3.1_GENMA)$.
End.
RM_3.1_proj_space:
with P-do.
RM_3.1_activity-
canonical_form_of_code-
"RM_3.1"-genma-save_ago-label."RM_3.1"-
canonical_form_of_genma-
"RM_3.1"-classification_prefix: "RM_3.1"-
End.

Group order 1344
# RM_3.1_object0_INP_flag_orbits.csv

RM_3.1_group_and_diagram:
define F: finite_field -q^2-end.
define genma: vector_field F-format 4.
Compact $(CODE_RM_3.1_GENMA)$.
End.
define P: projective_space 3-F-end.
with P-do.

Group of order 24
# group_of_order_24

Group and diagram:
define F: finite_field -q^2-end.
define genma: vector_field F-format 4.
Compact $(CODE_RM_3.1_GENMA)$.
End.
define P: projective_space 3-F-end.
with P-do.
-projective_space_activity\
-canonical_form_of_code\
"RM_3_1".genma-save Ago-label."RM_3_1".\n-classification_prefix."RM_3_1".\n-end\n-pdflatex RM_3_1_classification.tex
-open RM_3_1_classification.pdf
$(ORBITER_PATH)orbiter.out-v.2-draw matrix-
-input_csv_file-RM_3_1_object0_INP_flag_orbits.csv-
-secondary_input_csv_file-RM_3_1_object0_INP.csv-
-box_width:16-bit_depth:24-
-end\n-open RM_3_1_object0_INP_flag_orbits.bmp
-open RM_3_1_object0_TDA_flag_orbits.bmp

RM_4_1_group:
$(ORBITER_PATH)orbiter.out-v.2-
-defineF-finite_field-q.2-end-
-define-genma-vector-fieldF-format.5-
-compact $(CODE_RM_4_1_GENMA)-end-
-defineP-projective_space4F-end-
-withP-do-
-projective_space_activity-
-classification_prefix."RM_4_1".\n-end\n-pdflatex RM_4_1_classification.tex
-open RM_4_1_classification.pdf
$(ORBITER_PATH)orbiter.out-v.2-draw matrix-
-input_csv_file-RM_4_1_object0_INP_flag_orbits.csv-
-secondary_input_csv_file-RM_4_1_object0_INP.csv-
-box_width:16-bit_depth:24-
-end\n-open RM_4_1_object0_TDA_flag_orbits.bmp
-open RM_4_1_object0_TDA_flag_orbits.bmp
\begin{verbatim}
16172 \texttt{\textasciicircum -box_width}16\texttt{\textasciicircum -bit_depth}24\texttt{\textasciicircum -end}\
16173 \texttt{\textasciicircum -open-RM.4.1_object0_INP_flag_orbits_draw.bmp}\
16174 \texttt{\textasciicircum -open-RM.4.1_object0_TDA_flag_orbits_draw.bmp}\
16175 \texttt{\textasciicircum -open-RM.4.1_object0_INP_flag_orbits_draw.bmp}\
16176 16177 16178 \# \texttt{\textasciicircum -group-order}322560=24\texttt{\textasciicircum *30*28*16}\
16179 16180 16181 RS_6.4.7_group: 16182 \texttt{\textasciicircum -$\{$ORBITER\_PATH\}orbiter.out\textasciicircum -v.20\texttt{\textasciicircum -}}\
16183 \texttt{\textasciicircum -define-F\textasciicircum -finite_field\textasciicircum -q7\texttt{\textasciicircum -end}}\
16184 \texttt{\textasciicircum -define-genma\textasciicircum -vector\textasciicircum -field\textasciicircum -F\textasciicircum -format}4\texttt{\textasciicircum -}\
16185 \texttt{\textasciicircum -compact-$\{$CODE_RS_6.4.7\}$}}\
16186 \texttt{\textasciicircum -end}}\
16187 \texttt{\textasciicircum -define-P\textasciicircum -projective_space\textasciicircum -3\textasciicircum -F\textasciicircum -end}}\
16188 \texttt{\textasciicircum -with-P\textasciicircum -do}}\
16189 \texttt{\textasciicircum -projective_space_activity}}\
16190 \texttt{\textasciicircum -canonical_form_of_code}}\
16191 \texttt{\textasciicircum -"RS_6"\textasciicircum -genma\textasciicircum -save_ago\textasciicircum -label\textasciicircum -"RS_6"}}\
16192 \texttt{\textasciicircum -classification_prefix\textasciicircum -"RS_6"}}\
16193 \texttt{\textasciicircum -end}}\
16194 \texttt{\textasciicircum -end}}\
16195 16196 16197 GV_n15_k6_d5_group: 16198 \texttt{\textasciicircum -$\{$ORBITER\_PATH\}orbiter.out\textasciicircum -v.20\texttt{\textasciicircum -}}\
16199 \texttt{\textasciicircum -define-F\textasciicircum -finite_field\textasciicircum -q2\texttt{\textasciicircum -end}}\
16200 \texttt{\textasciicircum -define-genma\textasciicircum -vector\textasciicircum -field\textasciicircum -F\textasciicircum -format}6\texttt{\textasciicircum -}\
16201 \texttt{\textasciicircum -compact-$\{$CODE_GV_N15_K6\}$}}\
16202 \texttt{\textasciicircum -end}}\
16203 \texttt{\textasciicircum -define-P\textasciicircum -projective_space\textasciicircum -5\textasciicircum -F\textasciicircum -end}}\
16204 \texttt{\textasciicircum -with-P\textasciicircum -do}}\
16205 \texttt{\textasciicircum -projective_space_activity}}\
16206 \texttt{\textasciicircum -canonical_form_of_code}}\
16207 \texttt{\textasciicircum -"GV_n15_k6_d5"\textasciicircum -genma\textasciicircum -save_ago\textasciicircum -label\textasciicircum -"GV_n15_k6_d5"}}\
16208 \texttt{\textasciicircum -classification_prefix\textasciicircum -"GV_n15_k6_d5"}}\
16209 \texttt{\textasciicircum -end}}\
16210 \texttt{\textasciicircum -end}}\
16211 \texttt{pdflatex\textasciicircum -GV_n15_k6_d5\_classification.tex}\
16212 \texttt{open\textasciicircum -GV_n15_k6_d5\_classification.pdf}\
16213 16214 \# ago=12}\
16215 16216 16217 16218 code_n15_k6_d6_a_group:
\end{verbatim}
Hamming_graph_7_with_Hamming_code:
-define G -graph -Hamming:7.2 \\
-define \".\" -subset "Hamming\_code"."\\_with\"Hamming\"\_code".\" \\
$(HAMMING\_CODE\_CODEWORDS) -end \\
 -with G -do \\
 -graph\_theoretic\_activity -export\_csv -end \\
 -with G -do \\
 -graph\_theoretic\_activity -export\_graphviz -end \\
 -with G -do \\
 -graph\_theoretic\_activity -save -end \\
 -with G -do \\
 -graph\_theoretic\_activity -automorphism\_group -end \\
 pdflatex Hamming:7.2 Hamming\_code\_report.tex \\
 open Hamming:7.2 Hamming\_code\_report.pdf

# group of order 2688 = 16 * 168

# Section 15.7: Graphs

SECTION CANONICAL FORMS OF GRAPHS:

Cycle\_13\_aut: \\
$\$(ORBITER\_PATH)orbiner\_out -v\:2 \\
 -define Gamma -graph -cycle:13 -end \\\n -with Gamma -do \\
 -graph\_theoretic\_activity -automorphism\_group -end \\
 -end

Chain\_232\_aut: \\
$\$(ORBITER\_PATH)orbiner\_out -v\:2 \\
 -define P1 -vector -dense 2,3,2 -end \\
 -define P2 -vector -dense 2,3,2 -end \\
 -define Gamma -graph \\
 -chain_graph P1\_P2 \\
 -end \\
 -with Gamma -do \\
 -graph\_theoretic\_activity -automorphism\_group -end \\
 -end

dflatex chain\_graph\_report.tex
open-chain_graph_report.pdf

JK_graph_pp16_1:
$(ORBITER_PATH)orbiter.out-v.2\n-define-Gamma-graph-load_dimacs\n../JUNTTILA_KASKI/benchmarks/pp/pp16-1\n-end\n-with-Gamma-do\n-graph_theoretic_activity-save-end\n-with-Gamma-do\n-graph_theoretic_activity-automorphism_group-end\n
#go=34217164800

nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack1=6
nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack2=134

JK_graph_pp16_2:
$(ORBITER_PATH)orbiter.out-v.2\n-define-Gamma-graph-load_dimacs\n../JUNTTILA_KASKI/benchmarks/pp/pp16-2\n-end\n-with-Gamma-do\n-graph_theoretic_activity-save-end\n-with-Gamma-do\n-graph_theoretic_activity-automorphism_group-end\n
#does-not-finish

JK_graph_pp16_9:
$(ORBITER_PATH)orbiter.out-v.2\n-define-Gamma-graph-load_dimacs\n../JUNTTILA_KASKI/benchmarks/pp/pp16-9\n-end\n-with-Gamma-do\n-graph_theoretic_activity-save-end\n-with-Gamma-do\n-graph_theoretic_activity-automorphism_group-end\n
JK_graph_grid_3_3:
$(ORBITER_PATH)orbiter.out-v.2\n-define-Gamma-graph-load_dimacs\n../JUNTTILA_KASKI/benchmarks/grid/grid-w-3-3\n
806
16358 ▷▶ -end\n16359 ▷▶ -with-Gamma-do\n16360 ▷▶ -graph_theoretic_activity-save-end\n16361 ▷▶ -with-Gamma-do\n16362 ▷▶ -graph_theoretic_activity-automorphism_group-end\n16363
16364
16365 #nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack1=4
16366 #nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack2=9
16367 #Written file grid-w-3-3_group.makefile of size 579
16368 #User time: 0 of a second, dt=0 tps=100
16369 #nb.calls_to_densenausty=1
16370
16371
16372 JK_graph_sts_13:
16373 ▷ $(ORBITER_PATH)orbiter.out-v.2\n16374 ▷ ▷ -define-Gamma-graph-load_dimacs\n16375 ▷ ▷ ▷ ../JUNTTILA_KASKI/benchmarks/srg/sts-13\n16376 ▷ ▷ -end\n16377 ▷ ▷ -with-Gamma-do\n16378 ▷ ▷ -graph_theoretic_activity-save-end\n16379 ▷ ▷ -with-Gamma-do\n16380 ▷ ▷ -graph_theoretic_activity-automorphism_group-end
16381 ▷ make ORBITER_PATH=$(ORBITER_PATH)-f sts-13_group.makefile sts-13
16382
16383
16384 #nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack1=3
16385 #nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack2=24
16386
16387
16388 HJ_aut:
16389 ▷ $(ORBITER_PATH)orbiter.out-v.6\n16390 ▷ ▷ -define-G-graph\n16391 ▷ ▷ ▷ -load_csv_no_border\n16392 ▷ ▷ ▷ halljanko315.csv\n16393 ▷ ▷ -end\n16394 ▷ ▷ -with-G-do\n16395 ▷ ▷ ▷ -graph_theoretic_activity-automorphism_group\n16396 ▷ ▷ -end\n16397 ▷ ▷ -with-G-do\n16398 ▷ ▷ ▷ -graph_theoretic_activity-properties\n16399 ▷ ▷ -end
16400

807
HJ_group_and_orbits:

```bash
# $(ORBITER_PATH) orbiter.out -v.2
define-gens -vector-file
halljanko315.gens.csv -end
define-G -permutation_group
bsgs-halljanko315 "File\halljanko315".v2
315 -1209600 0,1,2 -6 gens
-end

-with-G -do

define-Gamma -graph
-orbital_graph_G.3
-end

-with-Gamma -do
-graph_theoretic_activity
-properties
-end

-with-Gamma -do
-graph_theoretic_activity
-save
-end
```

HJ_orbital_graph.3:

```bash
#ROW,REP,AGO,OL
0 0,1 96 12600
1 0,2 48 25200
2 0,4 768 15750
3 0,8 120 10080
#END
```

HJ_orbital_graph.3:
#Group_Perms15_Orbital_3.colored_graph

#Degree-type: (64^-{315}.)

PGO_5.2_graph_group:0.5.2_incidence_matrix.csv

$(ORBITER\ PATH) orbiter.out -v.3$

-define-Inc-vector-file:0.5.2_incidence_matrix.csv -end$

-define-Gamma-graph-collinearity_graph:Inc -end$

- with Gamma - do$

- graph_theoretic_activity$

- automorphism_group$

- end$

- with Gamma - do$

- graph_theoretic_activity$

- eigenvalues$

- end$

pdflatex collinearity_graph_eigenvalues.tex

open-collinearity_graph_eigenvalues.pdf

#Section 15.8: Quartic Curves

SECTION_CANONICAL_FORMS_OF_QUARTIC_CURVES:

F_17_edge:

$(ORBITER\ PATH) orbiter.out -v.3$

-define-F-finite_field-q17 -end$

- with F - do - finite_field_activity$

- cheat_sheet_GF - end

pdflatex GF_17.tex

open GF_17.pdf
16501 Edge_curve_17_nauty:\n16502 ▷ $(ORBITER_PATH)orbiter.out-v.3:\n16503 ▷ ▷ -define C-combinatorial_objects:\n16504 ▷ ▷ ▷ -file_of_points Edge_q17.txt:\n16505 ▷ ▷ -end:\n16506 ▷ ▷ -define F-finite_field=q.17-end:\n16507 ▷ ▷ -define P-projective_space=2 F-end:\n16508 ▷ ▷ -with C-do:\n16509 ▷ ▷ -combinatorial_object_activity:\n16510 ▷ ▷ ▷ -canonical_form PG P:\n16511 ▷ ▷ ▷ ▷ -classification_prefix Edge_curve_q17:\n16512 ▷ ▷ ▷ ▷ -label Edge_curve_q17:\n16513 ▷ ▷ ▷ ▷ -save ago:\n16514 ▷ ▷ ▷ ▷ -save_transversal:\n16515 ▷ ▷ ▷ ▷ -max_TDO_depth=10:\n16516 ▷ ▷ ▷ ▷ -end:\n16517 ▷ ▷ ▷ -report:\n16518 ▷ ▷ ▷ ▷ -prefix Edge_curve_q17:\n16519 ▷ ▷ ▷ ▷ -export_flag_orbits:\n16520 ▷ ▷ ▷ ▷ -show_TDO:\n16521 ▷ ▷ ▷ ▷ -show TDA:\n16522 ▷ ▷ ▷ ▷ -dont_show_incidence_matrices:\n16523 ▷ ▷ ▷ ▷ -export_group:\n16524 ▷ ▷ ▷ -end:\n16525 ▷ ▷ -end
16526 ▷ pdflatex Edge_curve_q17_classification.tex
16527 ▷ open Edge_curve_q17_classification.pdf
16528 ▷ $(ORBITER_PATH) orbiter.out-v.2-draw_matrix:\n16529 ▷ ▷ -input_csv_file Edge_curve_q17 object0 TDA_flag_orbits.csv:\n16530 ▷ ▷ -secondary_input_csv_file Edge_curve_q17 object0 TDA.csv:\n16531 ▷ ▷ -box_width=4-bit_depth=24:\n16532 ▷ ▷ -end
16533 ▷ open Edge_curve_q17 object0 TDA_flag_orbits_draw.bmp
16534
16535 # 9 backtrack nodes total
16536
16537
16538 # aut=24
16539 # User.time: 0.04 of a second, dt=4.tps=.100
16540
16541

810
# generators for a group of order 24:
1,0,0,0,13,0,0,0,4,
1,0,0,0,16,0,16,0,
0,1,16,2,4,4,15,4,4,

Edge_curve_17_group:
$(ORBITER\ PATH) orbiter.out \-v \-3 \$

define G \-linear_group \-PGL\-3\-17 \$

subgroup_by_generators \-Stab\_Edge""\-24\-3 \$

"1,0,0,13,0,0,0,4"\$

"1,0,0,0,16,0,16,0"\$

"0,1,16,2,4,4,15,4,4"\$

end \$

with G \-do \$

- group_theoretic_activities \$

- print_elements.tex \$

- group_table \$

- report \$

- end \$

pdf latex PGL\_3\_17\_Subgroup\_Stab\_Edge\_24\_report.tex
open PGL\_3\_17\_Subgroup\_Stab\_Edge\_24\_report.pdf

F_7_tables:
$(ORBITER\ PATH) orbiter.out \-v \-3 \$

define F \-finite_field \-q\-7 \-end \$

with F \-do \-finite_field_activity \$

cheat sheet GF \$

end \$

$(ORBITER\ PATH) orbiter.out \-v \-2 \$

draw matrix \$

- input_csv_file GF_q7_addition_table.csv \$

SECTION_GRAPHICAL_OUTPUT:

16589 ▶ ▶ ▶ -box_width:40\  
16590 ▶ ▶ ▶ -bit_depth:24\  
16591 ▶ ▶ ▶ -partition:3:7:7\  
16592 ▶ ▶ -end  
16593 ▶ open-GF_q7_addition_table_draw.bmp  
16594  
16595  
16596  
16597 PG_2.4_cyclic_incma:  
16598 ▶ $(ORBITER_PATH)orbiter.out-\v:2\  
16599 ▶ ▶ -define-F\-finite_field\-q:4\-end\  
16600 ▶ ▶ -define-P\-projective_space\-2\-F\-end\  
16601 ▶ ▶ -with-P\-do\-projective_space_activity\  
16602 ▶ ▶ ▶ -cheat_sheet\for\decomposition\by\element\PG\  
16603 ▶ ▶ ▶ 1:"0,1,0,-0,1,2,1,1,0"."PG_2.4_singer".\  
16604 ▶ ▶ -end  
16605 ▶ $(ORBITER_PATH)orbiter.out-\v:4\  
16606 ▶ ▶ -list_arguments\  
16607 ▶ ▶ -define-R\-vector\-repeat\-1\-21\-end\  
16608 ▶ ▶ -define-C\-vector\-repeat\-1\-21\-end\  
16609 ▶ ▶ -draw_matrix\  
16610 ▶ ▶ -input_csv_file\PG_2.4_singer_incma_cyclic.csv\  
16611 ▶ ▶ -box_width:40\-bit_depth:24\  
16612 ▶ ▶ -partition:3:R:C\  
16613 ▶ ▶ -end  
16614 ▶ open-PG_2.4_singer_incma_cyclic_draw.bmp  
16615  
16616  
16617  
16618  
16619 PGL_4.2_Wedge_4.0_graphical_output:  
16620 ▶ $(ORBITER_PATH)orbiter.out-\v:4\  
16621 ▶ ▶ -define-G\-linear_group\-PGL:4:2\  
16622 ▶ ▶ ▶ -wedge_detached\  
16623 ▶ ▶ -end\  
16624 ▶ ▶ -with-G\-do\  
16625 ▶ ▶ -group_theoretic_activity\  
16626 ▶ ▶ ▶ -report\  
16627 ▶ ▶ -end  
16628 ▶ pdflatex:\PGL_4.2_Wedge_4.2_detached\report.tex  
16629 ▶ open-PGL_4.2_Wedge_4.2_detached\report.pdf  
16630  
16631 #▶ ▶ ▶ -draw_options\-radius:200\-end\  
16632  
16633 schreier_tree_graphical_output:  
16634 ▶ $(ORBITER_PATH)orbiter.out-\v:4\  
16635 ▶ ▶ -draw_options\  

812
Queens_graph:

```
$\text{(ORBITER\_PATH)}\text{orbiter.out-}\text{-v.2}\$

-define G -graph -non_attacking_queens_graph -end

-width G -do

-graph_theoretic_activity -export_csv -end

-width G -do

-graph_theoretic_activity -export_graphviz -end

-width G -do

-graph_theoretic_activity -save -end

-width G -do

-graph_theoretic_activity -automorphism_group -end

-width G -do

-graph_theoretic_activity -find_cliques

-target_size 8 -output_file 8queens -end

-end
```

cube:

```
$\text{(ORBITER\_PATH)}\text{orbiter.out-}\text{-v.2-povray}\$

-round 0 -nb_frames default 30

-output_mask cube_%d_%03d.pov

-video_options -W 1024 -H 768

-global_picture_scale 0.5

-default_angle 75
```
-clipping_radius=2.7\ 
-end\ 
-scene_objects\ 
-obj_file-cube_centered.obj\ 
-edge"0,1"\ 
-edge"0,2"\ 
-edge"0,4"\ 
-edge"1,3"\ 
-edge"1,5"\ 
-edge"2,3"\ 
-edge"2,6"\ 
-edge"3,7"\ 
-edge"4,5"\ 
-edge"4,6"\ 
-edge"5,7"\ 
-edge"6,7"\ 
-group_of_things_as_interval=0.8\ 
spheres=0.0.3\ 
"texture{Polished_Chrome.pigment{quick_color=White}}".\ 
-group_of_things_as_interval=0.6\ 
-prisms=1.0.05\ 
"texture{pigment{color=Yellow.transmit=0.7}}".\ 
finish{diffuse=0.9.phong=0.6}.\ 
-group_of_things_as_interval=0.12\ 
cylinders=2.0.15\ 
"texture{pigment{color=Red}.finish{diffuse=0.9.phong=0.6}}".\ 
-scene_objects_end\ 
povray_end\ 
rm-POV\ 
kdir-POV\ 
mv-cube_0_*.pov-POV\ 
mv-makefile_animation-POV\ 

monkey:\ 
$(ORBITER_PATH)orbiter.out-v.2-povray\ 
-round=0-nb_frames_default=30\ 
-output_mask-monkey_\%d_\%03d.pov\ 
-video_options=W=1024-H=768\ 
global_picture_scale=0.8\ 
default_angle=75\ 
-clipping_radius=0.8\ 
camera=0,0,1,1,0.5,0,0,0\ 
-rotate_about_z_axis\ 
-end\ 

-scene_objects:\n-cubic_lex:$\langle MONKEY\_SADDLE\_CUBIC\rangle$-plane_by_dual_coordinates:"0,0,1,0"-group_of_things:"0"-group_of_things:"0"
cubics-0:"texture{pigment{Gold}.finish}-ambient 0.4.diffuse 0.5.roughness 0.001.reflection 0.1.specular .8}-planes-1:"texture{pigment{color Blue}.finish}transmit 0.5}.finish{diffuse 0.9.roughness 2 phong 0.2}-scene_objects_end-povray_end-rm rf POV mkdir POV mv monkey_0.*.pov POV mv makefile_animation POV

Eckardt:
$(ORBITER\_PATH)orbiter.out-\_v.2.povray-round 0-nb_frames_default 30-output_mask Eckardt_%d%03d.pov-video_options -W 1024 -H 768-global_picture_scale 0.9-default_angle 75-clipping_radius 2.4-camera 0."1,1,1".-3,1,3."0.12,0.12,0.12-end-scene_objects-Hilbert Cohn Vossen_surface-group_of_things:"0"-cubics-0:"texture{pigment{White*0.5.transmit 0.5}finish{ambient 0.4.diffuse 0.5.roughness 0.001.specular 0.1}-group_of_things_as_interval 0.6-group_of_things_as_interval 6.6-group_of_things_as_interval_with_exceptions 12.15-14,19,23-lines 1.0.02:"texture{pigment{color Red}.finish}diffuse 0.9.roughness 1 phong 1-lines 1.0.02:"texture{pigment{color Blue}.finish}diffuse 0.9.roughness 1 phong 1-lines 3.0.02:"texture{pigment{color Yellow}.finish}diffuse 0.9.roughness 1 phong 1
-label-0."a1".
-label-2."a2".
-label-4."a3".
-label-6."a4".
-label-8."a5".
-label-10."a6".
-label-12."b1".
-label-14."b2".
-label-16."b3".
-label-18."b4".
-label-20."b5".
-label-22."b6".
-label-24."c12".
-label-26."c13".
-label-30."c15".
-label-32."c16".
-label-34."c23".
-label-36."c24".
-label-40."c26".
-label-42."c34".
-label-44."c35".
-label-48."c45".
-label-50."c46".
-label-52."c56".
-group_of_things_as_interval-0.6.
-texts-4.0.2.0.15."texture{pigment{Black}.} no_shadow".
-group_of_things_as_interval-6.6.
-texts-5.0.2.0.15."texture{pigment{Black}.} no_shadow".
-texts-6.0.2.0.15."texture{pigment{Black}.} no_shadow".
-scene_objects_end.
povray_end
-rm-rf-POV
mkdir-POV
mv-Eckardt_0_*.pov-POV
mv-makefile_animation-POV
mv-makefile

M:=Matrix([[−4.5, 3.5, 6],[1, 1, 1]])
NullSpace(M)

W:=-W1024—H:768
W:=-W2560—H:1920
W:=-W4096—H:3072
Eckardt_deform:

```bash
$(ORBITER_PATH)orbiter.out -v.2 -povray -round.0 -nb_frames_default.93 -output_mask-Eckardt_deform_%d_%03d.pov -video_options:-W1024 -H768 -global_picture_scale.9 -default_angle.75 -clipping_radius.2.4 -camera.0."1,1,1"."-3,1,3"."0.12,0.12,0.12" -end

```

```bash
scene_objects
```

```bash
-Hilbert_Cohn_Vossen_surface -group_of_things.0 -deformation_of_cubic_lex.93.1.107148718.1.570796327.0 -$(ECKARDT_CUBIC_DEFORM1_LEX) -$(ECKARDT_CUBIC_DEFORM2_LEX) -group_of_things_as_interval.0.93 -group_is_animated.1 -cubics.1."texture{pigment{White*0.5.transmit.0.5}}" -ambient.0.4.diffuse.0.5.roughness.0.001.reflection.0.1.specular.8 -scene_objects_end -povray_end -rm .rf POV mkdir POV mv Eckardt_deform.0.*.pov POV mv .makefile.animation.POV

Eckardt_deform_2:

```bash
$(ORBITER_PATH)orbiter.out -v.2 -povray -round.0 -nb_frames_default.30 -output_mask-Eckardt_deform_%d_%03d.pov -video_options:-W1024 -H768 -global_picture_scale.9 -default_angle.75 -clipping_radius.2.4 -camera.0."1,1,1"."-3,1,3"."0.12,0.12,0.12" -end

```

```bash
scene_objects
```

```bash
-Hilbert_Cohn_Vossen_surface
```
-group_of_things:"0">
\$ECKARDT_CUBIC_DEFORM1_LEX>
\$ECKARDT_CUBIC_DEFORM2_LEX>
--group_of_things_as_interval:0.93
--group_is_animated:1
-group_of_things:"0">
-cubics:1:"texture{pigment{White*0.5.transmit:0.5}:}
finish{ambient:0.4.diffuse:0.5.roughness:0.001.reflection:0.1.specular:.8}"
-group_of_things:"24">
-cubics:2:"texture{pigment{Red*0.5.transmit:0.5}:}
finish{ambient:0.4.diffuse:0.5.roughness:0.001.reflection:0.1.specular:.8}"
-cubics:3:"texture{pigment{Blue*0.5.transmit:0.5}:}
finish{ambient:0.4.diffuse:0.5.roughness:0.001.reflection:0.1.specular:.8}"
-scene_objects_end
-povray_end
-rf-POV
mkdir-POV
mv-Eckardt_deform_0_*.pov-POV
mv-makefile_animation-POV
 Clebsch:
$ORBITER_PATH/Orbiter.out-v.2-povray
-round:0:-nb_frames:default:30
-output_mask:Clebsch_%d_%03d.pov
-video_options:-W:1024:-H:768
-global_picture_scale:0.9
-default_angle:80
-clipping_radius:2.4
-camera:0:"1,1,1"."-4.5,3.5,6"."0,0,0".
-end
-scene_objects
-Clebsch_surface
-group_of_things:"0">
cubics:0:"texture{pigment{White*0.5}:}finish
{ambient:0.4.diffuse:0.5.roughness:0.001.reflection:0.1.specular:.8}"
-group_of_things_as_interval:0.6
-group_of_things_as_interval:6.6
-group_of_things_as_interval:12.15
-lines:1.0.02:"texture{pigment{color:Red}:}
16917 finish{{diffuse 0.9-phong 1}}"\n16918  >  >  > -lines 2.0.02"texture{pigment{color Blue}}\n16919 finish{{diffuse 0.9-phong 1}}"\n16920  >  >  > -lines 3.0.02"texture{pigment{color Yellow}}\n16921 finish{{diffuse 0.9-phong 1}}"\n16922  >  >  > -group of things as interval 0.12\n16923  >  >  > -spheres 4.0.08"texture{pigment{Cyan 1.3}}\n16924 finish{ambient 0.4-diffuse 0.6-roughness 0.001\n16925 reflection 0-specular .8}"\n16926  >  > -scene_objects_end\n16927  >  > -povray_end\n16928  >  > -rm -rf POV\n16929  >  mkdir POV\n16930  >  mv Clebsch 0*.pov POV\n16931  >  mv makefile_animation POV\n16932\n16933\n16934\n16935 endrass8:\n16936  > $(ORBITER_PATH) orbiter.out -v 2 -povray\n16937  >  > -round 0 -nb_frames_default 30\n16938  >  > -output_mask endrass octic_%d %d pov\n16939  >  > -video options -W 1024 -H 768\n16940  >  > -global_picture_scale 0.75\n16941  >  > -default_angle 75\n16942  >  > -clipping_radius 3.7\n16943  >  > -no_bottom_plane\n16944  >  > -camera 0 "1, 1, 1" "6, 6, 3" "0, 0, 0"\n16945  >  > -rotate about 111\n16946  >  > -end\n16947  >  > -scene_objects\n16948  >  >  > -line through two points recentered from csv file\n16949  >  >  >  > coordinate_grid.csv\n16950  >  >  >  > -group of things "0"\n16951  >  >  >  > -group of things "1"\n16952  >  >  >  > -group of things "2"\n16953  >  >  >  > -group of things as interval 3.39\n16954  >  >  >  > -lines 0.0.15 "texture{pigment {color Red}}\n16955 finish{{diffuse 0.9-phong 1}}"\n16956  >  >  > -lines 1.0.15 "texture{pigment {color Green}}\n16957 finish{{diffuse 0.9-phong 1}}"\n16958  >  >  > -lines 2.0.15 "texture{pigment {color Blue}}\n16959 finish{{diffuse 0.9-phong 1}}"\n16960  >  >  > -lines 3.0.05 "texture{pigment {color Black}}\n16961 finish{{diffuse 0.9-phong 1}}"\n16962  >  >  > -octic_lex_165 $(ENDRASS OCTIC LEX 165)\n16963  >  >  > -plane by dual coordinates "0, 0, 1, 0"
GROUP OF THINGS

- octics 4
  "texture{pigment{White*0.5 transmit 0.5}}"
- planes 5
  "texture{pigment{color Blue transmit 0.5}}"

finish {ambient 0.4 diffuse 0.5 roughness 0.001
reflection 0.1 specular .8}"

- scene_objects_end

- povray_end

- monkey_video
  - rm -r FRAMES
  - mkdir FRAMES
  - rm monkey.mp4
  - $(ORBITER PATH)orbiter.out
  - prepare_frames
  - i 0 30 monkey 0 \%03d.png
  - output_starts_at 0
  - o FRAMES/FRAME\%04d.png
  - end
  - ffmpeg -r 5 -f image2 -i FRAMES/FRAME\%04d.png
  - "mp4 -q:v 0 -vcodec mpeg4 monkey.mp4"

- Eckardt_deform_video
  - rm -r FRAMES
  - mkdir FRAMES
  - rm Eckardt_deform.mp4
  - $(ORBITER PATH)orbiter.out
  - prepare_frames
  - i 0 93 Eckardt_deform 0 Eckardt_deform 0 \%03d.png
  - output_starts_at 0
  - o FRAMES/FRAME\%04d.png
  - end
ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png
-f mp4 -q:v 0 -vcodec mpeg4 Eckardt_deform.mp4

Eckardt_surface:
$(ORBITER_PATH)orbiter.out -v 2 -povray
-round 0 -nb_frames default 30
-output_mask Eckardt_%d%03d.pov
-video_options -W 1024 -H 768
-global_picture_scale 0.9
-default_angle 75
-clipping_radius 2.4
-camera 0 "1,1,1" "-3,1,3" "0.12,0.12,0.12"
-end
-scene_objects
-cubic Goursat 6,3,15
-group_of_things 0
-cubics 0
-texture pigment
White 0.5 transmit 0.5 finish ambient 0.4
diffuse 0.5 roughness 0.001 reflection 0.1
specular 0.8
-scene_objects_end
-povray_end
-rm -rf POV
mkdir POV
mv Eckardt_0*.pov POV
mv makefile_animation POV

Kummer_surface:
$(ORBITER_PATH)orbiter.out -v 2 -povray
-round 0 -nb_frames default 30
-output_mask Kummer_%d%03d.pov
-video_options -W 1024 -H 768
-global_picture_scale 0.9
-default_angle 75
-clipping_radius 2.4
-camera 0 "1,1,1" "-3,1,3" "0.12,0.12,0.12"
-end
-scene_objects
-quartic lex 35 $(KUMMER_QUARTIC_LEX 35)
-group_of_things 0
-quartics 0
-texture pigment
White 0.5 transmit 0.5 finish ambient 0.4
diffuse 0.5 roughness 0.001 reflection 0.1
specular 0.8
-scene_objects_end

17058 ▶ ▶ -povray_end
17059 ▶ -rm:-rf-POV
17060 ▶ mkdir-POV
17061 ▶ mv-Kummer_0_*-pov-POV
17062 ▶ mv-makefile_animation-POV
17063
17064
17065 # Maple:
17066 # Kummer := expand((x0^2+x1^2+x2^2+x3^2)^2-3*(x0^4+x1^4+x2^4+x3^4))
17067
17068
17069 Kummer_video:
17070 ▶ -rm:-r-FRAMES
17071 ▶ -mdir-FRAMES
17072 ▶ -rm-Kummer.mp4
17073 ▶ $(ORBITER_PATH)orbiter.out \ 
17074 ▶ ▶ -prepare_frames: \ 
17075 ▶ ▶ ▶ -i:0-30-KUMMER/Kummer_0_%03d.png \ 
17076 ▶ ▶ ▶ -output_starts_at:0- \ 
17077 ▶ ▶ ▶ -o-FRAMES/frame%04d.png \ 
17078 ▶ ▶ -end \ 
17079 ▶ ffmpeg -r:5-f-image2 -i-FRAMES/frame%04d.png \ 
17080 ▶ ▶ -f:mp4:-q:0:-vcodec:mpeg4-Kummer.mp4 \ 
17081
17082
17083
17084 Beauville_surface:
17085 ▶ $(ORBITER_PATH)orbiter.out -v:2 -povray: \ 
17086 ▶ ▶ -round:0:-nb_frames_default:30: \ 
17087 ▶ ▶ -output_mask-Beauville_%d_%03d.pov \ 
17088 ▶ ▶ -video_options:-W:1024:-H:768: \ 
17089 ▶ ▶ -global_picture_scale:0.3: \ 
17090 ▶ ▶ -default_angle:75: \ 
17091 ▶ ▶ -clipping_radius:2.4: \ 
17092 ▶ ▶ -camera:0:"1,1,1":"-3,1,3":"0.12,0.12,0.12": \ 
17093 ▶ ▶ -end: \ 
17094 ▶ ▶ -scene_objects: \ 
17095 ▶ ▶ ▶ -quintic_lex:56:$(BEAUVILLE_QUINTIC_LX_56): \ 
17096 ▶ ▶ ▶ -group_of_things:"0": \ 
17097 ▶ ▶ ▶ -quintics:0:"texture{pigment\{White*0.5\transmit:0.5\}: \ 
17098 finish:{ambient:0.4\diffuse:0.5\roughness:0.001:\ 
17099 reflection:0.1\specular:.8\}": \ 
17100 ▶ ▶ -scene_objects_end: \ 
17101 ▶ -povray_end
17102 ▶ -rm:-rf-POV
17103 ▶ mkdir-POV
17104 ▶ mv-Beauville_0_*-pov-POV

822
mv-makefile_animation-POV

# Clebsch-map-up-for-surface-created-using-arc-lifting
# We take a circle of radius r centered at the origin in the affine real plane.
# and map it up on the surface.
# The Clebsch surface has:
# a = d = \(\frac{3+\sqrt{5}}{2}\)
# b = c = \(\frac{1+\sqrt{5}}{2}\)

CLEBSCH_A=2.618033988
CLEBSCH_D=2.618033988
CLEBSCH_B=1.618033988
CLEBSCH_C=1.618033988
TWO_PI=6.283185308

# to go from the arclifting surface to the defining equation:
# Matrix(4,-4,[-0.44721360215312733,1.1708204000530853,-1.1708204000530853,-0.4472135957999158],[-1.1708204000530853,0.4472136021531272,1.4472136021531272,-0.4472135957999158],[4.2360680044124255,-4.2360680044124255,-4.2360680044124255,-4.2360680044124255],[1.6180340022062127,-2.6180340022062127,-1.6180340022062127,0.])

# to go from the arclifting surface to the defining equation:
# Matrix(4,-4,[-0.44721360215312733,1.1708204000530853,-1.1708204000530853,-0.4472135957999158],[-1.1708204000530853,0.4472136021531272,1.4472136021531272,-0.4472135957999158],[4.2360680044124255,-4.2360680044124255,-4.2360680044124255,-4.2360680044124255],[1.6180340022062127,-2.6180340022062127,-1.6180340022062127,0.])

# T00=-0.44721360215312733
# T01=1.1708204000530853
# T02=1.1708204000530853
# T03=-0.4472135957999158
# T10=-1.1708204000530853
# T11=0.4472136021531272
# T12=1.4472136021531272
# T13=0.4472135957999158
# T20=4.2360680044124255
# T21=-4.2360680044124255
# T22=-4.2360680044124255
# T23=0.
# T30=1.6180340022062127
17148 T31=-2.6180340022062127
17149 T32=-1.6180340022062127
17150 T33=0.
17151
17152
17153 CLEBSCH_CUBICS=

17154 ▶ ▶ ▶ ▶ push b push b mul push d push c push m mul add mul \n17155 ▶ ▶ ▶ ▶ push b push c push d push d push m mul add mul add \n17156 ▶ ▶ ▶ ▶ push a push d push d push m add mul mul add add \n17157 ▶ ▶ ▶ ▶ push a push c push m mul add mul \n17158 ▶ ▶ ▶ ▶ store c001 \n17159 ▶ ▶ ▶ ▶ push b push d mul \n17160 ▶ ▶ ▶ ▶ push b push 1 push m push c mul add mul \n17161 ▶ ▶ ▶ ▶ push d push a push 1 push m mul add mul add \n17162 ▶ ▶ ▶ ▶ push m push a mul add push c add \n17163 ▶ ▶ ▶ ▶ push c push m push a mul add \n17164 ▶ ▶ ▶ ▶ mul mul add \n17165 ▶ ▶ ▶ ▶ store c002 \n17166 ▶ ▶ ▶ ▶ push b push b \n17167 ▶ ▶ ▶ ▶ push a push c push m mul add mul \n17168 ▶ ▶ ▶ ▶ push c push a push m mul add mul add mul \n17169 ▶ ▶ ▶ ▶ push a push d mul push c push 1 push m mul add mul \n17170 ▶ ▶ ▶ ▶ push m mul add \n17171 ▶ ▶ ▶ ▶ push a push c push m mul add mul \n17172 ▶ ▶ ▶ ▶ store c011 \n17173 ▶ ▶ ▶ ▶ push b push b push c mul \n17174 ▶ ▶ ▶ ▶ push 1 push d push m mul add mul \n17175 ▶ ▶ ▶ ▶ push a push b mul push c push d push d push m mul mul add mul \n17176 ▶ ▶ ▶ ▶ push m mul add \n17177 ▶ ▶ ▶ ▶ push a push d mul push c push d push m mul add mul add \n17178 ▶ ▶ ▶ ▶ push a push c push m mul add mul \n17179 ▶ ▶ ▶ ▶ store c012 \n17180 ▶ ▶ ▶ ▶ push m push b push b \n17181 ▶ ▶ ▶ ▶ push b push d push m mul add push c mul \n17182 ▶ ▶ ▶ ▶ push d push b push 1 push m mul add mul push m mul add push a mul \n17183 ▶ ▶ ▶ ▶ push b push c mul push d push 1 push m mul add add mul \n17184 ▶ ▶ ▶ ▶ push b push d push m mul add mul \n17185 ▶ ▶ ▶ ▶ store d001 \n17186 ▶ ▶ ▶ ▶ push m push \n17187 ▶ ▶ ▶ ▶ push d push c push m mul add push a push a mul \n17188 ▶ ▶ ▶ ▶ push c push c mul push d push m mul add push a mul add \n17189 ▶ ▶ ▶ ▶ push m push b push c mul push d push 1 push m mul add mul add mul \n17190 ▶ ▶ ▶ ▶ push b push d push m mul add mul \n17191 ▶ ▶ ▶ ▶ store d011 \n17192 ▶ ▶ ▶ ▶ push m \n17193 ▶ ▶ ▶ ▶ push c push d mul push d push m mul add push a push a mul mul
17194 ▶ ◀ ◀ ◀ push·c·push·c·mult·push·d·push·m·mult·add·push·a·push·b·push·m·mult·mult·add\n17195 ▶ ◀ ◀ push·b·push·d·push·c·push·m·mult·add·mult·push·c·push·m·mult·mult·add\n17196 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult·mult\n17197 ▶ ◀ ◀ store·d012\n17198 ▶ ◀ ◀ push·d·push·1·push·m·mult·add·push·a·mult·push·m·push·b·mult·push·1·add·p\n push·c·mult·add\n17199 ▶ ◀ ◀ push·b·add·push·m·push·d·mult·add\n17200 ▶ ◀ ◀ push·a·push·c·mult·mult\n17201 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult\n17202 ▶ ◀ ◀ store·d112\n17203 ▶ ◀ ◀ push·m\n17204 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·push·c·mult·push·d·push·b·push·1·push·m·mul t·add·mult\n17205 ▶ ◀ ◀ push·m·mult·add·push·a·mult·push·b·push·c·mult·push·d·push·1·push·m·mult·add·mult·add\n17206 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult·mult\n17207 ▶ ◀ ◀ store·m002\n17208 ▶ ◀ ◀ push·m\n17209 ▶ ◀ ◀ push·d·push·c·push·m·mult·add·push·a·push·a·mult·mult\n17210 ▶ ◀ ◀ push·c·push·c·mult·push·d·push·m·mult·add·push·a·mult·add\n17211 ▶ ◀ ◀ push·b·push·c·push·m·mult·push·c·push·1·push·m·mult·add·mult·add\n17212 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult·mult\n17213 ▶ ◀ ◀ store·m012\n17214 ▶ ◀ ◀ push·m\n17215 ▶ ◀ ◀ push·c·push·d·mult·push·d·push·m·mult·add·push·a·push·a·mult·mult\n17216 ▶ ◀ ◀ push·m·push·c·push·c·mult·push·d·push·m·mult·add·push·a·push·b·mult·mult·mult·add\n17217 ▶ ◀ ◀ push·m·push·b·push·d·push·c·push·m·mult·add·push·c·push·m·mult·mult·mult·add\n17218 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult·mult\n17219 ▶ ◀ ◀ store·m022\n17220 ▶ ◀ ◀ push·d·push·1·push·m·mult·add·push·a·mult\n17221 ▶ ◀ ◀ push·m·push·b·mult·push·1·add·push·c·mult·add\n17222 ▶ ◀ ◀ push·b·add·push·m·push·d·mult·add\n17223 ▶ ◀ ◀ push·a·push·c·mult·mult\n17224 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult\n17225 ▶ ◀ ◀ store·m122\n17226 ▶ ◀ ◀ push·m·push·a·mult·push·c·add·push·d·mult·push·c·push·a·push·1·push·m·mult t·add·mult·add\n17227 ▶ ◀ ◀ push·b·mult\n17228 ▶ ◀ ◀ push·m·push·a·push·d·mult·mult·push·c·push·1·push·m·mult·add·mult·add\n17229 ▶ ◀ ◀ push·b·push·d·push·m·mult·add·mult\n17230 ▶ ◀ ◀ store·m002\n17231 ▶ ◀ ◀ push·m\n17232 ▶ ◀ ◀ push·c·push·d·push·m·mult·add·push·b·mult·push·m·push·d·push·c·push·1·push b·m·mult·add·mult·mult·mult·add\n17233 ▶ ◀ ◀ push·a·mult\n
825
17234 ▶ ▶ ▶ ▶ push-b-push-c-mul-push-d-push-1-push-m-mul-add-mul-add-mul-
17235 ▶ ▶ ▶ ▶ push-a-push-b-push-c-push-1-push-m-mul-push-m-adv-add-add-mul-
17236 ▶ ▶ ▶ ▶ store-n012;
17237 ▶ ▶ ▶ ▶ push-c-push-d-push-m-adv-push-b-mul-
17238 ▶ ▶ ▶ ▶ push-m-push-d-push-c-push-1-push-m-mul-push-m-adv-
17239 ▶ ▶ ▶ ▶ push-a-mul-
17240 ▶ ▶ ▶ ▶ push-b-push-c-push-1-push-m-adv-push-m-adv-
17241 ▶ ▶ ▶ ▶ push-a-push-d-push-m-push-b-push-c-push-1-push-m-adv-
17242 ▶ ▶ ▶ ▶ store-n022;
17243 ▶ ▶ ▶ ▶ push-m-
17244 ▶ ▶ ▶ ▶ push-c-push-d-push-m-adv-push-b-mul-
17245 ▶ ▶ ▶ ▶ push-m-push-d-push-c-push-1-push-m-adv-push-m-adv-
17246 ▶ ▶ ▶ ▶ push-a-mul-
17247 ▶ ▶ ▶ ▶ push-b-push-c-push-d-push-1-push-m-adv-push-m-adv-
17248 ▶ ▶ ▶ ▶ push-m-push-a-mul-push-c-push-1-push-m-adv-
17249 ▶ ▶ ▶ ▶ store-n112;
17250 ▶ ▶ ▶ ▶ push-m-
17251 ▶ ▶ ▶ ▶ push-c-push-d-push-m-adv-push-b-mul-
17252 ▶ ▶ ▶ ▶ push-m-push-d-push-c-push-1-push-m-adv-push-m-adv-
17253 ▶ ▶ ▶ ▶ push-a-mul-
17254 ▶ ▶ ▶ ▶ push-b-push-c-push-d-push-1-push-m-adv-push-m-adv-
17255 ▶ ▶ ▶ ▶ push-a-push-d-push-m-push-b-push-c-push-m-adv-push-m-adv-
17256 ▶ ▶ ▶ ▶ store-n122.
17257
17258 Clebsch_up_create_points:
17259 ▶ $\textit{ORBITER\_PATH}/\textit{orbiter.out} -v.2$
17260 ▶ ▶ -smooth_curve:"Clebsch_map_of_circle_to_defining_eqn_r2"$
17261 ▶ ▶ ▶ ▶ 0.07-1000 5.0 $\text{(TWO\_PI)}$
17262 ▶ ▶ ▶ ▶ -const-a-$\text{(CLEBSCH_A)}$\cdot b-$\text{(CLEBSCH_B)}$\cdot c-$\text{(CLEBSCH_C)}$\cdot d-$\text{(CLEBSCH_D)}$
17263 ▶ ▶ ▶ ▶ t00-$\text{(T00)}$\cdot t01-$\text{(T01)}$\cdot t02-$\text{(T02)}$\cdot t03-$\text{(T03)}$
17264 ▶ ▶ ▶ ▶ t10-$\text{(T10)}$\cdot t11-$\text{(T11)}$\cdot t12-$\text{(T12)}$\cdot t13-$\text{(T13)}$
17265 ▶ ▶ ▶ ▶ t20-$\text{(T20)}$\cdot t21-$\text{(T21)}$\cdot t22-$\text{(T22)}$\cdot t23-$\text{(T23)}$
17266 ▶ ▶ ▶ ▶ t30-$\text{(T30)}$\cdot t31-$\text{(T31)}$\cdot t32-$\text{(T32)}$\cdot t33-$\text{(T33)}$
17267 ▶ ▶ ▶ ▶ r-2.\text{one}-1.\text{m}-1$
17268 ▶ ▶ ▶ ▶ -const_end$
17269 ▶ ▶ ▶ ▶ -var-t$
17270 ▶ ▶ ▶ ▶ c001-c002-c011-c012$
17271 ▶ ▶ ▶ ▶ d001-d011-d012-d112$
17272 ▶ ▶ ▶ ▶ m002-m012-m022-m122$
17273 ▶ ▶ ▶ ▶ n002-n012-n112-n022-n122$
17274 ▶ ▶ ▶ ▶ y0-y1-y2$
17275 ▶ ▶ ▶ ▶ y001-y002-y011-y012-y022-y112-y122$
17276 ▶ ▶ ▶ ▶ x0-x1-x2-x3$
17277 ▶ ▶ ▶ ▶ -var_end$
17278 ▶ ▶ ▶ ▶ -code$
17279 ▶ ▶ ▶ ▶ push-t-cos-push-r-mult-store-y0$
17280 ▶ ▶ ▶ ▶ push-t-sin-push-r-mult-store-y1
17281 ▶ ▶ ▶ push-one-store-y2;
17282 ▶ ▶ ▶ push-y0-push-y0-push-y1-mult mult-store-y001;
17283 ▶ ▶ ▶ push-y0-push-y0-push-y2-mult mult-store-y002;
17284 ▶ ▶ ▶ push-y0-push-y1-push-y1-mult mult-store-y011;
17285 ▶ ▶ ▶ push-y0-push-y1-push-y2-mult mult-store y012;
17286 ▶ ▶ ▶ push-y0-push-y2-push-y2-mult mult-store-y022;
17287 ▶ ▶ ▶ push-y1-push-y1-push-y2-mult mult-store-y112;
17288 ▶ ▶ ▶ push-y1-push-y2-push-y2-mult mult-store-y122;
17289 ▶ ▶ ▶ $(CLEBSCH_CUBICS)-;
17290 ▶ ▶ ▶ push-c001-push-y001-mult-
17291 ▶ ▶ ▶ push-c002-push-y002-mult-add-
17292 ▶ ▶ ▶ push-c011-push-y011-mult-add-
17293 ▶ ▶ ▶ push-c012-push-y012-mult-add-
17294 ▶ ▶ ▶ store-x0-
17295 ▶ ▶ ▶ push-d001-push-y001-mult-
17296 ▶ ▶ ▶ push-d011-push-y011-mult-add-
17297 ▶ ▶ ▶ push-d012-push-y012-mult-add-
17298 ▶ ▶ ▶ push-d112-push-y112-mult-add-
17299 ▶ ▶ ▶ store-x1-
17300 ▶ ▶ ▶ push-m002-push-y002-mult-
17301 ▶ ▶ ▶ push-m012-push-y012-mult-add-
17302 ▶ ▶ ▶ push-m022-push-y022-mult-add-
17303 ▶ ▶ ▶ push-m122-push-y122-mult-add-
17304 ▶ ▶ ▶ store-x2-
17305 ▶ ▶ ▶ push-n002-push-y002-mult-
17306 ▶ ▶ ▶ push-n012-push-y012-mult-add-
17307 ▶ ▶ ▶ push-n022-push-y022-mult-add-
17308 ▶ ▶ ▶ push-n112-push-y112-mult-add-
17309 ▶ ▶ ▶ push-n122-push-y122-mult-add-
17310 ▶ ▶ ▶ store-x3-
17311 ▶ ▶ ▶ push-x0-push-x00-mult-
17312 ▶ ▶ ▶ push-x1-push-t10-mult-add-
17313 ▶ ▶ ▶ push-x2-push-t20-mult-add-
17314 ▶ ▶ ▶ push-x3-push-t30-mult-add-
17315 ▶ ▶ ▶ return-
17316 ▶ ▶ ▶ push-x0-push-t01-mult-
17317 ▶ ▶ ▶ push-x1-push-t11-mult-add-
17318 ▶ ▶ ▶ push-x2-push-t21-mult-add-
17319 ▶ ▶ ▶ push-x3-push-t31-mult-add-
17320 ▶ ▶ ▶ return-
17321 ▶ ▶ ▶ push-x0-push-t02-mult-
17322 ▶ ▶ ▶ push-x1-push-t12-mult-add-
17323 ▶ ▶ ▶ push-x2-push-t22-mult-add-
17324 ▶ ▶ ▶ push-x3-push-t32-mult-add-
17325 ▶ ▶ ▶ return-
17326 ▶ ▶ ▶ push-x0-push-t03-mult-
17327 ▶ ▶ ▶ push-x1-push-t13-mult-add-
Clebsch_surface:

```bash
17334 Clebsch_surface:
17335   $(ORBITER_PATH)orbiter.out -v 2 -povray
17336   -round 0 -nb_frames_default 30
17337   -output_mask Clebsch_%d_03d.pov
17338   -video_options -W 1024 -H 768
17339   -global_picture_scale 0.9
17340   -default_angle 75
17341   -clipping_radius 2.4
17342   -camera 0 "1,1,1" "-3,1,3" "0.12,0.12,0.12"
17343   -end
17344   -scene_objects
17345   -cubic_orbiter "0,0,0,0,0,-4.236067972,
17346       0,0,4.236067972,4.236067972,17.94427188,
17347       -17.94427188,0,0,-9.472135941,0,0,5.236067971,
17348       8.472135938,-27.41640782"
17349   -group_of_things "0"
17350   -cubics 0 "texture { pigments { White*0.5 transmit 0.5 } }
17351   finish { ambient 0.4 diffuse 0.5 roughness 0.001
17352     reflection 0.1 specular .8 }
17353   -point_list_from_csv_file
17354   -function Clebsch_map_of_circle N1000_points.csv
17355   -group_of_things_as_interval 0.954
17356   -spheres 1 0.07 "texture { pigment { color Red } }
17357   finish { diffuse 0.9 phong 1 }
17358   -scene_objects_end
17359   -povray_end
17360   -rm -rf POV
17361   mkdir POV
17362   mv Clebsch_0_*.pov POV
17363   mv makefile_animation POV
17364
17365 Clebsch_surface_defining_equation:
17366   $(ORBITER_PATH)orbiter.out -v 2 -povray
17367   -round 0 -nb_frames_default 30
17368   -output_mask Clebsch_%d_03d.pov
17369   -video_options -W 1024 -H 768
17370   -global_picture_scale 0.6
17371   -default_angle 75
17372   -clipping_radius 1.6
17373   -camera 0 "1,1,1" "-2,0,2" "0,0,0"
```

828
Clebsch_surface_defining_equation_and_curves:

$\$(\text{ORBITER\_PATH})\text{orbiter.out}\text{-v2.povray}\$

-\text{round-0.nb\_frames\_default}\text{-30}\$

-\text{output\_mask\_Clebsch\_2curves}\%d\_000\_003\_pov\$

-\text{video\_options}\text{-W}1024\text{-H}768

-\text{global\_picture\_scale}0.6$

-\text{default\_angle}75$

-\text{clipping\_radius}1.6$

-\text{camera}0.1,1,1.1,2,0,2,.0,0,0$

-\text{end}$

-\text{scene\_objects}$

-\text{cubic\_orbiter}\text{-0.0.0.1.1.1.1.1,1.1.1.2,2,2,2}$

-\text{group\_of\_things}0$

-\text{cubics}\text{-0.} \text{texture\{pigment\{White*0.5.\_transmit0.5.\}}$

\text{ambient-0.4.\_diffuse0.5.\_roughness0.001.\_reflection0.1.}\text{specular0.8}$

-\text{point\_list\_from\_csv\_file}$

-\text{function\_Clebsch\_map\_of\_circle\_to\_definining\_eqn\_N1000\_points.csv}$

-\text{group\_of\_things\_as\_interval}0.656$

-\text{spheres}1.0.0.07\text{-texture\{pigment\{color\_Red\}}$

\text{diffuse-0.9.}\text{phong1}$

-\text{point\_list\_from\_csv\_file}$

-\text{function\_Clebsch\_map\_of\_circle\_to\_definining\_eqn\_r2\_N1000\_points.csv}$

-\text{group\_of\_things\_as\_interval}656:1042$

-\text{spheres}2.0.0.07\text{-texture\{pigment\{color\_Blue\}}$

\text{diffuse-0.9.}\text{phong1}$

-\text{scene\_objects\_end}$

-\text{povray\_end}$

-\text{rm\text{-rf}POV}$

-\text{mkdirPOV}$

-\text{mv\_Clebsch\_0\_pov\_POV}$

-\text{mv\_makefile\_animation\_POV}$
17422 #> ▶ ▶ -point_list_from_csv_file: function_Clebsch_map_of_circle_N1000_points.csv
17423 #> ▶ ▶ -group_of_things_as_interval: 0.954
17424 #> ▶ ▶ -spheres: 1.0.07:"texture{pigment{color: Red} . finish{diffuse: 0.9. phong: 1}}"
17425
17426
17427
17428
17429 F7.povray:
17430 ▶ $ { ORBITER_PATH } orbiter.out-v:2-povray
17431 ▶ ▶ -round:0- nb_frames_default:30
17432 ▶ ▶ -output_mask: F7_15_lines_%d %d_03.pov
17433 ▶ ▶ -video_options: W:1024-H:768
17434 ▶ ▶ -global_picture_scale: 1.5
17435 ▶ ▶ -default_angle: 80
17436 ▶ ▶ -clipping_radius: 4.4
17437 ▶ ▶ -omit_bottom_plane
17438 ▶ ▶ -camera: 0."1,1,1"."4.5,3.5,6"."0,0,0".
17439 ▶ ▶ -end
17440 ▶ ▶ -scene_objects:
17441 ▶ ▶ ▶ -cubic_lex: "0,0.6,0,0,-13.39014946,-3.341901346,-6.972931640,5.82718,-1,7.390149464,-7.390149464,-6.972931640,-1.512349728,-8.485281372,0,0,0"
17442 ▶ ▶ ▶ -group_of_things: 0
17443 ▶ ▶ ▶ -cubics: 0."texture{pigment{White*0.5} . finish{ambient:0.4 . diffuse:0.5 . roughness:0.001 . reflection:0.1 . specular:0.8}}"
17444 ▶ ▶ ▶ -line_through_point_with_direction: "0,0,0,1,0,0"
17445 ▶ ▶ ▶ -line_through_point_with_direction: "0,0,-1,0,1,0"
17446 ▶ ▶ ▶ -line_through_point_with_direction: "0,0,0,0,0,-1"
17447 ▶ ▶ ▶ -line_through_point_with_direction: "1,0,0,1,1,1"
17448 ▶ ▶ ▶ -line_through_point_with_direction: "-1.414213562,0,0,4.146264370,1.732050808,1.732050808"
17449 ▶ ▶ ▶ -line_through_point_with_direction: "0,1.732050808,-1.2.414213562,-0.317837246,2.414213562"
17450 ▶ ▶ ▶ -line_through_point_with_direction: ",2.13352390,0,-1.1.674708020,1,0"
17451 ▶ ▶ ▶ -line_through_point_with_direction: ",2.539058015,0,0,0.2.11360755,1,0"
17452 ▶ ▶ ▶ -line_through_point_with_direction: "0,1.148188060,0,0,-0.9435440612,1"
17453 ▶ ▶ ▶ -line_through_point_with_direction: "0.971971171,0,0,1.162155272,0,1"
17454 ▶ ▶ ▶ -line_through_point_with_direction: "2.096037870,2.096037870,-1.065851905,-1.065851905,1"
17455 ▶ ▶ ▶ -line_through_point_with_direction: "3.921555783,2.921555781,0,0,1.722456585,1"

830
-group_of_things_as_interval-0.12"
lines:1.0.04":"texture{pigment{color:Yellow}finish{diffuse:0.9:phong:1}}"::
-scene_objects_end::
povray_end
-rm-r.POVR
mkdir-POV
mv-F7_15_lines_0_.pov-POV
mv-makefile_animation-POV

F7_video:
-rm-r.FRAMES
 mkdir-FILES
-rm-fifteen_with_lines.mp4
$(ORBITER_PATH)orbiter.out-
-prepare_frames::
 -i-30-F7b/F7_15_lines_0_03d.png
-output_starts_at0::
-o-FILES/frame%04d.png
-end
ffmpeg-r.5-f.image2-i-FILES/frame%04d.png
-f.mp4-q:0-v:0-vcodec:mpeg4:fifteen_with_lines.mp4

McKean_povray:
$(ORBITER_PATH)orbiter.out-v.2-povray-
-round0-:nb_frames_default.30-
-output_mask:McKean_0%03d.pov-
-video_options:-W:1024:-H:768-
global_picture_scale:1.5-
default_angle:80-
-clipping_radius:4.4-
-omit_bottom_plane-
camera0:1,1,1":-4.5,3.5,6":0,0,0":
-end::
-scene_objects::
cubic_lex:0,0,1,0,0,-1,-2,1,
2,0,0,1,1,-1,0,0,0":
group_of_things:"0":
cubics:0:"texture{pigment{White*0.5}}":
finish{ambient:0.4:diffuse:0.5:roughness:0.001:
reflection:0.1:specular:.8}"::

831
lissajous:

```
(ORBITER_PATH)orbiter.out -v 2
smooth_curve: "lissajous" 0.07 2000 0 18 85
const a 3 b 2 c 1.57 r 7 const_end
var t var_end
code
push t push a mult push c add sin push r mult return
push t push b mult sin push r mult return
push_d return
```

#function_lissajous_N2000_points.csv

lissajous_plot:

```
(ORBITER_PATH)orbiter.out -v 2 -povray
round 0 nb_frames_default 1
output_mask lissajous%d%03d.pov
```
17549 ▶ ▶ -video_options:-W1024-H768:\n17550 ▶ ▶ -global_picture_scale-0.40:\n17551 ▶ ▶ -default_angle-45:\n17552 ▶ ▶ -clipping_radius-5.\n17553 ▶ ▶ -omit_bottom_plane:\n17554 ▶ ▶ -camera-0-"0,-1,0","0,0,12","0,0,0":\n17555 ▶ ▶ -rotate_about_z_axis:\n17556 ▶ ▶ -end:\n17557 ▶ ▶ -scene_objects:\n17558 ▶ ▶ ▶ -line_through_two_points_recentered_from_csv_file:\n17559 ▶ ▶ ▶ coordinate_grid.csv:\n17560 ▶ ▶ ▶ -group_of_things-"0":\n17561 ▶ ▶ ▶ -group_of_things-"1":\n17562 ▶ ▶ ▶ -group_of_things-"2":\n17563 ▶ ▶ ▶ ▶ -lines-0.0.09-"texture{pigment{color-Yellow}}":\n17564 ▶ ▶ ▶ ▶ -lines-1.0.09-"texture{pigment{color-Yellow}}":\n17565 ▶ ▶ ▶ ▶ -lines-2.0.09-"texture{pigment{color-Yellow}}":\n17566 ▶ ▶ ▶ ▶ -group_of_things_as_interval-3:39:\n17567 ▶ ▶ ▶ ▶ -lines-3.0.02-"texture{pigment{color-Black}}":\n17568 ▶ ▶ ▶ ▶ -point_list_from_csv_file:\n17569 ▶ ▶ ▶ ▶ function_lissajous_N2000.points.csv:\n17570 ▶ ▶ ▶ ▶ -group_of_things_as_interval-0:6524:\n17571 ▶ ▶ ▶ ▶ -spheres-4.0.1-"texture{pigment{color-Red}}":\n17572 finish-{:diffuse-0.9-\n17573 ▶ ▶ ▶ ▶ plane_by_dual_coordinates-"0,0,1,0":\n17574 ▶ ▶ ▶ ▶ -group_of_things-\n17575 ▶ ▶ ▶ ▶ -planes-5-"texture{pigment{color-Blue*0.5}:\n17576 transmit-0.5-}":\n17577 ▶ ▶ -scene_objects_end-\n17578 ▶ -povray_end\n17579 ▶ -rm-\n17580 ▶ mkdir-\n17581 ▶ mv-lissajous_0_.pov-\n17582 ▶ mv-makefile_animation-\n17583 lissajous_3d:\n17584 ▶ $(ORBITER_PATH)orbiter.out-\-v-2-\n17585 ▶ -smooth_curve-lissajous_3d-0.07-2000.50-0.18.85-\n17586 ▶ -const-a-3-b-2-c-1.57r-7-\n17587 ▶ -const_end-\n17588 ▶ -var-t-\n17589 ▶ -code-\n17590 ▶ ▶ push-t-push-a-mult-push-c-add-sin-push-r-mult-return-\n17591 ▶ ▶ push-t-push-b-mult-sin-push-r-mult-return-\n17592 ▶ ▶ push-t-return-\n17593 ▶ ▶ -code_end-\n17594 lissajous_3d_plot:
$(ORBITER\_PATH)orbiter.out-v.2.-povray\ 
-round-0--nb_frames_default-30-\ 
-output_mask.lissajous_3d_%d_%03d.pov\ 
-video_options-W:1024-H:768-\ 
-global_picture_scale:0.40-\ 
default_angle:45-\ 
-clipping_radius:5-\ 
-omit_bottom_plane-\ 
camera:0."0,0,1"."7,7,5"."0,0,1"-\ 
-rotate_about_z_axis-\ 
-end-\ 
-scene_objects-\ 
-line_through_two_points_recentered_from_csv_file-\ 
-coordinate_grid.csv-\ 
group_of_things:"0"-\ 
group_of_things:"1"-\ 
group_of_things:"2"-\ 
-lines:0.0.09."texture{pigment{color:Yellow}.}"-\ 
-lines:1.0.09."texture{pigment{color:Yellow}.}"-\ 
-lines:2.0.09."texture{pigment{color:Yellow}.}"-\ 
group_of_things_as_interval:3:39-\ 
-lines:3.0.02."texture{pigment{color:Black}.}"-\ 
-point_list_from_csv_file-\ 
-function_lissajous_3d_N2000_points.csv-\ 
group_of_things_as_interval:0:6538-\ 
spheres:4.0.1."texture{pigment{color:Red}.}"
finish{diffuse:0.9-phong:1}}-\ 
-plane_by_dual_coordinates:"0,0,1,0"-\ 
group_of_things:"0"-\ 
-scene_objects_end-\ 
povray_end-\ 
-rm:-rf-POV
mkdir-POV
mv.lissajous_3d_0.*.pov-POV
mv.makefile_animation-POV
planes:5."texture{pigment{color:Blue*0.5-transmit:0.5}.}"-\ 
eightnotes
misc_select:
17651 $(ORBITER_PATH)orbiter.out-\(\text{-v.3}\) \\
17652  \-define\(\text{F-}\)finite_field-q\(\text{7-}\)end\) \\
17653  \-with\(\text{F-}\)do\(\text{-}\)finite_field_activity\(\text{-}\)cheat_sheet_GF\(\text{-}\)end \\
17654 $(ORBITER_PATH)orbiter.out-\(\text{-v.4-}\)csv_file_select_rows_and_cols\) \\
17655  \-GF\(\text{7_multiplication_table_reordered.csv}\) \\
17656  \"0,2,4\".\0,2,4\".
17657
17658
misc_join:
17660 $(ORBITER_PATH)orbiter.out-\(\text{-v.4}\) \\
17661  \-csv_file_join_poly_orbits_d3_n3_q2_select_F2.csv-\text{Orbit_idx}\) \\
17662  \-csv_file_join_poly_orbits_d3_n3_q2_select_F4.csv-\text{Orbit_idx}\) \\
17663  \-csv_file_join_poly_orbits_d3_n3_q2_select_F8.csv-\text{Orbit_idx}\) \\
17664  \-csv_file_join_poly_orbits_d3_n3_q2_select_F16.csv-\text{Orbit_idx}\) \\
17665  \-csv_file_join_poly_orbits_d3_n3_q2_select_F32.csv-\text{Orbit_idx}\)
17666
17667
table_mod_12:
17669 $(ORBITER_PATH)orbiter.out-\(\text{-v.2}\) \\
17670  \-define\(M\)-vector-load_csv_no_border_clock_mult_excel.csv\-end\) \\
17671  \-define\(\text{all_one_r-}\)vector\-repeat\(\text{-}\)1\(\text{-}\)12\(\text{-}\)end\) \\
17672  \-define\(\text{all_one_c-}\)vector\-repeat\(\text{-}\)1\(\text{-}\)12\(\text{-}\)end\) \\
17673  \-draw_matrix\) \\
17674  \-input\_\text{object}\(M\)\) \\
17675  \-box_width\(\text{50-}\)\-bit_depth\(\text{24}\) \\
17676  \-partition\(\text{3}\) \\
17677  \-all_one_r\(\text{-}\)all_one_c\) \\
17678  \-end
17679
17680
17681
17682 SECTION_LIMITATIONS:
17683
17684
17685
17686
17687
17688 ###
17689

835
17690
17691 # unclassified:
17692
17693
17694
17695 planes_in_pencil:
17696 ➤ $(ORBITER_PATH)orbiter.out -v:2:
17697 ➤ ▶ -define:F:-finite_field:-q:8:-end:
17698 ➤ ▶ -define:P:-projective_space:3-F:-end:
17699 ➤ ▶ -with:P:-do:
17700 ➤ ▶ ▶ -projective_space_activity:
17701 ➤ ▶ ▶ ▶ -planes_through_line:0:
17702 ➤ ▶ ▶ -end
17703
17704 on_planes:
17705 ➤ $(ORBITER_PATH)orbiter.out -v:2:
17706 ➤ ▶ -define:F:-finite_field:-q:8:-end:
17707 ➤ ▶ -define:P:-projective_space:3-F:-end:
17708 ➤ ▶ -define:G:-linear_group:-PGL:4:F:-on_k_subspaces:3:-end:
17709 ➤ ▶ ▶ -with:G:-do:
17710 ➤ ▶ ▶ -group_theoretic_activity:
17711 ➤ ▶ ▶ ▶ -apply:"0,8,1,6,4,3,7,2,5":"1,0,0,0,0,1,0,0,0,0,2,0,0,1,1":
17712 ➤ ▶ ▶ -end
17713 ➤ pdflatex PGL_4_8_Gr_4_3_apply.tex
17714 ➤ open PGL_4_8_Gr_4_3_apply.pdf
17715
17716
17717
Bibliography


[37] Fatma Karaoglu and Anton Betten. The number of cubic surfaces with 27 lines over a finite field. To appear in *Journal of Algebraic Combinatorics*.


[57] L. Schläfli. An attempt to determine the twenty-seven lines upon a surface of the third order and to divide such surfaces into species in reference to the reality of the lines upon the surface, *Quart. J. Math.* 2 (1858), 55–110.


