User’s Guide
Build Number 1420

Anton Betten

February 10, 2022
Contents

1 Introduction

2 Getting Started
  2.1 Running and Installing Orbiter
  2.2 The Orbiter Session
  2.3 Makefiles and Shell Scripts
  2.4 Objects and Activities
  2.5 Mathematical Data
  2.6 Set Builder
  2.7 Vector Builders

3 Basic Algebra
  3.1 Basic Number Theory
  3.2 Prime Fields
  3.3 Polynomials Over Finite Fields
  3.4 Extension Fields
  3.5 Linear Algebra Over Finite Fields
  3.6 Advanced Topics in Finite Fields

4 Geometry
  4.1 Finite Projective Spaces
  4.2 Indexing Points and Lines
  4.3 Finite Desarguesian Projective Planes
  4.4 The Grassmannian
  4.5 Algebraic Sets
  4.6 The Klein Quadric and the Plücker Map
  4.7 Orthogonal Spaces
  4.8 Hermitian Varieties
  4.9 Advanced Topics
  4.10 Geometric Objects

5 Group Theory
  5.1 Permutation Groups
  5.2 Linear Groups
  5.3 Subgroups
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.8</td>
<td>Tactical Decompositions</td>
<td>304</td>
</tr>
<tr>
<td>11</td>
<td>Finite Geometry</td>
<td>309</td>
</tr>
<tr>
<td>11.1</td>
<td>Spreads</td>
<td>309</td>
</tr>
<tr>
<td>11.2</td>
<td>Translation Planes</td>
<td>315</td>
</tr>
<tr>
<td>11.3</td>
<td>Packings</td>
<td>319</td>
</tr>
<tr>
<td>11.4</td>
<td>BLT-Sets</td>
<td>325</td>
</tr>
<tr>
<td>12</td>
<td>Graph Theory</td>
<td>329</td>
</tr>
<tr>
<td>12.1</td>
<td>Creating Graphs</td>
<td>329</td>
</tr>
<tr>
<td>12.2</td>
<td>Graph Theoretic Activities</td>
<td>336</td>
</tr>
<tr>
<td>12.3</td>
<td>Classification of Graphs and Tournaments</td>
<td>342</td>
</tr>
<tr>
<td>12.4</td>
<td>Clique Finding</td>
<td>344</td>
</tr>
<tr>
<td>13</td>
<td>Combinatorial Objects</td>
<td>349</td>
</tr>
<tr>
<td>13.1</td>
<td>Combinatorial Objects</td>
<td>349</td>
</tr>
<tr>
<td>13.2</td>
<td>Encoding Combinatorial Objects</td>
<td>353</td>
</tr>
<tr>
<td>14</td>
<td>Canonical Forms with Nauty</td>
<td>357</td>
</tr>
<tr>
<td>14.1</td>
<td>Overview of Canonical Forms</td>
<td>357</td>
</tr>
<tr>
<td>14.2</td>
<td>Canonical Forms of Objects in Projective Space</td>
<td>359</td>
</tr>
<tr>
<td>14.3</td>
<td>Canonical Forms of Incidence Geometries</td>
<td>362</td>
</tr>
<tr>
<td>14.4</td>
<td>Canonical Forms of Objects from Design Theory</td>
<td>374</td>
</tr>
<tr>
<td>14.5</td>
<td>Canonical Forms of Linear Codes</td>
<td>375</td>
</tr>
<tr>
<td>14.6</td>
<td>Canonical Forms of General Codes</td>
<td>382</td>
</tr>
<tr>
<td>14.7</td>
<td>Canonical Forms of Graphs</td>
<td>383</td>
</tr>
<tr>
<td>15</td>
<td>Interfaces</td>
<td>389</td>
</tr>
<tr>
<td>15.1</td>
<td>Graphical Output</td>
<td>389</td>
</tr>
<tr>
<td>15.2</td>
<td>The Povray Interface</td>
<td>396</td>
</tr>
<tr>
<td>15.3</td>
<td>Creating Animations</td>
<td>409</td>
</tr>
<tr>
<td>15.4</td>
<td>Continuous Function Plotter</td>
<td>410</td>
</tr>
<tr>
<td>16</td>
<td>Miscellaneous</td>
<td>415</td>
</tr>
<tr>
<td>16.1</td>
<td>Miscellaneous</td>
<td>415</td>
</tr>
<tr>
<td>16.2</td>
<td>Limitations</td>
<td>418</td>
</tr>
<tr>
<td>17</td>
<td>Orbiter on Windows</td>
<td>419</td>
</tr>
<tr>
<td>17.1</td>
<td>Using Windows Subsystem Linux</td>
<td>419</td>
</tr>
<tr>
<td>18</td>
<td>The Makefile</td>
<td>429</td>
</tr>
<tr>
<td>18.1</td>
<td>The Makefile</td>
<td>429</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>800</td>
</tr>
</tbody>
</table>
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 snipplets 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

\texttt{make}

Once compiled, the Orbiter executable is

\texttt{src/apps/orbiter/\_\_out}

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

\begin{verbatim}
  orbiter
  work
\end{verbatim}

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 environment. 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).

```
1 OP=~/orbiter
2 OP2=$(OP)/src/apps/orbiter/
3 DOCKER_OPTS=run--it-
4     --volume ${PWD}:/mnt
5     /mnt/abetten/orbiter
6 #ORBITER_PATH=docker-$(DOCKER_OPTS)
7 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 snipplets 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
48c2f66af66abe: Pull complete
234b70d0479d: Pull complete
6fa07a00e2f0: Pull complete
9187bd98e241: Pull complete
ae87b7ef500b: Pull complete
260a2765fa99: Pull complete
27d6ff93a58: Pull complete
7a09e574418: 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:

```bash
git clone <github-orbiter-path>
```

where `<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 `git clone` command. After cloning is complete, enter the orbiter directory (`cd orbiter`).

(c) Issue the following commands to compile Orbiter:

```bash
make
make install
```

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 `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><code>-v</code></td>
<td><code>v</code></td>
<td>Set verbosity to <code>v</code>. Larger values of <code>v</code> lead to more text output. <code>v = 0</code> gives minimal output.</td>
</tr>
<tr>
<td><code>-list_arguments</code></td>
<td></td>
<td>Prints the command line arguments.</td>
</tr>
<tr>
<td><code>-seed</code></td>
<td><code>s</code></td>
<td>Seed the pseudo random number generator with the integer value <code>s</code>.</td>
</tr>
<tr>
<td><code>-memory_debug</code></td>
<td></td>
<td>Turn on dynamic memory debugging.</td>
</tr>
<tr>
<td><code>-override_polynomial</code></td>
<td><code>poly</code></td>
<td>Set the override polynomial for finite fields to <code>poly</code>.</td>
</tr>
<tr>
<td><code>-orbiter_path</code></td>
<td><code>p</code></td>
<td>Set the orbiter path to <code>p</code>. 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><code>-magma_path</code></td>
<td><code>p</code></td>
<td>Set the magma path to <code>p</code>. This is useful in case the Orbiter session has to create a magma process.</td>
</tr>
<tr>
<td><code>-fork</code></td>
<td><code>L M f t s</code></td>
<td>Fork new Orbiter sessions in parallel. The new sessions will be indexed by the values <code>i</code> that result from a loop with start value <code>f</code> and increment <code>s</code> bounded from above by <code>t</code>, equivalent to a C-loop of type “for (i=f; i &lt; t; i+= s).” Every occurrence of the string <code>L</code> in the argument list is replaced by the resulting value of the loop variable <code>i</code>. The forked process will write to a file whose name is described through the mask <code>M</code>. The actual file name results from using the printf command from the C-library for <code>M</code> 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 <code>L</code> replaced by the integer value of the loop counter. The number of Orbiter sessions forked is <code>(t − f)/s</code>. The orbiter path from -orbi ter_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 a 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 18.1. 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 a orbiter command with several options, say

```sh
orbiter.out -v 3 -define F -finite_field -q 16 -end \ 
 -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:

```
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
```

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 18.1. 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 `-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

```
-define LABEL KEYWORD EXTRAS -end
```

Here, `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 `KEYWORD` can be any of the commands in Table 2.2. The `EXTRAS` depend on the type of object created. the command `-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

```
object F_2:
  $(ORBITER_PATH)orbi...F--finite_field--q:2--end
```

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

```
object PG_3_2:
  $(ORBITER_PATH)orbi...F--finite_field--q:2--end
```

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 `-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 3.4.</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 9.6.</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 12.1.</td>
</tr>
<tr>
<td><code>-spread_table</code></td>
<td>A table of spreads. See Section 11.3.</td>
</tr>
<tr>
<td><code>-packing_with_symmetry_assumption</code></td>
<td>A generator for packings with assumed symmetry. See Section 11.3.</td>
</tr>
<tr>
<td><code>-packing_choose_fixed_points</code></td>
<td>A selection of fixed orbits for packings with assumed symmetry. See Section 11.3.</td>
</tr>
<tr>
<td><code>-packing_long_orbits</code></td>
<td>A search for long orbits for packings with assumed symmetry. See Section 11.3.</td>
</tr>
<tr>
<td><code>-graph_classification</code></td>
<td>An object which allows classifying graphs and tournaments. See Section 12.3.</td>
</tr>
<tr>
<td><code>-diophant</code></td>
<td>A diophantine system, i.e., a system of positive integer equations). See Section 10.2.</td>
</tr>
<tr>
<td><code>-design</code></td>
<td>A combinatorial design. See Section 10.5.</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 10.5.</td>
</tr>
<tr>
<td><code>-large_set_with_symmetry_assumption</code></td>
<td>An object to create a large set of designs. See Section 10.5.</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 11.1 and 11.2.

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><code>-finite_field_activity</code></td>
<td>An activity for finite fields, see Sections 3.2 and 3.4.</td>
</tr>
<tr>
<td><code>-projective_space_activity</code></td>
<td>An activity for a projective space, see Section 4.1.</td>
</tr>
<tr>
<td><code>-orthogonal_space_activity</code></td>
<td>An activity for an orthogonal space, see Section 4.7.</td>
</tr>
<tr>
<td><code>-group_theoretic_activity</code></td>
<td>An activity for a group, see Section 5.6.</td>
</tr>
<tr>
<td><code>-cubic_surface_activity</code></td>
<td>An activity for a cubic surface, see Section 7.1.</td>
</tr>
<tr>
<td><code>-quartic_curve_activity</code></td>
<td>An activity for a quartic curve, see Section 7.2.</td>
</tr>
<tr>
<td><code>-combinatorial_object_activity</code></td>
<td>An activity for a combinatorial object, see Section 4.5.</td>
</tr>
<tr>
<td><code>-graph_theoretic_activity</code></td>
<td>An activity for a graph, see Section 12.1.</td>
</tr>
<tr>
<td><code>-classification_of_cubic_surfaces_with_double_sixes_activity</code></td>
<td>An activity for a cubic surface, see Section 7.1.</td>
</tr>
<tr>
<td><code>-spread_table_activity</code></td>
<td>An activity associated with a table of spreads, see Section 11.3.</td>
</tr>
<tr>
<td><code>-packing_with_symmetry_assumption_activity</code></td>
<td>An activity related to creating packings with assumed symmetry group, see Section 11.3.</td>
</tr>
<tr>
<td><code>-packing_fixed_points_activity</code></td>
<td>An activity related to creating packings with assumed symmetry group, see Section 11.3.</td>
</tr>
<tr>
<td><code>-graph_classification_activity</code></td>
<td>An activity for a classification of graphs problem, see Section 12.3.</td>
</tr>
<tr>
<td><code>-diophant_activity</code></td>
<td>An activity for a diophantine system, see Section 10.2.</td>
</tr>
<tr>
<td><code>-design_activity</code></td>
<td>An activity for a combinatorial design, see Section 10.5.</td>
</tr>
<tr>
<td><code>-large_set_with_symmetry_assumption_activity</code></td>
<td>An activity related to creating large sets of designs with assumed symmetry group, see Section 10.6.</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 $\text{PG}(2, 2^e)$ 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-q5-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 11.4.

The command

```
create_surface_4_0:
  $(ORBITER_PATH)orbiter.out-v.3\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  -define_surface S4_0:-q.4-catalogue 0:-end\n  -end\n  -with S4_0:-do\n  -cubic_surface_activity\n  -report\n  -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 $\{2,3,5,7,11,13\}$:

```
set_of_primes:
▷ $(ORBITER_PATH)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.

```
set_interval:
▷ $(ORBITER_PATH)orbiter.out-v.2-define S-set:-loop 0.64.1-"end-
▷ ▷ -print_symbols
```

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

```
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 16.2.

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\n $> -define:F:-finite_field:-q:5:-end\n $> -define:v:-vector-field:F:-dense:0,1,2,3,4:-end\n $> -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\n $> -define:F:-finite_field:-q:5:-end\n $> -define:v:-vector-field:F:-format:2:-dense:0,1,2,3,4,0:-end\n $> -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:
▷ $(ORBITER_PATH)orbiter.out:-v.2\n▷  ▷ -define:F:-finite_field:-q:2:-end\n▷  ▷ -define:v:-vector:-field:F:-format:4:-dense:$HAMMING_CODE:-end\n▷  ▷ -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="110110001100001010000\n111111011100011001101\n0101010000010001101\n000001000001100101\n010000000010010101\n001000110000011111\n1110100111010001011\n000000000011100101\n000000000101000101\n0110111110100011111\n000000000011100101\n000000000101000101\n010000000001100101\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110\n000000001100000110"
```

```
matrix_example_co_1:
▷ $(ORBITER_PATH)orbiter.out:-v.2\n▷  ▷ -define:F:-finite_field:-q:2:-end\n▷  ▷ -define:v:-vector:-field:F:-format:22:-compact:$CONWAY_GEN1:-end\n▷  ▷ -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:

```bash
$ (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 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:

```bash
$ (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:

```bash
$ (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 \texttt{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 Euler’s totient function \( \varphi(n) \) and the cofactor \( \varphi(n)/\text{ord}(q,n) \).

For instance,

\begin{verbatim}
order of q mod n:
\& $(\text{ORBITER PATH})\text{orbiter.out -v -order of q mod n:2:3:151}\\
\& $(\text{ORBITER PATH})\text{orbiter.out -v 1 -csv_file latex 1:\}
\& \& \text{order of q mod n q2:3:151.csv}\\
\& \text{pdflatex order of q mod n q2:3:151.tex}\\
\& \text{open 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:
\& \& $(\text{ORBITER PATH})\text{orbiter.out -v 1 -eulerfunction interval 1 150}\\
\& \& $(\text{ORBITER PATH})\text{orbiter.out -v 1 -csv_file latex 1:\}
\& \& \& \text{table eulerfunction 1 150.csv}\\
\& \& \text{pdflatex table eulerfunction 1 150.tex}\\
\& \& \text{open 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 \( a \mapsto a^k \mod 11 \):

\begin{verbatim}
order of q mod n:
\& $(\text{ORBITER PATH})\text{orbiter.out -v -order of q mod n:2:3:151}\\
\& $(\text{ORBITER PATH})\text{orbiter.out -v -csv_file latex 1:\}
\& \& \text{order of q mod n q2:3:151.csv}\\
\& \text{pdflatex order of q mod n q2:3:151.tex}\\
\& \text{open order of q mod n q2:3:151.pdf}
\end{verbatim}
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-power_mod</td>
<td>$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 $g$, $u$, and $v$ 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>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>11</td>
<td>2</td>
<td>13</td>
<td>4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>13</td>
<td>12</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>12</td>
<td>1</td>
<td>13</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>16</td>
<td>2</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>22</td>
<td>2</td>
<td>19</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>21</td>
<td>6</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
<td>22</td>
<td>2</td>
<td>23</td>
<td>11</td>
<td>22</td>
<td>2</td>
<td>25</td>
<td>20</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>27</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>29</td>
<td>28</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>30</td>
<td>30</td>
<td>2</td>
<td>29</td>
<td>30</td>
<td>30</td>
<td>2</td>
<td>31</td>
<td>5</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>31</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>33</td>
<td>20</td>
<td>20</td>
<td>2</td>
<td>35</td>
<td>24</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>33</td>
<td>12</td>
<td>24</td>
<td>2</td>
<td>35</td>
<td>36</td>
<td>36</td>
<td>1</td>
<td>37</td>
<td>36</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>12</td>
<td>24</td>
<td>2</td>
<td>39</td>
<td>12</td>
<td>12</td>
<td>1</td>
<td>41</td>
<td>20</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>41</td>
<td>14</td>
<td>42</td>
<td>3</td>
<td>43</td>
<td>14</td>
<td>42</td>
<td>3</td>
<td>45</td>
<td>12</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>43</td>
<td>14</td>
<td>42</td>
<td>3</td>
<td>47</td>
<td>23</td>
<td>46</td>
<td>2</td>
<td>49</td>
<td>21</td>
<td>42</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>12</td>
<td>24</td>
<td>2</td>
<td>47</td>
<td>23</td>
<td>46</td>
<td>2</td>
<td>49</td>
<td>21</td>
<td>42</td>
<td>2</td>
</tr>
<tr>
<td>49</td>
<td>8</td>
<td>32</td>
<td>4</td>
<td>51</td>
<td>8</td>
<td>32</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: The values of the Eulerfunction

<table>
<thead>
<tr>
<th>( n )</th>
<th>PHI</th>
<th>( n )</th>
<th>PHI</th>
<th>( n )</th>
<th>PHI</th>
<th>( n )</th>
<th>PHI</th>
<th>( n )</th>
<th>PHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>26</td>
<td>12</td>
<td>51</td>
<td>32</td>
<td>76</td>
<td>36</td>
<td>101</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>27</td>
<td>18</td>
<td>52</td>
<td>24</td>
<td>77</td>
<td>60</td>
<td>102</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>28</td>
<td>12</td>
<td>53</td>
<td>52</td>
<td>78</td>
<td>24</td>
<td>103</td>
<td>102</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>29</td>
<td>28</td>
<td>54</td>
<td>18</td>
<td>79</td>
<td>78</td>
<td>104</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>30</td>
<td>8</td>
<td>55</td>
<td>40</td>
<td>80</td>
<td>32</td>
<td>105</td>
<td>48</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>31</td>
<td>30</td>
<td>56</td>
<td>24</td>
<td>81</td>
<td>54</td>
<td>106</td>
<td>52</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>32</td>
<td>16</td>
<td>57</td>
<td>36</td>
<td>82</td>
<td>40</td>
<td>107</td>
<td>106</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>33</td>
<td>20</td>
<td>58</td>
<td>28</td>
<td>83</td>
<td>82</td>
<td>108</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>34</td>
<td>16</td>
<td>59</td>
<td>58</td>
<td>84</td>
<td>24</td>
<td>109</td>
<td>108</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>35</td>
<td>24</td>
<td>60</td>
<td>16</td>
<td>85</td>
<td>64</td>
<td>110</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>36</td>
<td>12</td>
<td>61</td>
<td>60</td>
<td>86</td>
<td>42</td>
<td>111</td>
<td>72</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>37</td>
<td>36</td>
<td>62</td>
<td>30</td>
<td>87</td>
<td>56</td>
<td>112</td>
<td>48</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>38</td>
<td>18</td>
<td>63</td>
<td>36</td>
<td>88</td>
<td>40</td>
<td>113</td>
<td>112</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>39</td>
<td>24</td>
<td>64</td>
<td>32</td>
<td>89</td>
<td>88</td>
<td>114</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>40</td>
<td>16</td>
<td>65</td>
<td>48</td>
<td>90</td>
<td>24</td>
<td>115</td>
<td>88</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>41</td>
<td>40</td>
<td>66</td>
<td>20</td>
<td>91</td>
<td>72</td>
<td>116</td>
<td>56</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>42</td>
<td>12</td>
<td>67</td>
<td>66</td>
<td>92</td>
<td>44</td>
<td>117</td>
<td>72</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>43</td>
<td>42</td>
<td>68</td>
<td>32</td>
<td>93</td>
<td>60</td>
<td>118</td>
<td>58</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>44</td>
<td>20</td>
<td>69</td>
<td>44</td>
<td>94</td>
<td>46</td>
<td>119</td>
<td>96</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>45</td>
<td>24</td>
<td>70</td>
<td>24</td>
<td>95</td>
<td>72</td>
<td>120</td>
<td>32</td>
</tr>
<tr>
<td>21</td>
<td>12</td>
<td>46</td>
<td>22</td>
<td>71</td>
<td>70</td>
<td>96</td>
<td>32</td>
<td>121</td>
<td>110</td>
</tr>
<tr>
<td>22</td>
<td>10</td>
<td>47</td>
<td>46</td>
<td>72</td>
<td>24</td>
<td>97</td>
<td>96</td>
<td>122</td>
<td>60</td>
</tr>
<tr>
<td>23</td>
<td>22</td>
<td>48</td>
<td>16</td>
<td>73</td>
<td>72</td>
<td>98</td>
<td>42</td>
<td>123</td>
<td>80</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>49</td>
<td>42</td>
<td>74</td>
<td>36</td>
<td>99</td>
<td>60</td>
<td>124</td>
<td>60</td>
</tr>
<tr>
<td>25</td>
<td>20</td>
<td>50</td>
<td>20</td>
<td>75</td>
<td>40</td>
<td>100</td>
<td>40</td>
<td>125</td>
<td>100</td>
</tr>
</tbody>
</table>
Table 3.4: The function \( a \mapsto a^2 \mod 11 \)

\[
\begin{array}{|c|c|}
\hline
A & APOWK \\
\hline
0 & 0 \\
1 & 1 \\
2 & 4 \\
3 & 9 \\
4 & 5 \\
5 & 3 \\
6 & 3 \\
7 & 5 \\
8 & 9 \\
9 & 4 \\
10 & 1 \\
\hline
\end{array}
\]

power_function_2_mod_11:
\[
\text{$(ORBITER\ PATH)orbiter.out-v.5-power_function_mod_n.2:11$}
\]
\[
\text{$(ORBITER\ PATH)orbiter.out-v.1-csv\_file\_latex.1-power_function_k2_n11.csv$}
\]
\[
\text{pdflatex-power_function_k2_n11.tex}
\]
\[
\text{open-power_function_k2_n11.pdf}
\]

The result is shown in Table 3.3.

The command

PR29:
\[
\text{$(ORBITER\ PATH)orbiter.out-v.1-smallest\_primitive\_root.29$}
\]

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

PR2:
\[
\text{$(ORBITER\ PATH)orbiter.out-v.5-primitive\_root.915839$}
\]

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

PM2:
\[
\text{$(ORBITER\ PATH)orbiter.out-v.5-power\_mod.43085.49842.915839$}
\]

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:

\$\text{(ORBITER\_PATH)orbiter.out\,-v\,-inverse\_mod\,1865025205\,-2147483647}\n
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:

\$\text{(ORBITER\_PATH)orbiter.out\,-v\,-extended\_gcd\,1865025205\,-2147483647}\n
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:

\$\text{(ORBITER\_PATH)orbiter.out\,-v\,-power\_mod\,16807\,-1073741823\,-2147483647}\n
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:

\$\text{(ORBITER\_PATH)orbiter.out\,-v\,-square\_root\_mod\,33\,-41}\n
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
Figure 3.1: Cycle of powers of $b$ modulo 13

The command

draw_mod_13:

\$\texttt{(ORBITER\_PATH)or} \texttt{biter.out -v.2:}\$
\$\texttt{-draw_options -embedded -end:}\$
\$\texttt{-draw_mod_n -n.13 -file_mod_13 -power_cycle.2 -end}\$
\$\texttt{pdflatex-mod_13\_draw.tex}\$
\$\texttt{open-mod_13\_draw.pdf}\$

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

```
$\text{\textbackslash ORBITER\textbackslash PATH}\text{\textasciitilde orbiter.out}\text{-v.2}\text{-draw_options}\text{-scale}\text{\textasciitilde 0.8}\text{-embedded}\text{-end}\text{-draw_mod}_n\text{-n.127}\text{-file}\text{-mod\_127}\text{-power_cycle\_3}\text{-end}$
```

creates the drawing shown in Figure 3.2.
3.2 Prime Fields

Let \( F_q \) 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.

Important comment: *Orbiter implements finite fields using tables for addition and multiplication. This imposes a limitation on the size of the field that can be created.*

See Section 16.2 for a list of limitations of Orbiter.

If \( p \) is a prime number, the integer factor ring \( \mathbb{Z}/I(p) \) is a finite field. Here,

\[
I(p) = p\mathbb{Z} = \{pk \mid k \in \mathbb{Z}\} = \{0, \pm k, \pm 2k, \pm 3k, \ldots\}
\]

is the ideal of all integer multiples of \( p \). The elements of \( F_p \) 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 \( F_p \). For instance, in Figure 3.3 the addition and multiplication tables of \( F_7 \) 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 \( F_q \) 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,

```
F_2:
▷ $(ORBITER_PATH)orbiter.out-v3-defineF-finite_field-q2-end·\n▷ -withF-do-finite_field_activity-cheat_sheet_GF-end
▷ pdflatex·GF_2.tex
▷ open·GF_2.pdf
```

creates the finite field \( F_2 \) and produces a report for it.

By default, Orbiter precomputes tables for the arithmetic in finite fields. The presence of these tables helps improve the performance. However, there is a cost associated with computing these tables. The option `without_tables` prevents these tables from being computed.
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>-q</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>-override_polynomial</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 3.4.</td>
</tr>
<tr>
<td>-without_tables</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

This may be helpful for large fields. However, in this case the field operations are generally slower.

Table 3.6 lists basic Orbiter activities for finite fields. More activities will follow in Section 3.4. Here is the cheat sheet for \( \mathbb{F}_7 \). The element \( \alpha \) is a primitive element.

\[ Z_i = \log_\alpha (1 + \alpha^i) \]

<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>6</td>
<td>DNE</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>
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 $-1 \mod 11$:

```
F_11_product_of_all_nonzero_elements:
▷ $(ORBITER\_PATH)\verb|orbiter.out-\v.3|$
▷ ▷ -define:F-\finite_field-q.11.-end\verb|\$
▷ ▷ -define:S-vector-field.F-loop.1.11.1.-end\verb|\$
▷ ▷ -with:F-do--finite_field_activity--product_of_S--end
```

Suppose we want to create the Vandermonde matrix whose entries are $x_j^i$. 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.

```
F_7_vandermonde:
▷ $(ORBITER\_PATH)\verb|orbiter.out-\v.3|$
```

\[
\begin{array}{cccccc}
+ & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 1 & 2 & 3 & 4 & 5 & 6 & 0 \\
2 & 2 & 3 & 4 & 5 & 6 & 0 & 1 \\
3 & 3 & 4 & 5 & 6 & 0 & 1 & 2 \\
4 & 4 & 5 & 6 & 0 & 1 & 2 & 3 \\
5 & 5 & 6 & 0 & 1 & 2 & 3 & 4 \\
6 & 6 & 0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

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

$3^0 \equiv 1 \quad 3^4 \equiv 4$

$3^1 \equiv 3 \quad 3^5 \equiv 5$

$3^2 \equiv 2 \quad 3^6 \equiv 1$

$3^3 \equiv 6$
The output is shown below. The first matrix is $V = (x_i^j)$. The second matrix is $V^{-1}$:

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

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

$0, 3^0, 3^1, 3^2, \ldots, 3^6 = 0, 1, 3, 2, 6, 4, 5, 1$.

In the second ordering, the addition table of the prime field no longer exhibits cyclic structure.
Figure 3.4: Addition and multiplication table of $\mathbb{F}_7$ using a primitive element
3.3 Polynomials Over Finite Fields

For \( p \) prime, the finite field \( \mathbb{F}_p \) of order \( p \) can be constructed as factoring 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
\[
\text{-finite\_field\_activity} \quad \ldots \quad \text{-end}
\]
command sequence can be used to start a command requiring a finite field. The \(-q q\) option can be used to specify the order of the finite field. The \(-\text{override\_polynomial} \ a\) option can be used to specify the polynomial \( m(X) \) as integer \( a \) in the base \( p \) representation. This option can be ommitted, in which case Orbiter will use a precomputed and built-in polynomial. Table 3.7 lists Orbiter activities for polynomials over finite fields. For instance, the command

\[
\text{poly\_division2}: \\
\text{poly\_division2}:
\]
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
\[
Q(X) = X^7 + X^6 + X^5 + X^3 + 1, \quad R(X) = X^2.
\]
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 \(-\text{define}\) 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.
<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>$A(X) B(X)$ $M(X)$</td>
<td>Multiply the polynomials $A(X)$ and $B(X)$ modulo $M(X)$ in $\mathbb{F}_q[X]$.</td>
</tr>
<tr>
<td>-Berlekamp_matrix</td>
<td>$A(X)$</td>
<td>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: $1, X, X^2, \ldots, X^{d-1}$.</td>
</tr>
<tr>
<td>-polynomial_find_roots</td>
<td>$A(X)$</td>
<td>Find the roots of $A(X)$ over $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-make_table_of_irreducible_polynomials</td>
<td>$d$</td>
<td>Produces a list of all irreducible polynomials of degree $d$ over $\mathbb{F}_q$.</td>
</tr>
<tr>
<td>-find_CRC_polynomials</td>
<td>$t n k$</td>
<td>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 9.4.</td>
</tr>
</tbody>
</table>

Table 3.7: Finite Field Activities Related to Polynomials
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,

```
poly_gcd:
$ (ORBITER_PATH) orbiter.out -v 2$
```
3.4 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

\[
0, \\
1, \\
X, \\
X + 1.
\]

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

\[
\begin{array}{cccc}
0 & 1 & X & X+1 \\
0 & 0 & 1 & X \\
1 & 1 & 0 & X+1 \\
X & X & X+1 & 0 \\
X+1 & X+1 & X & 1 \\
\end{array}
\]

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 $X+1$:

\[
\begin{align*}
X \cdot X &= X^2 \equiv X + 1 \pmod{X^2 + X + 1}, \\
X \cdot (X + 1) &= X^2 + X \equiv 1 \pmod{X^2 + X + 1}, \\
(X + 1) \cdot X &= X^2 + X \equiv 1 \pmod{X^2 + X + 1}, \\
(X + 1) \cdot (X + 1) &= X^2 + 1 \equiv X \pmod{X^2 + X + 1},
\end{align*}
\]

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

\[
\begin{array}{cccc}
0 & 1 & X & X+1 \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & X \\
X & 0 & X & X+1 \\
X+1 & 0 & X+1 & 1 \\
\end{array}
\]

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 3.8 lists Orbiter activities for finite fields. This extends Table 3.6 in Section 3.4.

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 $F_{q^d}$.</td>
</tr>
</tbody>
</table>

Table 3.8: 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 $F_q$. This just means that it is a generator of the multiplicative group. So, any non-zero element in $F_q$ is a suitable power of $\alpha$.

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

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

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

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

As $\gamma$ ranges over all field element in $F_q$, the rank values take on every value in the interval $[0, q-1]$. The ordering of elements of $F_q$ 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 3.9. 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 3.9: Orbiter primitive polynomials for fields \( \mathbb{F}_q \) with \( q \leq 1024 \)
Table 3.10: The field $\mathbb{F}_{16}$

$\mathbb{F}_{16}$:

\[
\begin{align*}
&\text{\texttt{\$(ORBITER\_PATH)\ orbiter.out\ -v\ -3\ \}}} \\
&\text{\texttt{\$\ -define\ -finite\_field\ -q\ -16\ -end\ \}}} \\
&\text{\texttt{\$\ -with\ -F\ -do\ -finite\_field\_activity\ -cheat\_sheet\_GF\ -end}}
\end{align*}
\]

creates a cheat sheet for $\mathbb{F}_{16}$. This command produces Table 3.10.

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 3.11
<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 3.11: 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><code>-RREF</code></td>
<td><code>m n L</code></td>
<td>Compute the RREF of the $m \times n$ matrix $L$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td><code>-nullspace</code></td>
<td><code>m n L</code></td>
<td>Compute a basis for the right nullspace of the $m \times n$ matrix $L$</td>
</tr>
<tr>
<td><code>-normalize_from_the_right</code></td>
<td></td>
<td>Normalizes the result of <code>-RREF</code> or <code>nullspace</code> from the right</td>
</tr>
<tr>
<td><code>-normalize_from_the_left</code></td>
<td></td>
<td>Normalizes the result of <code>-RREF</code> or <code>nullspace</code> from the left</td>
</tr>
<tr>
<td><code>-eigenstuff</code></td>
<td><code>d M</code></td>
<td>Computes the eigenvalues and eigenvectors of the given $d \times d$ matrix $M$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td><code>-all_rational_normal_forms</code></td>
<td><code>d</code></td>
<td>Produces a report of all rational normal forms of endomorphisms of $\mathbb{F}_q^d$</td>
</tr>
</tbody>
</table>

Table 3.12: Finite Field Activities for Linear Algebra

### 3.5 Linear Algebra Over Finite Fields

In Table 3.12, some finite field activities regarding linear algebra are shown. For instance, the command

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

computes the RREF form of the matrix

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

over $\mathbb{F}_2$. The output is the matrix

$$
\begin{bmatrix}
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1
\end{bmatrix}
$$
The \(-\text{RREF}\) 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 \(-\text{RREF}\) command:

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

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

A matrix over the field \(\mathbb{F}_7\):

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

55
Position \((i,j) = (0,0)\), found pivot in column 0

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

After making pivot 1:

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

After elimination below pivot:

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

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

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

After making pivot 1:
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 \((i,j) = (2,2)\), found pivot in column 2

After making pivot 1:

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

After elimination below pivot:

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

After elimination below pivot:

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

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

After making pivot 1:

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

After elimination below pivot:

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

Position \((i, j) = (4, 4)\), found pivot in column 4

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 1 & 3 & 4 & 6 & 4 & 0 & 0 \\
0 & 0 & 0 & 1 & 6 & 4 & 6 & 1 & 4 & 3 & 6 & 0 \\
0 & 0 & 0 & 3 & 2 & 6 & 1 & 3 & 6 & 3 & 1 & 0 \\
0 & 0 & 0 & 1 & 4 & 3 & 4 & 6 & 6 & 4 & 0 & 1 \\
0 & 0 & 0 & 3 & 1 & 4 & 3 & 6 & 3 & 1 & 0 & 1 \\
\end{bmatrix}
\]
After making pivot 1:

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

After elimination below pivot:

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

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

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

After making pivot 1:
After elimination below pivot:

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

After elimination below pivot:

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

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

After making pivot 1:

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

After elimination below pivot:

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

Did not find pivot. The rank of the matrix is 7.

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

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

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

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

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

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

After elimination above pivot 3 in position (3,3):
\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 5 & 1 & 5 & 1 & 2 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 6 & 5 & 3 & 3 & 5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 6 & 6 & 1 & 6 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 6 & 3 & 5 & 5 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination above pivot 2 in position (2,2):
\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 6 & 3 & 2 & 5 & 4 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 6 & 5 & 3 & 3 & 5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 6 & 6 & 1 & 6 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 6 & 3 & 5 & 5 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination above pivot 1 in position (1,1):
\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 6 & 3 & 2 & 5 & 4 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 6 & 5 & 3 & 3 & 5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 6 & 6 & 1 & 6 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 6 & 3 & 5 & 5 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

After elimination above pivot 0 in position (0,0):
\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 6 & 3 & 2 & 5 & 4 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 6 & 5 & 3 & 3 & 5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 6 & 6 & 1 & 6 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 6 & 3 & 5 & 5 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 6 & 6 & 6 & 6 \\
\end{bmatrix}
\]

The inverse matrix agrees with the output obtained in Section 3.2.

Another task is computing the nullspace of a matrix. The command
nullspace:

```bash
$ (ORBITER_PATH) orbiter.out -v 2 -
  -define F2 -finite_field -q 2 -
  -define v -vector -field F2 -format 2 -
  -dense: "1,1,1,0,1,1,0,1" -
  -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

```bash
eigenstuff:

$ (ORBITER_PATH) orbiter.out -v 6 -
  -define F -finite_field -q 5 -
  -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

```bash
classes_GL_3_2:

$ (ORBITER_PATH) orbiter.out -v 7 -
```

63
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:

- Class 0 / 6
  
  \[
  \begin{bmatrix}
  0 & 0 & 1 \\
  1 & 0 & 1 \\
  0 & 1 & 0
  \end{bmatrix},
  \begin{bmatrix}
  0 & 0 & 1 \\
  1 & 0 & 0 \\
  0 & 1 & 1
  \end{bmatrix},
  \begin{bmatrix}
  0 & 1 & 0 \\
  1 & 1 & 0 \\
  0 & 0 & 1
  \end{bmatrix},
  \begin{bmatrix}
  1 & 0 & 0 \\
  1 & 1 & 0 \\
  1 & 1 & 0
  \end{bmatrix},
  \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
  \end{bmatrix}
  \]

  centralizer order 7
  class size 24

- Class 1 / 6
  
  \[
  \begin{bmatrix}
  0 & 0 & 1 \\
  1 & 0 & 1 \\
  0 & 1 & 0
  \end{bmatrix}
  \]

  centralizer order 7
  class size 24

- Class 2 / 6
  
  \[
  \begin{bmatrix}
  0 & 0 & 1 \\
  1 & 0 & 0 \\
  0 & 1 & 1
  \end{bmatrix}
  \]
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
3.6 Advanced Topics in Finite Fields

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

First, in Tables 3.13-3.14, 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:
```

\[
\begin{array}{ccc}
1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 0 & 1 \\
\end{array}
\]

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.
<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 9.4.</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 $\mathbb{F}_{q^d}$ over $\mathbb{F}_q$.</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 3.13: 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 $x_j^i$ where $w_0, \ldots, w_{q-1}$ 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 3.14: Finite Field Activities (Part II)
Figure 3.8: The field reduction from $\mathbb{F}_{64}$ to $\mathbb{F}_8$

The output 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 $\mathbb{F}_{64}$ has many subfields. Figure 3.9 shows the field reduction from $\mathbb{F}_{64}$ to $\mathbb{F}_4$ (left) and from $\mathbb{F}_{64}$ to $\mathbb{F}_2$ (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 $\mathbb{F}_q$ can be computed using the `nth_roots` command, which is a finite field activity. The activity if applied to the field $\mathbb{F}_q$ over which the $n$-th roots are defined. The command constructs the field extension $\mathbb{F}_{q^m}$ 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 $\mathbb{F}_{q^m}$ 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 $\mathbb{F}_q$. The output lists the $n$-th roots first, generated by $\beta$. After that,
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 \( \mathbb{F}_8 \):

\[
\text{F}_8\text{Nth\_roots\_21:}
\]
\[
\text{
\begin{itemize}
\item $(\text{ORBITER\_PATH})\text{orbiter.out-v.3}$
\item $\text{-define-\text{F\_finite\_field\_q.8\_override\_polynomial\_11\_end}$}
\item $\text{-with-\text{F\_do\_finite\_field\_activity\_n\_roots\_21\_end}$}
\item $\text{pdflatex\_Nth\_roots\_q8\_n21.tex}$
\item $\text{open\_Nth\_roots\_q8\_n21.pdf}$
\end{itemize}
\]

The output is:

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

\[
\begin{align*}
\beta^0 &= 100000 = 1 \\
\beta^1 &= 000100 = \alpha^3 \\
\beta^2 &= 100001 = \alpha^5 + 1 \\
\beta^3 &= 111101 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1 \\
\beta^4 &= 011111 = \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha \\
\beta^5 &= 101010 = \alpha^4 + \alpha^2 + 1 \\
\beta^6 &= 110100 = \alpha^3 + \alpha + 1 \\
\beta^7 &= 100111 = \alpha^5 + \alpha^4 + \alpha^3 + 1 
\end{align*}
\]
\[ \beta^8 = 101101 = \alpha^5 + \alpha^3 + \alpha^2 + 1 \]
\[ \beta^9 = 011101 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha \]
\[ \beta^{10} = 011011 = \alpha^5 + \alpha^4 + \alpha^2 + \alpha \]
\[ \beta^{11} = 001011 = \alpha^5 + \alpha^4 + \alpha^2 \]
\[ \beta^{12} = 001001 = \alpha^5 + \alpha^2 \]
\[ \beta^{13} = 111000 = \alpha^2 + \alpha + 1 \]
\[ \beta^{14} = 000111 = \alpha^5 + \alpha^4 + \alpha^3 \]
\[ \beta^{15} = 101001 = \alpha^5 + \alpha^2 + 1 \]
\[ \beta^{16} = 111100 = \alpha^3 + \alpha^2 + \alpha + 1 \]
\[ \beta^{17} = 100110 = \alpha^4 + \alpha^3 + 1 \]
\[ \beta^{18} = 010100 = \alpha^3 + \alpha \]
\[ \beta^{19} = 100011 = \alpha^5 + \alpha^4 + 1 \]
\[ \beta^{20} = 001100 = \alpha^3 + \alpha^2 \]

Let \( \gamma \) be a primitive 7-th root in \( \text{GF}(64) \), so \( \gamma = \alpha^9 \).
\[ \gamma^0 = 100000 = 1 \]
\[ \gamma^1 = 111001 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1 \]
\[ \gamma^2 = 110100 = \alpha^3 + \alpha + 1 \]
\[ \gamma^3 = 011001 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha \]
\[ \gamma^4 = 001001 = \alpha^5 + \alpha^2 \]
\[ \gamma^5 = 101001 = \alpha^5 + \alpha^2 + 1 \]
\[ \gamma^6 = 010100 = \alpha^3 + \alpha \]

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

\{ 0 \}
\{ 1, 8 \}
\{ 2, 16 \}
\{ 3 \}
\{ 4, 11 \}
\{ 5, 19 \}
\{ 6 \}
\{ 7, 14 \}
\{ 9 \}
\{ 10, 17 \}
\{ 12 \}
\{ 13, 20 \}
\{ 15 \}
\{ 18 \}
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>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:

\[\text{F}_8\.\text{vandermonde}:\]

\[\text={[ORBITER\_PATH]}\text{orbiter.out} -v 3 \]\n
\[\text{define:F:finite_field:-q:8:end}\]

\[\text{with:F:do:finite_field_activity}\]

\[\text{Vandermonde_matrix}\]
The output is shown below. Again, the first matrix is $V = (x_i^j)$. The second matrix is $V^{-1}$:

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 4 & 5 & 7 & 3 & 6 & 1 \\
1 & 3 & 5 & 2 & 6 & 7 & 4 & 1 \\
1 & 4 & 7 & 6 & 2 & 5 & 3 & 1 \\
1 & 5 & 6 & 4 & 3 & 2 & 7 & 1 \\
1 & 6 & 3 & 7 & 5 & 4 & 2 & 1 \\
1 & 7 & 2 & 3 & 4 & 6 & 5 & 1
\end{bmatrix}
\quad
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 6 & 4 & 3 & 7 & 2 & 5 \\
0 & 1 & 3 & 7 & 5 & 2 & 4 & 6 \\
0 & 1 & 7 & 6 & 2 & 3 & 5 & 4 \\
0 & 1 & 5 & 2 & 6 & 4 & 7 & 3 \\
0 & 1 & 4 & 5 & 7 & 6 & 3 & 2 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

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 $\mathbb{F}_{1024}$:

```bash
F_1024_vandermonde:
$$(ORBITER\ PATH)\ orbiter.\ out\ -$\ v.\ 3\ \$$
$$(ORBITER\ PATH)\ orbiter.\ out\ -$\ v.\ 2\ \$$
```

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 `rm` command.

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

```bash
Berlekamp_matrix_2_3:
$$(ORBITER\ PATH)\ orbiter.\ out\ -$\ v.\ 2\ \$$
```

73
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:
```

```
$\text{ORBITER\ PATH}/\text{orbiter.out}\ -v\cdot 3\$
```

```
$\text{ORBITER\ PATH}/\text{orbiter.out}\ -v\cdot 3\$
```

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}\ -v\cdot 6\$
```

```
$\text{ORBITER\ PATH}/\text{orbiter.out}\ -v\cdot 6\$
```

```
$\text{ORBITER\ PATH}/\text{orbiter.out}\ -v\cdot 6\$
```

```
$\text{ORBITER\ PATH}/\text{orbiter.out}\ -v\cdot 6\$
```

```
$\text{ORBITER\ Path}/\text{orbiter.out}\ -v\cdot 6\$
```

```
$\text{ORBITER\ Path}/\text{orbiter.out}\ -v\cdot 6\$
```

```
$\text{ORBITER\ Path}/\text{orbiter.out}\ -v\cdot 6\$
```

74
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

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:

```
NTT_k4_q17.cpp:
  $(ORBITER_PATH)orbiter.out -v 3
  -define=finite_field=q17
  -with=finite_field_activity=NTT417
```

This produces a C++ file $\text{NTT}_k4_q17.cpp$. This file should be compiled and linked against the Orbiter library. Because of this, we use the following makefile variables.
The command

```
F_17_NTT_compile::NTT_k4_q17.cpp
  $(MY_CPP)::NTT_k4_q17.cpp $(CPPFLAGS)\
  $(LIB) $(LFLAGS) -o NTT_k4_q17.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.
Chapter 4

Geometry

4.1 Finite Projective Spaces

Orbiter can create the projective space 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 PG\((3, 2)\). The following command sequence first creates the finite field \(F_2\). The symbol \(F\) is used to store the field. After that, the projective space PG\((3, F)\) is created and stored in the symbol \(P\). No other commands are given:

\[
\text{PG}_3.2\_\text{easy}: \\
\text{\texttt{\$(ORBITER\_PATH)orbiter.out-v.3\·}} \\
\text{\texttt{\·}} \\
\text{\texttt{\·}} \\
\text{\texttt{-define\cdot F\cdot-finite_field\cdot-q\cdot2\cdot-end\·}} \\
\text{\texttt{-define\cdot P\cdot-projective_space\cdot3\cdotF\cdot-end}}
\]

This means that Orbiter offers indexing for the subspaces of 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 PG\((n, q)\) and list the various elements. The following command creates a cheat sheet for PG\((2, 4)\) using a finite field object:

\[
\text{PG}_2.4: \\
\text{\texttt{\$(ORBITER\_PATH)orbiter.out-v.2\·}} \\
\text{\texttt{\·}} \\
\text{\texttt{\·}} \\
\text{\texttt{-define\cdot F\cdot-finite_field\cdot-q\cdot4\cdot-end\·}} \\
\text{\texttt{-define\cdot P\cdot-projective_space\cdot2\cdotF\cdot-end\·}}
\]
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:
\[ P_0 = (1,0,0) = (1,0,0) \quad P_{11} = (2,1,1) = (\alpha,1,1) \]
\[ P_1 = (0,1,0) = (0,1,0) \quad P_{12} = (3,1,1) = (\alpha^2,1,1) \]
\[ P_2 = (0,0,1) = (0,0,1) \quad P_{13} = (0,2,1) = (0,\alpha,1) \]
\[ P_3 = (1,1,1) = (1,1,1) \quad P_{14} = (1,2,1) = (1,\alpha,1) \]
\[ P_4 = (1,1,0) = (1,1,0) \quad P_{15} = (2,2,1) = (\alpha,\alpha,1) \]
\[ P_5 = (2,1,0) = (\alpha,1,0) \quad P_{16} = (3,2,1) = (\alpha^2,\alpha,1) \]
\[ P_6 = (3,1,0) = (\alpha^2,1,0) \quad P_{17} = (0,3,1) = (0,\alpha^2,1) \]
\[ P_7 = (1,0,1) = (1,0,1) \quad P_{18} = (1,3,1) = (1,\alpha^2,1) \]
\[ P_8 = (2,0,1) = (\alpha,0,1) \quad P_{19} = (2,3,1) = (\alpha,\alpha^2,1) \]
\[ P_9 = (3,0,1) = (\alpha^2,0,1) \quad P_{20} = (3,3,1) = (\alpha^2,\alpha^2,1) \]
\[ P_{10} = (0,1,1) = (0,1,1) \]

Baer subgeometry:

\[ P_0 = (1,0,0) \quad P_2 = (0,0,1) \quad P_4 = (1,1,0) \quad P_{10} = (0,1,1) \]
\[ P_1 = (0,1,0) \quad P_3 = (1,1,1) \quad P_7 = (1,0,1) \]

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

\[ P_0 = (1,0,0) \quad P_6 = (1,2,0) \quad P_{12} = (1,2,2) \quad P_{18} = (1,3,1) \]
\[ P_1 = (0,1,0) \quad P_7 = (1,0,1) \quad P_{13} = (0,1,3) \quad P_{19} = (1,2,3) \]
\[ P_2 = (0,0,1) \quad P_8 = (1,0,3) \quad P_{14} = (1,2,1) \quad P_{20} = (1,1,2) \]
\[ P_3 = (1,1,1) \quad P_9 = (1,0,2) \quad P_{15} = (1,1,3) \]
\[ P_4 = (1,1,0) \quad P_{10} = (0,1,1) \quad P_{16} = (1,3,2) \]
\[ P_5 = (1,3,0) \quad P_{11} = (1,3,3) \quad P_{17} = (0,1,2) \]

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 \( \text{PG}(3, 2) \).

\[ \text{PG}_3 \text{.}2:\]

defined

\[
\text{\{ ORBITER
PATH\}orbiter.out
\cdot-\text{v.0}\\
\text{\{ ORBITER
PATH\}orbiter.out
\cdot-\text{v.0}\\
\text{-define \{ F \} \text{-finite
field\{-q.2\}-end}\\
\text{-define \{ P \} \text{-projective
space\{-3\}-F\{-end\}\\
\text{-with \{ P \} \text{-do \{-projective
space\-activity\{-cheat_sheet\\{-end
\text{pdflatex \{ PG_3.2.tex
\text{open \{ PG_3.2.pdf

\]}

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
Number of lines = 35
Number of lines on a point = 7
Number of points on a line = 3

**The points of PG(3, 2)**

PG(3, 2) has 15 points:

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

Normalized from the left:

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

**The lines of PG(3, 2)**

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

\[ L_0 = \begin{bmatrix} 0001 \\ 0100 \end{bmatrix} = \text{Pl}(1, 0, 0, 0, 0, 0) \]
\[ L_1 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} = \text{Pl}(1, 0, 1, 0, 0, 0) \]
\[ L_2 = \begin{bmatrix} 1000 \\ 0110 \end{bmatrix} = \text{Pl}(1, 0, 0, 0, 1, 0) \]
\[ L_3 = \begin{bmatrix} 1000 \\ 0111 \end{bmatrix} = \text{Pl}(1, 0, 1, 0, 1, 0) \]
\[ L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} = \text{Pl}(0, 0, 1, 0, 0, 0) \]
\[ L_5 = \begin{bmatrix} 1000 \\ 0011 \end{bmatrix} = \text{Pl}(0, 0, 1, 0, 1, 0) \]
\[ \vdots \]
\[ L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} = \text{Pl}(0, 1, 0, 0, 0, 0) \]
Lines sorted by Pluecker coordinates

\[ 0 = \mathbf{Pl}(1, 0, 0, 0, 0, 0) = L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \]

\[ 1 = \mathbf{Pl}(0, 1, 0, 0, 0, 0) = L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} \]

\[ 2 = \mathbf{Pl}(0, 0, 1, 0, 0, 0) = L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} \]

\[ 3 = \mathbf{Pl}(0, 0, 0, 1, 0, 0) = L_{30} = \begin{bmatrix} 0100 \\ 0001 \end{bmatrix} \]

\[ 4 = \mathbf{Pl}(0, 0, 0, 0, 1, 0) = L_6 = \begin{bmatrix} 1000 \\ 0001 \end{bmatrix} \]

\[ 5 = \mathbf{Pl}(0, 0, 0, 0, 0, 1) = L_{28} = \begin{bmatrix} 0100 \\ 0010 \end{bmatrix} \]

\[ \vdots \]

\[ 34 = \mathbf{Pl}(0, 1, 1, 1, 1, 1) = L_{26} = \begin{bmatrix} 1101 \\ 0011 \end{bmatrix} \]

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

\[ L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} = \mathbf{Pl}(1, 0, 0, 0, 0, 0) \]

\[ L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} = \mathbf{Pl}(0, 0, 1, 0, 0, 0) \]

\[ L_6 = \begin{bmatrix} 1000 \\ 0001 \end{bmatrix} = \mathbf{Pl}(0, 0, 0, 0, 1, 0) \]

\[ L_{28} = \begin{bmatrix} 0100 \\ 0010 \end{bmatrix} = \mathbf{Pl}(0, 0, 0, 0, 0, 1) \]

\[ L_{30} = \begin{bmatrix} 0100 \\ 0001 \end{bmatrix} = \mathbf{Pl}(0, 0, 0, 1, 0, 0) \]

\[ L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} = \mathbf{Pl}(0, 1, 0, 0, 0, 0) \]

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 \textsc{rank} and \textsc{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:
  $\text{\$(\text{ORBITER\_PATH})\text{\$orbiter.out:-v.2\$)\$}$
  $\text{\$do-define.F.-finite_field.-q.4.-end.$}$
  $\text{\$do-define.F.-do-\text{\$finite_field_activity.$}$}$
  $\text{\$do-rank_point_in_PG.2."3,3,1".-end.$}$
```
Algorithm 1 Rank

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

\[ a = \text{Rank}(x) \quad x = \text{Unrank}(a) \]

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></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>(1, 1, 0)</td>
</tr>
<tr>
<td>5</td>
<td>(1, 0, 1)</td>
</tr>
<tr>
<td>6</td>
<td>(0, 1, 1)</td>
</tr>
</tbody>
</table>

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

1: procedure Unrank(int : a, field : $F_q$, 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: $x_j \leftarrow \text{Unrank}(F_q, r)$
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:
  $ (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 \n  $ (ORBITER_PATH)orbiter.out -v.2 \n  $ -define:all_one:-vector:-repeat:1:73 -end \n  $ -draw_matrix \n  $ -input_csv_file:PG_n2_q8_incidence_matrix.csv \n  $ -box_width:20 -bit_depth:8 \n  $ -partition:3 \n  $ -all_one_all_one \n  $ -end
open: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 PG(2, q) deserve special attention. They are examples of a more general structure called projective planes. The 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 11.2.

The points in the desarguesian projective plane PG(2, q) have the coordinates P(x, y, z), with x, y, z ∈ Fq. 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 AG(2, q).

The command

```
PG_2.16:
▷ $(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:\
▷ ▷ -with:F:-do:-finite_field_activity:\
▷ ▷ ▷ -cheat_sheet_PG:2:-end
▷ pdflatex-PG_2_16.tex
▷ open-PG_2_16.pdf
```

produces the drawing of PG(2, 16) shown in Figure 4.3. The -nodes_empty command is used to suppress the drawing of the nodes. The -xin 20000 and -yin 20000 options double the input coordinate system (recall from Table 15.2 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 PG(2, 4)

```
P_2.4_with_decomposition:
▷ $(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:\
▷ ▷ ▷ 1:"0,1,0,0,1,2,1,1,0":"\n▷ ▷ ▷ "PG_2.4_singer":\n▷ ▷ ▷ -end
▷ #pdflatex-PG_2_4.tex
▷ #open-PG_2_4.pdf
```

The output is shown below:
Figure 4.3: The plane $\text{PG}(2, 16)$

Considering the cyclic group generated by

$$
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\omega & 1 & 1 \\
\end{bmatrix}
\cdot
\begin{bmatrix}
010 \\
001 \\
211 \\
\end{bmatrix}
= \begin{bmatrix}
010 \\
001 \\
211 \\
\end{bmatrix}
$$

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:

```
PG_2_4_incma_cyclic:
  $\$(ORBITER_PATH)orbiter.out-\v-4\$
  -list_arguments.
  -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 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:

```plaintext
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.

Figure 4.5: Tactical decomposition of the incidence matrix of PG(2, 4)
4.4 The Grassmannian

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

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \prod_{i=0}^{k-1} \frac{q^{n-i} - 1}{q^{k-i} - 1},$$

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 $N = \begin{bmatrix} n \\ k \end{bmatrix}_q$. 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 $\text{PG}(n,q)$ (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:
$$
\begin{verbatim}
\$ (ORBITER_PATH) orbiter.out \n\$ define F -finite_field -q 2 -end \n\$ with F -do -finite_field_activity \n\$ \$ cheat_sheet_Gr 3_2_2 -end
\$ pdflatex Gr_3_2_2.tex 
\$ open Gr_3_2_2.pdf
\end{verbatim}
```

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

$$
\begin{align*}
L_0 &= \begin{bmatrix} 100 \\ 010 \end{bmatrix} & L_3 &= \begin{bmatrix} 101 \\ 010 \end{bmatrix} & L_6 &= \begin{bmatrix} 010 \\ 001 \end{bmatrix} \\
L_1 &= \begin{bmatrix} 100 \\ 011 \end{bmatrix} & L_4 &= \begin{bmatrix} 101 \\ 011 \end{bmatrix} & \text{ } & \text{ } \\
L_2 &= \begin{bmatrix} 100 \\ 001 \end{bmatrix} & L_5 &= \begin{bmatrix} 110 \\ 001 \end{bmatrix} & \text{ } & \text{ }
\end{align*}
$$
Table 4.2: The partition ordering of monomials of degree 1, 2, 3 and 4 in a plane

<table>
<thead>
<tr>
<th>h</th>
<th>mon</th>
<th>vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$X_0$</td>
<td>(1, 0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>$X_1$</td>
<td>(0, 1, 0)</td>
</tr>
<tr>
<td>2</td>
<td>$X_2$</td>
<td>(0, 0, 1)</td>
</tr>
<tr>
<td>3</td>
<td>$X_0X_1$</td>
<td>(1, 1, 0)</td>
</tr>
<tr>
<td>4</td>
<td>$X_0X_2$</td>
<td>(1, 0, 1)</td>
</tr>
<tr>
<td>5</td>
<td>$X_1X_2$</td>
<td>(0, 1, 1)</td>
</tr>
</tbody>
</table>

<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)</td>
</tr>
<tr>
<td>1</td>
<td>$X_1^3$</td>
<td>(0, 3, 0)</td>
</tr>
<tr>
<td>2</td>
<td>$X_2^3$</td>
<td>(0, 0, 3)</td>
</tr>
<tr>
<td>3</td>
<td>$X_0^2X_1$</td>
<td>(2, 1, 0)</td>
</tr>
<tr>
<td>4</td>
<td>$X_0^2X_2$</td>
<td>(2, 0, 1)</td>
</tr>
<tr>
<td>5</td>
<td>$X_0X_1^2$</td>
<td>(1, 2, 0)</td>
</tr>
<tr>
<td>6</td>
<td>$X_1X_2^2$</td>
<td>(0, 2, 1)</td>
</tr>
<tr>
<td>7</td>
<td>$X_0X_2^2$</td>
<td>(1, 0, 2)</td>
</tr>
<tr>
<td>8</td>
<td>$X_1^2X_2$</td>
<td>(0, 1, 2)</td>
</tr>
<tr>
<td>9</td>
<td>$X_0X_1X_2$</td>
<td>(1, 1, 1)</td>
</tr>
<tr>
<td>10</td>
<td>$X_0^2X_1^2$</td>
<td>(2, 0, 2)</td>
</tr>
<tr>
<td>11</td>
<td>$X_0^2X_2^2$</td>
<td>(2, 0, 2)</td>
</tr>
<tr>
<td>12</td>
<td>$X_0^2X_1X_2$</td>
<td>(2, 1, 1)</td>
</tr>
<tr>
<td>13</td>
<td>$X_1^2X_2^2$</td>
<td>(1, 2, 1)</td>
</tr>
<tr>
<td>14</td>
<td>$X_0X_1X_2^2$</td>
<td>(1, 1, 2)</td>
</tr>
</tbody>
</table>

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 $V = v(p_1, \ldots, p_r)$. 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 (use option -monomial_type_PART) is grouping terms according to the partition that results from the degrees of the variables first, and then uses the lexicographic ordering as a tie breaker. The lexicographic ordering (use option -monomial_type_LEX) orders the monomials lexicographically. Table 4.2 shows the monomials in the partition ordering for degrees 2, 3 and 4 in a plane. Suppose we are interested in $\mathbb{F}_{11}$ rational points of the elliptic curve $y^2 = x^3 + x + 3$. We write $x^3 + 3 - y^2 + x = 0$. Homogenizing yields $X^3 + 3Z^3 - Y^2Z + XZ = 0$. Using $X_0, X_1, X_2$ instead of $X, Y, Z$ yields

$$X_0^3 + 3X_2^3 + 10X_1^2X_2 + X_0X_2^2 = 0.$$
Using the indexing of monomials from Table 4.2, we record the coefficient vector of the equation as sequence
\[(1, 0, 3, 0, 0, 0, 10, 1, 0, 0).\]

The Orbiter command

\[
\text{EC}\_11\_\text{EQUATION}="1,0,3,0,0,0,10,1,0,0"
\]

\[
\text{EC}\_11\_\text{txt}:
\begin{verbatim}
\$\{\text{ORBITER}\_\text{PATH}\}\text{orbiter.out:}-v.2:\text{\textbar}\text{\textbar}\text{\textbar}-\text{define-F:}-\text{finite}_\text{field}-\text{q.11:}-\text{end:}\text{\textbar}\text{\textbar}\text{\textbar}-\text{define-P:}-\text{projective}_\text{space}-2-F:}-\text{end:}\text{\textbar}\text{\textbar}\text{\textbar}-\text{define-EC:}-\text{geometric}_\text{object}-P:}\text{\textbar}\text{\textbar}\text{\textbar}-\text{projective}_\text{variety."EC}\_11":"EC\_\text{11}:3:$(EC\_11\_\text{EQUATION}):\text{\textbar}\text{\textbar}\text{\textbar}-\text{monomial}_\text{type}_\text{PART:}\text{\textbar}\text{\textbar}\text{\textbar}-\text{end:\textbar}\text{\textbar}\text{\textbar}-\text{with-EC:}-\text{do:}-\text{combinatorial}_\text{object}_\text{activity:}-\text{save:}\text{\textbar}\text{\textbar}\text{\textbar}-\text{end:}\text{\textbar}\end{verbatim}
\]

creates the algebraic set associated to the cubic curve \(y^2 = x^3 + x + 3\) in \(\text{PG}(2, 11)\). 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 \(\text{PG}(3, q)\). Based on
Table 4.3: The Orbiter ordering of monomials of degree 1, 2 and 3 in PG(3, q)
the partition ordering, the equation is coded as coefficient vector

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

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

\[
\text{HIRSCHFELD\_SURFACE\_EQUATION} = "0,0,0,0,0,1,0,1,0,0,1,0,1,0,0,0,0,0,0"
\]

Hirschfeld\_surface\_q4.txt:

\[
\texttt{\$} \text{(ORBITER\_PATH)} \text{orbiter.out}\text{-v.2}\text{-}\text{-define}\text{-finite\_field}\text{-q.4}\text{-end}\text{-}\text{-define}\text{-projective\_space}\text{-3}\text{-F}\text{-end}\text{-}\text{-define}\text{-H4}\text{-geometric\_object}\text{-P}\text{-}\text{-define}\text{-projective\_variety}\text{\"Hirschfeld\_surface\_q4\"}\text{-}\text{-define}\text{-monomial\_type}\text{-PART}\text{-}\text{-end}\text{-}\text{-with}\text{-H4}\text{-do}\text{-combinatorial\_object\_activity}\text{-save}\text{-}\text{-end}\text{-}
\]

A file called `Hirschfeld\_surface\_q4.txt` is created. The file contains the Orbiter ranks of the 45 points on the surface.

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 delPezzo surfaces

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

We define a delPezzo surface by giving the right hand side of the equation only. We wish to produce a report on the two surfaces over the field \( \mathbb{F}_{13} \). The following command can be used:

\[
\text{del\_Pezzo\_F13ab\_report}:
\]

\[
\texttt{\$} \text{(ORBITER\_PATH)} \text{orbiter.out}\text{-v.3}\text{-}\text{-define}\text{-finite\_field}\text{-q.13}\text{-end}\text{-}\text{-define}\text{-projective\_space}\text{-3}\text{-F}\text{-end}\text{-}\text{-define}\text{-f3}\text{-formula}\text{\"del\_Pezzo\_F13a\"}\text{\"x,y,z\"}\text{-}\text{-define}\text{-f4}\text{-formula}\text{\"del\_Pezzo\_F13b\"}\text{\"x,y,z\"}\text{-}\text{-define}\text{-del\_Pezzo13}\text{-collection}\text{\"f3,f4\"}\text{-}\text{-with}\text{-P}\text{-do}\text{-}\text{-projective\_space\_activity}\text{-}\text{-}
\]

97
The third argument after the \texttt{-formula} command specifies the managed variables, which are \( x, y, z \). The command \texttt{-collection} is used to group objects together. In this case, both surfaces are group together under new name. That way, we can issue the \texttt{-analyze\_del\_Pezzo\_surface} once, and it applies to both surfaces.
4.6 The Klein Quadric and the Plücker Map

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

The command

\[
\text{GR}_{4.2.2}: \quad \text{ pdflatex:Gr}_4.2.2\text{.tex} \quad \text{open:Gr}_4.2.2\text{.pdf}
\]

creates the elements of \( G_{4,2,2} \) and lists them together with their Plücker coordinates. The following output is shortened:

<table>
<thead>
<tr>
<th>Line</th>
<th>Plücker Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_0 )</td>
<td>[ \begin{bmatrix} 1000 \ 0100 \end{bmatrix} \text{ Pl}(1, 0, 0, 0, 0, 0) ]</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>[ \begin{bmatrix} 1000 \ 0110 \end{bmatrix} \text{ Pl}(1, 0, 1, 0, 0, 0) ]</td>
</tr>
<tr>
<td>( L_2 )</td>
<td>[ \begin{bmatrix} 1000 \ 0101 \end{bmatrix} \text{ Pl}(1, 0, 0, 0, 1, 0) ]</td>
</tr>
<tr>
<td>( \vdots )</td>
<td></td>
</tr>
<tr>
<td>( L_{34} )</td>
<td>[ \begin{bmatrix} 0010 \ 0001 \end{bmatrix} \text{ Pl}(0, 1, 0, 0, 0, 0) ]</td>
</tr>
</tbody>
</table>

The Plücker 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 Plücker map, this gives a second way to label the lines of \( PG(3, q) \). In the example of \( PG(3, 2) \) this yields the following list (output shortened):
0 = \mathbf{P_l}(1, 0, 0, 0, 0) = L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \quad 2 = \mathbf{P_l}(0, 0, 1, 0, 0, 0) = L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} \\
1 = \mathbf{P_l}(0, 1, 0, 0, 0) = L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} \quad : \\
34 = \mathbf{P_l}(0, 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>$\frac{n-1}{2} \sum_{i=0}^{n-q-1} X_{2i}X_{2i+1}$</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>$p(X_{n-1}, X_n) + \frac{n-1}{2} \sum_{i=0}^{n-q-1} X_{2i}X_{2i+1}$</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>$X_n^2 + \frac{n-1}{2} \sum_{i=0}^{n-q-1} X_{2i}X_{2i+1}$</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 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 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, $p(X, Y) = c_1X^2 + c_2XY + c_3Y^2 \in \mathbb{F}_q[X, Y]$ is irreducible over $\mathbb{F}_q$. 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

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

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: 
& $(ORBITER\_PATH)\text{orbiter.out-v.2}\backslash$ 
& $\triangleright$ `define:finite_field:-q.2:-end\backslash$ 
& $\triangleright$ `define:O:-orthogonal_space:1.4:F:-without_group:-end\backslash$ 
& $\triangleright$ `with:O:-do:-orthogonal_space_activity\backslash$ 
& $\triangleright$ `cheat_sheet_orthogonal:-end
` pdflatex:0.1_4.2_report.tex 
& `open:0.1_4.2_report.pdf
```
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-label_txt</td>
<td>L</td>
<td>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.</td>
</tr>
<tr>
<td>-label_tex</td>
<td>L</td>
<td>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.</td>
</tr>
<tr>
<td>-without_group</td>
<td></td>
<td>Do not create the orthogonal group.</td>
</tr>
</tbody>
</table>

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.

0.5.2 incidence_matrix.csv:

```bash
don $(ORBITER_PATH)orbiter.out -v.2 -define F:-finite_field:-q.2 -end -define 0:-orthogonal_space:0.5F:-without_group:-end -with 0:-do:-orthogonal_space_activity -end -export_point_line_incidence_matrix -end
$ $(ORBITER_PATH)orbiter.out -v.2 -define all_one_r: -vector:-repeat 1.15:-end -define all_one_c: -vector:-repeat 1.15:-end -draw matrix -input_csv_file:0.5.2 incidence_matrix.csv -box width:20 -bit_depth:8 -partition 2 -end open 0.5.2 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
```
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:
  $(ORBITER_PATH)orbiter.out -v\cdot2\$
  \$-define F\cdot-finite_field\cdot q\cdot2\cdot -end\$
  \$-define O\cdot-orthogonal_space \cdot 1\cdot 6\cdot F\cdot -without_group\cdot -end\$
  \$-with O\cdot-do\cdot orthogonal_space_activity\$
  \$-cheat_sheet_orthogonal\cdot -end$
  pdflatex\cdot 0\cdot 1\cdot 6\cdot 2\cdot report.tex$
  open\cdot 0\cdot 1\cdot 6\cdot 2\cdot 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 11.4.</td>
</tr>
</tbody>
</table>

Table 4.6: Activities related to orthogonal spaces

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

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

The number of points is 35 points:
- $P_0 = (1,0,0,0,0,0)$
- $P_1 = (0,1,0,0,0,0)$
- $P_2 = (0,0,1,0,0,0)$
- $P_3 = (1,0,1,0,0,0)$
- $P_4 = (0,1,1,0,0,0)$
- $P_5 = (0,0,0,1,0,0)$
- $P_6 = (1,0,0,1,0,0)$
\[ P_7 = (0, 1, 0, 1, 0, 0) \]
\[ P_8 = (1, 1, 1, 0, 0) \]
\[ P_9 = (0, 0, 0, 0, 1, 0) \]
\[ P_{10} = (1, 0, 0, 0, 1, 0) \]
\[ P_{11} = (0, 1, 0, 1, 0) \]
\[ P_{12} = (0, 0, 1, 0, 1, 0) \]
\[ P_{13} = (1, 0, 1, 0, 1, 0) \]
\[ P_{14} = (0, 1, 0, 1, 1, 0) \]
\[ P_{15} = (0, 0, 0, 1, 0, 0) \]
\[ P_{16} = (1, 0, 0, 1, 1, 0) \]
\[ P_{17} = (0, 1, 0, 1, 0) \]
\[ P_{18} = (1, 1, 1, 1, 0) \]
\[ P_{19} = (0, 0, 0, 0, 0, 1) \]
\[ P_{20} = (1, 0, 0, 0, 0, 1) \]
\[ P_{21} = (0, 1, 0, 0, 0, 1) \]
\[ P_{22} = (0, 0, 1, 0, 0, 1) \]
\[ P_{23} = (1, 0, 1, 0, 0, 1) \]
\[ P_{24} = (0, 1, 1, 0, 0, 1) \]
\[ P_{25} = (0, 0, 0, 1, 0, 1) \]
\[ P_{26} = (1, 0, 0, 1, 0, 1) \]
\[ P_{27} = (0, 1, 0, 1, 0, 1) \]
\[ P_{28} = (1, 1, 1, 1, 0, 1) \]
\[ P_{29} = (1, 1, 0, 0, 1, 1) \]
\[ P_{30} = (1, 1, 1, 0, 1, 1) \]
\[ P_{31} = (1, 1, 0, 1, 1, 1) \]
\[ P_{32} = (0, 0, 1, 1, 1, 1) \]
\[ P_{33} = (1, 0, 1, 1, 1, 1) \]
\[ P_{34} = (0, 1, 1, 1, 1, 1) \]

The number of lines is 105

\[
L_0 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0
\end{bmatrix}
\{P_0, P_{32}, P_{33}\}
\]

\[
L_1 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0
\end{bmatrix}
\{P_1, P_{32}, P_{34}\}
\]

\[
L_{104} = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\{P_8, P_9, P_{18}\}
\]

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\`

105
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 $Q(5,64)$:

\begin{verbatim}
q64_report:
 ▶ $\$(ORBITER_PATH)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

\end{verbatim}

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:

\[
X_0X_1 + X_2X_3 + X_4X_5 = 0
\]

\begin{tabular}{c|cccccccc}
$\rightarrow$ & 16769025 & 109025800 & 532350 & 532350 & 130 & 4225 & 4225 \\
16511040 & 65 & 4160 & 0 & 0 & 0 & 0 & 0 \\
266175 & 0 & 4096 & 128 & 0 & 0 & 1 & 0 \\
266175 & 0 & 4096 & 0 & 128 & 0 & 0 & 1 \\
4225 & 3969 & 0 & 126 & 126 & 2 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 4225 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 4225 \\
\end{tabular}
The number of points is 17047617
Too many points to print.
The number of lines is 1108095105

To study BLT-sets in $Q(4, q)$, see Section 11.4.
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:
\(\text{\$\{ORBITER\_PATH\}\text{\orbiter\_out-\v\_2}}\)
\(\text{\=define F\_\text{\-finite\_field\_\-q\_4\_\-end}}\)
\(\text{\=with F\_\text{\-do\_\-finite\_field\_activity}}\)
\(\text{\=cheat\_sheet\_hermitian\_2\_\-end}\)
\(\text{pdflatex\_H\_2\_4.tex}\)
\(\text{open\_H\_2\_4.pdf}\)
```

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

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

$P_0 = (1, 1, 0) = 4$
$P_1 = (2, 1, 0) = 5$
$P_2 = (3, 1, 0) = 6$
$P_3 = (1, 0, 1) = 7$
$P_4 = (2, 0, 1) = 8$
$P_5 = (3, 0, 1) = 9$
$P_6 = (0, 1, 1) = 10$
$P_7 = (0, 2, 1) = 13$
$P_8 = (0, 3, 1) = 17$

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

The command

```
H_3_4:
\(\text{\$\{ORBITER\_PATH\}\text{\orbiter\_out-\v\_2}}\)
\(\text{\=define F\_\text{\-finite\_field\_\-q\_4\_\-end}}\)
\(\text{\=with F\_\text{\-do\_\-finite\_field\_activity}}\)
\(\text{\=cheat\_sheet\_hermitian\_3\_\-end}\)
\(\text{pdflatex\_H\_3\_4.tex}\)
\(\text{open\_H\_3\_4.pdf}\)
```

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

108
The Hermitian variety $H(3, 4)$ contains 45 points:

$$
\begin{align*}
P_0 &= (1, 1, 0, 0) = 5 & P_{23} &= (3, 3, 1, 1) = 52 \\
P_1 &= (2, 1, 0, 0) = 6 & P_{24} &= (0, 0, 1, 1) = 38 \\
P_2 &= (3, 1, 0, 0) = 7 & P_{25} &= (1, 1, 2, 1) = 58 \\
P_3 &= (1, 0, 1, 0) = 8 & P_{26} &= (2, 1, 2, 1) = 59 \\
P_4 &= (2, 0, 1, 0) = 9 & P_{27} &= (3, 1, 2, 1) = 60 \\
P_5 &= (3, 0, 1, 0) = 10 & P_{28} &= (1, 2, 2, 1) = 62 \\
P_6 &= (0, 1, 1, 0) = 11 & P_{29} &= (2, 2, 2, 1) = 63 \\
P_7 &= (0, 2, 1, 0) = 15 & P_{30} &= (3, 2, 2, 1) = 64 \\
P_8 &= (0, 3, 1, 0) = 19 & P_{31} &= (1, 3, 2, 1) = 66 \\
P_9 &= (1, 0, 0, 1) = 23 & P_{32} &= (2, 3, 2, 1) = 67 \\
P_{10} &= (2, 0, 0, 1) = 24 & P_{33} &= (3, 3, 2, 1) = 68 \\
P_{11} &= (3, 0, 0, 1) = 25 & P_{34} &= (0, 0, 2, 1) = 53 \\
P_{12} &= (0, 1, 0, 1) = 26 & P_{35} &= (1, 1, 3, 1) = 74 \\
P_{13} &= (0, 2, 0, 1) = 30 & P_{36} &= (2, 1, 3, 1) = 75 \\
P_{14} &= (0, 3, 0, 1) = 34 & P_{37} &= (3, 1, 3, 1) = 76 \\
P_{15} &= (1, 1, 1, 1) = 4 & P_{38} &= (1, 2, 3, 1) = 78 \\
P_{16} &= (2, 1, 1, 1) = 43 & P_{39} &= (2, 2, 3, 1) = 79 \\
P_{17} &= (3, 1, 1, 1) = 44 & P_{40} &= (3, 2, 3, 1) = 80 \\
P_{18} &= (1, 2, 1, 1) = 46 & P_{41} &= (1, 3, 3, 1) = 82 \\
P_{19} &= (2, 2, 1, 1) = 47 & P_{42} &= (2, 3, 3, 1) = 83 \\
P_{20} &= (3, 2, 1, 1) = 48 & P_{43} &= (3, 3, 3, 1) = 84 \\
P_{21} &= (1, 3, 1, 1) = 50 & P_{44} &= (0, 0, 3, 1) = 69 \\
P_{22} &= (2, 3, 1, 1) = 51 & \\
\end{align*}
$$

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 $\mathbb{F}_4$.  

109
4.9 Advanced Topics

In Tables 4.7-4.8 summarize the Orbiter projective space activities. These are commands associated with a projective space \( \text{PG}(n, q) \).

Table 4.10 lists Orbiter commands related to projective geometries which are not tied to a finite field activity.

Suppose we are looking for a projectivity of \( \text{PG}(3, 16) \) 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, we want to map

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

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

The command

\[
\text{trans:} \quad \$(\text{ORBITER\_PATH})\text{orbiter.out-\v\cdot5} -v \cdot 5 \cdot \\
\text{define:}\text{\cdot} \text{finite_field-q\cdot16-\end} \cdot funeral - \text{define:}\text{\cdot} \text{projective_space-3\cdotF-\end} \cdot funeral - \text{with:}\text{\cdot} \text{do} \cdot \\
\text{define:}\text{\cdot} \text{projective_space_activity} \cdot \\
\text{\cdot define:}\text{\cdot} \text{move_two_lines_in_hyperplane_stabilizer_text} \cdot \\
\text{\cdot \cdot \cdot define:}\text{\cdot} "1,0,0,0,0,0,0,1.""1,1,0,2,0,0,1,0" \cdot \\
\text{\cdot \cdot \cdot define:}\text{\cdot} "1,0,0,0,-0,0,0,1.""0,1,0,1,0,0,1,0" \cdot \\
\text{\cdot \cdot \cdot define:}\text{\cdot} -$end
\]

computes a projectivity which does 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 \( \delta^4 = \delta^3 + 1 \).
<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 9.</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 $PG(n,q)$.</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 $PG(n,q)$. 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>-classify_surfaces_through_arcs_and_two_lines</td>
<td></td>
<td>See Section 7.3.</td>
</tr>
<tr>
<td>-test_nb_Eckardt_points</td>
<td>nbE</td>
<td>See Section 7.3.</td>
</tr>
<tr>
<td>-classify_surfaces_through_arcs_and_trihedral_pairs</td>
<td></td>
<td>See Section 7.3.</td>
</tr>
<tr>
<td>-sweep</td>
<td>fname</td>
<td></td>
</tr>
<tr>
<td>-sweep_4</td>
<td>fname surface-descr</td>
<td></td>
</tr>
<tr>
<td>-sweep_4_27</td>
<td>fname surface-descr</td>
<td></td>
</tr>
<tr>
<td>-six_arcs_not_on_conic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-filter_by_nb_Eckardt_points</td>
<td>nbE</td>
<td></td>
</tr>
<tr>
<td>-surface_quartic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-surface_clebsch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-surface_codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-trihedral_control</td>
<td>poset-control</td>
<td></td>
</tr>
<tr>
<td>-trihedra2_control</td>
<td>poset-control</td>
<td></td>
</tr>
<tr>
<td>-control_six_arcs</td>
<td>poset-control</td>
<td></td>
</tr>
<tr>
<td>-make_gilbert_varshamov_code</td>
<td>n d</td>
<td>See Section 9.8.</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><code>k</code> control</td>
<td>See Section 11.1.</td>
</tr>
<tr>
<td><code>-classify_semifields</code></td>
<td>descr</td>
<td></td>
</tr>
<tr>
<td><code>-cheat_sheet</code></td>
<td></td>
<td>Produce a cheat sheet for PG(n,q)</td>
</tr>
<tr>
<td><code>-classify_quartic_curves_nauty</code></td>
<td>fname-mask <code>N</code></td>
<td>Classify quartic curves using Nauty.</td>
</tr>
<tr>
<td><code>-classify_quartic_curves_with_substructure</code></td>
<td>fname-mask <code>N</code> <code>k</code> <code>d</code> fname</td>
<td>Classify quartic curves using substructure algorithm.</td>
</tr>
<tr>
<td><code>-set_stabilizer</code></td>
<td><code>k</code> fname-mask <code>N</code> 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><code>sz d d_min s</code></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><code>sz d d_min s t T</code></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><code>sz d d_min s t T u U</code></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>ε</td>
<td>Creates a table of points on a specific quartic curve. Consecutive points are no more than ε apart.</td>
</tr>
<tr>
<td>-create_points_on_parabola</td>
<td>ε, a, b, c</td>
<td>Creates a table of points on the parabola $y = ax^2 + bx + c$. Consecutive points are no more than ε apart.</td>
</tr>
<tr>
<td>-smooth_curve</td>
<td>ε, N, b, t_{min}, t_{max}, function</td>
<td>Creates at least N points on a continuous curve given by “function”. Consecutive points are no more than ε apart. The function must be in terms of a parameter $t$. The values of $t$ are taken from the interval $[t_{min}, t_{max}]$.</td>
</tr>
<tr>
<td>-create_spread</td>
<td>description</td>
<td>Creates a spread according to the description. See Section 11.1.</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
4.10 Geometric Objects

Orbiter can create objects in projective space. To do so, define an object of type -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

\[
\text{elliptic\_curve\_b1\_c3\_q11.txt:}
\]

\[
\begin{align*}
&\text{define F:finite_field-q.11:-end}\backslash
&\text{define P:projective_space-2:F:-end}\backslash
&\text{define EC:geometric_object-P}\backslash
&\text{elliptic\_curve-1\_3}\backslash
&\text{end}\backslash
&\text{with EC:do:combinatorial\_object\_activity:-save}\backslash
&\text{end}
\end{align*}
\]

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

\[
\text{elliptic\_curve\_b1\_c3\_q11.txt.}
\]

The following command creates an elliptic quadric ovoid on PG(3,8):

\[
\text{elliptic\_quadric\_void\_q8:}
\]

\[
\begin{align*}
&\text{define F:finite_field-q.8:-end}\backslash
&\text{define P:projective_space-3:F:-end}\backslash
&\text{define O:geometric_object-P}\backslash
&\text{elliptic\_quadric\_void}\backslash
&\text{end}\backslash
&\text{with O:do:combinatorial\_object\_activity:-save}\backslash
&\text{end}
\end{align*}
\]

The next command creates the Suzuki-Tits ovoid in PG(3,8):

\[
\text{ovoid\_ST\_q8:}
\]

\[
\begin{align*}
&\text{define F:finite_field-q.8:-end}\backslash
&\text{define P:projective_space-3:F:-end}\backslash
&\text{define O:geometric_object-P}\backslash
&\text{end}
\end{align*}
\]
<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 Adelaide 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, \ldots } )</td>
</tr>
<tr>
<td>-elliptic_quadric_ ovoid</td>
<td></td>
<td>Create an elliptic quadric ovoid in ( \text{PG}(3,q) ).</td>
</tr>
<tr>
<td>-ovoid_ST</td>
<td></td>
<td>Create the Suzuki Tits ovoid in ( \text{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>( \epsilon )</td>
<td>Create the ( \mathcal{Q}^\epsilon(n,q) ) quadric</td>
</tr>
<tr>
<td>-hermitian</td>
<td></td>
<td>Create the Hermitian variety given by ( \sum_{i=0}^{n} X_i^{q+1} = 0 )</td>
</tr>
<tr>
<td>-cuspial_cubic</td>
<td></td>
<td>Create the cuspidal cubic ( (s^3, ts^2, t^3) ) in ( \text{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 ( \text{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_hyperoval</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><code>-unital_XXq_YZq_ZYq</code></td>
<td></td>
<td>Create the unital with equation $XX^q+YZ^q+ZY^q=0$</td>
</tr>
<tr>
<td><code>-desarguesian_line_spread_in_PG_3_q</code></td>
<td></td>
<td>Create the desarguesian line spread in PG(3,q) as a set of 2-subspaces</td>
</tr>
<tr>
<td><code>-Buekenhout_Metz</code></td>
<td></td>
<td>Create the Buekenhout Metz unital</td>
</tr>
<tr>
<td><code>-Uab a b</code></td>
<td></td>
<td>Create the Buekenhout Metz unital in the form of Barwick and Ebert [5]</td>
</tr>
<tr>
<td><code>-whole_space</code></td>
<td></td>
<td>Create the whole space</td>
</tr>
<tr>
<td><code>-hyperplane pt</code></td>
<td></td>
<td>Create the hyperplane given by dual coordinates associated with the given point</td>
</tr>
<tr>
<td><code>-segre Variety a b</code></td>
<td></td>
<td>Create the Segre variety</td>
</tr>
<tr>
<td><code>-Maruta_Hamada_arc</code></td>
<td></td>
<td>Create the Maruta Hamada arc</td>
</tr>
<tr>
<td><code>-projective Variety lab_asci lab_tex d coeffs</code></td>
<td></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><code>-intersection_of_zariski_open_sets l d n C_1 \ldots C_n</code></td>
<td></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><code>-projective_curve l r d C</code></td>
<td></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><code>-embedded_in_PG_4_q</code></td>
<td></td>
<td>Create the BLT-set with ranks in PG($n,q$) instead of orthogonal point ranks</td>
</tr>
<tr>
<td><code>-BLT_in_PG</code></td>
<td></td>
<td>Create the BLT-set with ranks in PG($n,q$) instead of orthogonal point ranks</td>
</tr>
<tr>
<td><code>-monomial_type_LEX</code></td>
<td></td>
<td>Select lexicographic ordering of coefficients in an algebraic equation.</td>
</tr>
<tr>
<td><code>-monomial_type_PART</code></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 PG(2, 17) and saves it to file. The equation is encoded using the ordering of quartic monomials from Table 4.2.

```
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:

```
Edge_curve_17_line_type:
  echo$(FILE_Q17)>edge_q17.csv
```

118
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 3.4. 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 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.

\[
\text{GEN} \_\text{C13}="1,2,3,4,5,6,7,8,9,10,11,12,0"
\#
\text{\textbullet\textbullet\textbullet}(0,1,2,3,4,5,6,7,8,9,10,11,12)
\]

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

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:

\begin{verbatim}
GENERATOR_C13_0:=\$
  "1,2,3,4,5,6,7,8,9,10,11,12,0"

C13:
  $(ORBITER_PATH)orbiter.out\ -v\ 2\ 
  -define gens\-vector\-file C13\_gens.csv\-end\ 
  -define G\-permutation\group\ 
  -bsgs C13\ "C\{13\}"\ .13\ .13\ .0\ .1\ gens\-end\ 
\end{verbatim}

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

\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
Level & Base pt & Orbit length & Subgroup order \\
\hline
0 & 0 & 13 & 13 \\
\hline
\end{tabular}
\end{center}

\textbf{Basic Orbit 0}

\begin{center}
\begin{tikzpicture}
  \foreach \i in {0,\ldots,12}
  \path[fill] (\i,0) circle (0.1cm);
  \foreach \i in {0,\ldots,12}
  \draw[fill=white] (\i,0) circle (0.05cm);
\end{tikzpicture}
\end{center}
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:

<table>
<thead>
<tr>
<th>Row</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot;0,1,2,3,4,5,6,7,8,9,10,11,12&quot;</td>
</tr>
<tr>
<td>1</td>
<td>&quot;1,2,3,4,5,6,7,8,9,10,11,12,0&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;2,3,4,5,6,7,8,9,10,11,12,0,1&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&quot;3,4,5,6,7,8,9,10,11,12,0,1,2&quot;</td>
</tr>
<tr>
<td>4</td>
<td>&quot;4,5,6,7,8,9,10,11,12,0,1,2,3&quot;</td>
</tr>
<tr>
<td>5</td>
<td>&quot;5,6,7,8,9,10,11,12,0,1,2,3,4&quot;</td>
</tr>
<tr>
<td>6</td>
<td>&quot;6,7,8,9,10,11,12,0,1,2,3,4,5&quot;</td>
</tr>
<tr>
<td>7</td>
<td>&quot;7,8,9,10,11,12,0,1,2,3,4,5,6&quot;</td>
</tr>
<tr>
<td>8</td>
<td>&quot;8,9,10,11,12,0,1,2,3,4,5,6,7&quot;</td>
</tr>
<tr>
<td>9</td>
<td>&quot;9,10,11,12,0,1,2,3,4,5,6,7,8&quot;</td>
</tr>
<tr>
<td>10</td>
<td>&quot;10,11,12,0,1,2,3,4,5,6,7,8,9&quot;</td>
</tr>
<tr>
<td>11</td>
<td>&quot;11,12,0,1,2,3,4,5,6,7,8,9,10&quot;</td>
</tr>
<tr>
<td>12</td>
<td>&quot;12,0,1,2,3,4,5,6,7,8,9,10,11&quot;</td>
</tr>
</tbody>
</table>

END

Let us look at a symmetric group. The following command creates Sym(4):

```
Symmetric_4:
  ▶ $(ORBITER_PATH)orbiter.out-v.10\n  ▶ ▶ -define G -permutation_group -symmetric_group_4 -end\n  ▶ ▶ -with G -do\n  ▶ ▶ -group_theoretic_activity\n  ▶ ▶ ▶ -export_orbiter\n  ▶ ▶ ▶ -end\n  ▶ ▶ -with G -do\n  ▶ ▶ -group_theoretic_activity\n  ▶ ▶ ▶ -report\n  ▶ ▶ ▶ -end\n  ▶ ▶ -with G -do\n  ▶ ▶ -group_theoretic_activity\n  ▶ ▶ ▶ -save_elements_csv "Symmetric4_elts.csv"\n  ▶ ▶ -end
 ▶ pdflatex Perm4_report.tex
 ▶ open Perm4_report.pdf
```

The report is 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>4</td>
<td>24</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Basic Orbit 0

Basic orbit 0 has size 4
0, 1, 2, 3

Basic Orbit 1
Basic orbit 1 has size 3
1, 2, 3

Basic Orbit 2

Basic orbit 2 has size 2
2, 3

For comparison, let us have a look at a linear group. Suppose we want to create $\text{PGL}(4,2)$ 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:\n  >   -define:F:-finite_field:-q:2:-end:\n  >   -define:G:-linear_group:-PGL:4:F:-end:\n  >   -with:G:-do:\n  >     -group_theoretic_activity:\n  >     > -report:\n  >     > -end\n  >     > -with:G:-do:\n  >     > -group_theoretic_activity:\n  >     > > -export_orbiter:\n  >     > -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 $\text{PGL}_4_2\_report\.tex$. The second activity exports the permutation representation in Orbiter makefile format. The file $\text{PGL}_4_2\_makefile$ is created:

```
PGL_4_2:
  > $(ORBITER_PATH)orbiter.out:-v.2:\n```

126
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 \texttt{PGL\_4\_2\_gens.csv}:

\begin{verbatim}
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
END
\end{verbatim}

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 $\text{Sym}(13)$.

\begin{verbatim}
C13_as_subgroup:
  $(\text{ORBITER\_PATH})\text{orbiter.out} -v 10\$
  $(\text{-define G -permutation_group})$
  $(\text{-symmetric_group 13})$
  $(\text{-subgroup_by_generators C13 13])$
  $(\text{-export_orbiter})$
  $(\text{-report})$
  $(\text{-save_elements_csv C13_elts.csv})$
#pdflatex Perm13_Subgroup_C13_13_report.tex
#open Perm13_Subgroup_C13_13_report.pdf
\end{verbatim}

The \texttt{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. [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 $V \simeq \mathbb{F}_q^n$ 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 $\text{GL}(n+1,q)/\mathbb{F}_q^\times$. 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

$$\Phi_0 : \mathbb{F}_q \to \mathbb{F}_q, \ x \mapsto x^p$$

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

$$(\Psi_M, \Phi_0^k) : \text{PG}(n,q) \to \text{PG}(n,q), \ P(x) \mapsto P(y), \ y = (x \cdot M)^{\Phi_0^k}$$

128
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

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

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

The affine group AGL(\( n, q \)) is the semidirect product of GL(\( n, q \)) with \( F_q^n \). The affine semilinear group AΓL(\( n, q \)) is the semidirect product of AGL(\( n, q \)) with Aut(\( F_q \)). The elements of AΓL(\( n, q \)) 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 AΓL(\( n, q \)) is

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

The inverse of an element is

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

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 \( F_q \) as described in two steps. In the first step, the field \( F_q \) is created as described in Sections 3.2 and 3.4. 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,
<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>$\mathfrak{S}r_1(V)$</td>
<td>$\frac{q^n-1}{q-1}$</td>
</tr>
<tr>
<td>Wreath product GL($d,q$) *Sym($n$)</td>
<td>$\mathfrak{S}r_1((\mathbb{F}_q^d)^\otimes n)$ 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>$\frac{q^n-1}{q-1}$</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>

Table 5.1: Basic actions

```latex
PGL_{4,2}:
\begin{verbatim}
▷ $(\text{ORBITER\_PATH})\text{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
▷ pdflatex PGL_4_2_report.tex
▷ open PGL_4_2_report.pdf
\end{verbatim}
```

creates the group PGL(4,2) acting on the 15 elements of $\mathfrak{S}r_1(\mathbb{F}_2^4)$. At first, the field $\mathbb{F}_2$ is created. Secondly, the group $G = \text{PGL}(3,2)$ 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.

**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>
<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>ΓL(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>ΣL(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>PΓL(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>PΣL(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>AΓL(n, q)</td>
</tr>
<tr>
<td>-ASL</td>
<td>n q</td>
<td>ASL(n, q)</td>
</tr>
<tr>
<td>-ASSL</td>
<td>n q</td>
<td>AΣL(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>ΓGO(n, q)</td>
</tr>
<tr>
<td>-PGGOp</td>
<td>n q</td>
<td>ΓGO⁺(n, q)</td>
</tr>
<tr>
<td>-PGGOm</td>
<td>n q</td>
<td>ΓGO⁻(n, q)</td>
</tr>
<tr>
<td>-GL_{d,q}_{wr} Sym_n</td>
<td>d q n</td>
<td>GL(d, q) ⋊ Sym(n)</td>
</tr>
</tbody>
</table>

Table 5.2: Basic types of Orbiter matrix groups
Strong generators for a group of order 20160:
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0
\end{bmatrix},
\begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix},
\begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

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 \( \mathbb{F}_2 \)

\[ Z_i = \log_\alpha(1 + \alpha^i) \]

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

\[
\begin{array}{ccc}
  + & 0 & 1 \\
  0 & 0 & 1 \\
  1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{cc}
  \cdot & 1 \\
  1 & 1 \\
\end{array}
\]

\( 1^0 \equiv 1 \)
\( 1^1 \equiv 1 \)

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:

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

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,
1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,1,
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>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

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

Basic Orbit 2

Basic orbit 2 has size 12
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

PGO_5_2:
▷ $(ORBITER_PATH)orbiter.out -v.2\$
▷ ▷ -define:F:-finite_field:-q:2:-end:\$
▷ ▷ -define:G:-linear_group:-PGO:5:F:-end:\$
▷ ▷ -with:G-do:\$
▷ ▷ -group_theoretic_activity:\$
▷ ▷ -report:\$
▷ ▷ -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:

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

Group action $\text{PGO}(5,2)$ 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, 1, 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 $\mathbb{F}_2$

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

<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>

$\cdot$

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

$1^0 \equiv 1$

$1^1 \equiv 1$

**Base and Stabilizer Chain**

Group order 720

$tl=15, 8, 3, 1, 1, 2,$

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

Strong generators for a group of order 720:

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

1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,1,0,0,1,1,
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,
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,
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,
1,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,
1,0,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,1,1,0,0,0,0,1,

**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>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

Basic Orbit 5

Basic orbit 5 has size 2

5, 12

GAP export:

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

Magma export:

Compact form:

Generators in compact permutation form are:
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
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:
```

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 \\n  pdflatex PGO_7_2_report.tex \\n  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>$f\ l$</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>$l\ o\ n\ s_1\ ...\ s_n$</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:
  $\$(ORBITER_PATH)orbiter.out\-v\-3\$
  $\\\n  $\>
  $\>
  \-define\G\-linear\_group\-PGL\-7\-11\-Janko1\-end\$
  $\>
  $\>
  \-with\G\-do\$
  $\>
  $\>
  \-group\theoretic\_activity\$
  $\>
  $\>
  \-report\$
  $\>
  \-end$
  $\$
  pdflatex PGL_7_11_Subgroup_Janko1_report.tex
  $\$
  open PGL_7_11_Subgroup_Janko1_report.pdf
```
creates the first Janko group as a subgroup of 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 GL(d, q) is a generator for a subgroup of order \( q^d - 1 \). It induces an element of order \( \frac{d^2 - 1}{q^d - 1} \) on the associated projective geometry 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 PG(2, 11), extended by a group of order 3 that arises from the field extension \( \mathbb{F}_{11}^3 \) 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 SL(2, 3). The Orbiter command

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

145
-subgroup_by_generators: "quaternion": "8": 3:
    
    "1, 1, 1, 2, 2, 1, 1, 1, 0, 2, 1, 0":
    
    -end:\
    
    -with G:-do:\
    
    -group_theoretic_activity:\
    
    -print_elements_tex:\
    
    -group_table:\
    
    -report:\
    
    -end

pdflatex GL_2_3_Subgroup_quaternion_8_elements.tex
    
open GL_2_3_Subgroup_quaternion_8_elements.pdf

pdflatex GL_2_3_Subgroup_quaternion_8_report.tex
    
open GL_2_3_Subgroup_quaternion_8_report.pdf

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

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)
The group table is created as csv file:

<table>
<thead>
<tr>
<th>Row, C0, C1, C2, C3, C4, C5, C6, C7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0, 1, 2, 3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>1, 1, 2, 3, 0, 5, 6, 7, 4</td>
</tr>
<tr>
<td>2, 2, 3, 0, 1, 6, 7, 4, 5</td>
</tr>
<tr>
<td>3, 3, 0, 1, 2, 7, 4, 5, 6</td>
</tr>
<tr>
<td>4, 4, 7, 6, 5, 2, 1, 0, 3</td>
</tr>
<tr>
<td>5, 5, 4, 7, 6, 3, 2, 1, 0</td>
</tr>
<tr>
<td>6, 6, 5, 4, 7, 0, 3, 2, 1</td>
</tr>
<tr>
<td>7, 7, 6, 5, 4, 1, 0, 3, 2</td>
</tr>
</tbody>
</table>

The group of the cube can be created over the field \( \mathbb{F}_3 \):

cube_group:

```
$ (ORBITER_PATH) orbiter.out -v 3 \
```
The tetrahedral subgroup can be created as well:

```
tetra_group:
  $(ORBITER_PATH)orbiter.out -v -3
  -define G -linear_group -GL 3 3
  -subgroup_by_generators "tetra" "12" 2
  "0,1,0,0,0,1,0,0,0,1,2,0,0,2,0,0,1"
  -end
  -with G -do
  -group_theoretic_activity
  -print_elements.tex
  -report
  -end
```

```
pdflatex GL 3 3 Subgroup tetra 12 report.tex
open GL 3 3 Subgroup tetra 12 report.pdf
pdflatex GL 3 3 Subgroup tetra 12 elements.tex
open GL 3 3 Subgroup tetra 12 elements.pdf
```

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

```
GENERATORS_HESSE_GROUP="\n3,0,0,0,3,0,0,0,3,0,\n2,0,0,0,2,0,1,2,3,0,\n1,0,0,0,1,0,0,1,1,1,\n"
```
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:

```
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,\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"
```

Weyl_E8:
\begin{verbatim}
\$ (ORBITER_PATH)orbiter.out -v 3 \$
\$ -define gens -vector -dense \$
\$ -define $(GENERATORS_WEYL_GROUP_E8) \$
\$ -end \$
\$ -define G -linear_group -GL 8 3 \\
\$ -subgroup_by_generators \$
\$ -define "Weyl_E8" "696729600" 2 \$
\$ -define $(GENERATORS_WEYL_GROUP_E8) \$
\$ -end \$
\$ -with -G -do \$
\$ -group_theoretic_activity \$
\$ -report -do \$
\$ -end \$
\$ pdflatex GL_8_3_Subgroup_Weyl_E8_696729600_report.tex \$
\$ open GL_8_3_Subgroup_Weyl_E8_696729600_report.pdf \$
\end{verbatim}

A latex report is generated in the file $GL_8_3_Subgroup_Weyl_E8_696729600_report.tex$. 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 3.4, 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

$$X^2 + 2$$

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

```
-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

```
-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

```
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:\
  -group_theoretic_activity:\
  -report:\
  -end
  pdflatex-PGL_3_9_Subgroup_U_3_3_6048_report.tex
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 Co$_3$. 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.

```
CONWAY_GEN1="11011100100001010000
1110101111101000010111
000001000000100010101
111110011010001001110
010101000000010011101
00001000000100010101\n00100000000100010101\n00100001100000011111\n1110100100110100010111"
```

152
CONWAY_GEN2="0101000010111010111111
0110010001111101011111
0011010001111111011011
0001101110001011001111
1010010001000101000111
11001011001111010101
1000110100110101010101
0100110001010000001111
1100000010100110100010
0101110111000000100010
1011110101001111011001
1000101001010010100001
0001010001111100001111
0110010011010110100011
0101110111000001111101
1101011001110000100101
0101110100111100101111
1100101100110101010101
1101011001111111100011
0100101001001000100001
1100101100001001110011
0101110110010100001000
0000011011110011001111
0101110100111100000001
1100101100110101010110
0101110111000001111010
1101011001110000100101
0000011011110011001111
1101011001111111100011"

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~

153
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.

\begin{verbatim}
Ree_gen1="21,5,1,6,17,1,1,\cdot3,13,5,21,6,6,18,\cdot21,3,21,21,22,6,14,\cdot14,18,1,5,13,6,7,\cdot3,3,2,1,24,16,3,\cdot17,3,22,10,16,24,26,\cdot21,21,6,18,20,2,5"
Ree_gen2="16,3,11,5,16,22,20,\cdot24,6,18,24,7,1,26,\cdot9,23,17,18,23,20,13,\cdot9,7,2,15,17,5,11,\cdot3,3,6,21,4,24,16,\cdot25,8,6,24,21,12,7,\cdot24,15,2,13,11,14,24"
\end{verbatim}

Ree_27:
\begin{verbatim}
$\$(ORBITER_PATH)orbiter.out -v 6$
-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$
\end{verbatim}
<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) ♯ Sym(n) on the tensor space</td>
</tr>
<tr>
<td>-on_rank_one_tensors</td>
<td></td>
<td>induced action of GL(d, q) ♯ 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>

Table 5.4: Commands for creating new actions from old

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

T3_on_tensors:
- $(ORBITER\_PATH)orbi\_\text{t}er\_\text{\text{.out}}$-$v\_4$
- define$\_G$
- linear$\_\text{\text{group}}$-GL_d_q_{wr}Sym_n$2\_2\_3$-on_tensors-end
- with$\_G$-do
- group_theoretic_activity
- -report
- -end
- pdflatex$\_\text{GL2\_2\_wreath\_Sym3\_report}\_\text{.tex}$
- open$\_\text{GL2\_2\_wreath\_Sym3\_report}\_\text{.pdf}$

creates the group GL(2, 2) ♯ 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 $GL(2, 2) \wr Sym(3)$

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

The Action

Group action $GL(2, 2) \wr Sym(3)_{res255}$ 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.

\[
\begin{align*}
(\left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right], \left[ \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right]; id),
(\left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right], \left[ \begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array} \right]; id),
(\left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right], \left[ \begin{array}{cc} 0 & 1 \\ 1 & 1 \end{array} \right]; id),
(\left[ \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right], \left[ \begin{array}{cc} 1 & 1 \\ 1 & 0 \end{array} \right]; id),
(\left[ \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right], \left[ \begin{array}{cc} 1 & 1 \\ 1 & 0 \end{array} \right]; (1, 2)),
(\left[ \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right], \left[ \begin{array}{cc} 1 & 1 \\ 0 & 0 \end{array} \right]; (0, 1)),
\end{align*}
\]
It is also possible to restrict the action on all rank-one tensors, as the following example shows:

T3r1:

```bash
$ (ORBITER_PATH) orbiter.out -v 4 \
-def G \ 
-linear_group -GL_d q wr_Sym_n 2 2 3 -on_rank_one_tensors -end \ 
-with G -do \ 
-group_theoretic_activity \ 
-report \ 
-end \ 
pdf latex 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 \( \text{GL}(2, 2) \wr \text{Sym}(3) \)

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

The Action

Group action \( \text{GL}(2, 2) \wr \text{Sym}(3) \)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 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 $\Gamma L(2, 8)$ of $PG(1, 8)$ and act on $PG(2, 8)$:

```
PG_GL_2_8_on_conic:
▷ $(ORBITER_PATH)orboriter.out.-v.4.$
▷ ▷ -define::G:\n▷ ▷ -linear_group:-PGGL.2.8.-PGL2OnConic.-end:\n▷ ▷ -with::do:\n▷ ▷ -group_theoretic_activity:\n▷ ▷ ▷ -report:\n▷ ▷ -end
▷ pdflatex-PGL_2_8_OnConic_2_8_report.tex
▷ open-PGL_2_8_OnConic_2_8_report.pdf
```

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

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

The order of the group $\Gamma L(2, 8)OnConic(2, 8)$ is 1512.
The group acts on a set of size 73
Strong generators for a group of order 1512:

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

### The Action

Group action $\Gamma L(2, 8)OnConic$ of degree 73
We act on the following set:
The group is a matrix group.
The base action is on projective space \( \text{PG}(1, 8) \)
\( q = 8 \)
\( p = 2 \)
\( e = 3 \)
\( n = 1 \)
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 = 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>7</td>
<td>2</td>
<td>5</td>
<td>1</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>
Base and Stabilizer Chain

Group order 1512

\( tl=9, 8, 7, 3, \)

Stabilizer chain

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Level} & \text{Base pt} & \text{Orbit length} & \text{Subgroup order} \\
\hline
0 & 0 & 9 & 1512 \\
1 & 1 & 8 & 168 \\
2 & 2 & 7 & 21 \\
3 & 4 & 3 & 3 \\
\hline
\end{array}
\]
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
Basic Orbit 2

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

Basic Orbit 3

Basic orbit 3 has size 3
4, 6, 7
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 \texttt{-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 \\n  \$define G -linear_group -PGL:3:2 -end \\n  \$with G -do \\n  \$group_theoretic_activity \\n  \$save_elements_csv "PGL_3_2_elements.csv" \\n  \$end
\end{verbatim}

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

The command

\begin{verbatim}
PGL_3_4_singer:
  \$ (ORBITER_PATH) orbiter.out -v 5 \\n  \$define G -linear_group -PGL:3:4 -end \\n  \$with G -do \\n  \$group_theoretic_activity \\n  \$find_singer_cycle \\n  \$end
\end{verbatim}

finds all Singer cycles in 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 GL(2, 8). We multiply the elements coded by 0,1,2,3 and 4,5,6,7:
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<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>-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_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>see Table 6.2</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>Stores the group table as csv-file.</td>
</tr>
<tr>
<td>-group_table</td>
<td></td>
<td>Produce a latex report about the group.</td>
</tr>
<tr>
<td>-report</td>
<td></td>
<td>Include Sylow subgroups in the report (requires -report).</td>
</tr>
<tr>
<td>-sylow</td>
<td></td>
<td>Produce a latex report about the group.</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 $g_ig_j$, $1 \leq i, j \leq n$.</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>-classify_arcs</td>
<td>description</td>
<td>Classify arcs in geometries. See Section 6.5.</td>
</tr>
<tr>
<td>-linear_codes</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 $\text{PGL}(r,q)$. For each $n \leq n_{\text{max}}$, the $[n,k,\geq d]$ codes are classified with $n-k=r$. See Section 9.</td>
</tr>
<tr>
<td>-tensor_classify</td>
<td>$d$</td>
<td>Classifies tensors of tensor-rank at most $d$.</td>
</tr>
<tr>
<td>-tensor_permutations</td>
<td></td>
<td>Computes the permutation representation of generators of wreath product.</td>
</tr>
<tr>
<td>-reverse_iso morphism_exterior_square</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>-classify_cubic_curves</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)

```
GL_2.8_multiply:
  $\$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_\text{-v.5.}\$
  $\$ -\text{define}\_\text{G}\_\text{-linear}\_\text{group}\_\text{-GL.2.8}\_\text{-end}\$
  $\$ -\text{with}\_\text{G}\_\text{-do}\$
  $\$ -\text{group}\_\text{theoretic}\_\text{activity}\$
  $\$ -\text{multiply}\_\text{"0,1,2,3","4,5,6,7\"}\$
  $\$ -\text{end}\$
  $\$ \text{pdflatex}\_\text{GL.2.8}\_\text{mult}.\text{tex}$
  $\$ \text{open}\_\text{GL.2.8}\_\text{mult}.\text{pdf}$
```

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 $7 \equiv 0 \mod 7$ is:

\[
GL_2(\mathbb{F}_7) \text{ multiply:}
\]

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

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

We can compute the inverse of a group element:

\[
GL_2(\mathbb{F}_7) \text{ inv:}
\]

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

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

```
GL_2_7_power:
▷ $(ORBITER_PATH)orbiter.out -v -5 -
▷ ▷ -define G -linear_group -GL 2 7 -end -
▷ ▷ -with G -do -
▷ ▷ -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
```

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 \n  $ $(MAGMA_PATH) / magma-PGGL_2_4_classes.magma
```

169
computes the classes of elements in $\text{PGL}(2,4)$ using Orbiter-Magma-Orbiter. The first Orbiter command produces the file `PGGL_2_4_classes.magma`. The magma command reads this file and produces the file `PGGL_2_4_classes_out.txt`. The second Orbiter command reads the file `PGGL_2_4_classes_out.txt` and produces the latex report `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}

defined as $c_1 = (0,1,1,0,1)$ of order 2 and with 3 fixed points, 0,1,1,0,1.

The normalizer is generated by:

Strong generators for a group of order 12:

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

The command sequence

```
PGGL_2_4_cent_2A:
  $(ORBITER_PATH)orbiter.out.-v.3.\$
  > -define:G:\$
  > -linear_group:-PGGL-2-4.-end.\$
  > -with:G.-do.\$
  > -group_theoretic_activity.\$
  >  > -centralizer_of_element."2A"."1,0,0,1,1".\$
```

170
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:

<table>
<thead>
<tr>
<th>Strong generators for a group of order 40320:</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
\end{bmatrix}, \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
\end{bmatrix} \\
\] |
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 \(\text{PGL}(4, 5)\) of order 31. The command

\begin{verbatim}
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, 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
\end{verbatim}

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:

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

1,0,0,0,0,4,0,0,0,0,4,0,0,0,0,4,
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:
  > $(ORBITER_PATH)orbiter.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 $\text{PGL}(2,9)_{\text{Subgroup}Z22_{\text{order}4}}$ 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, and hence does not scale well. 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 14.2.

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>$f l$</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>$f$</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>$s$</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>$f$name-$C$ $f$name-$T$</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 $f$name-$C$ and $f$name-$T$ are the files containing the elements of the class and the transporter, respectively.</td>
</tr>
</tbody>
</table>

Table 6.1: Basic Orbit algorithms
Figure 6.1: The Schreier tree for the action on rank-one tensors

orbits_cubic_curves_q2:

```
$ (ORBITER_PATH)orbiter.out -v 4 \
define G = linear_group -PGL 3 2 -end \
with G do \
group_theoretic_activity \
orbits_on_polynomials 3 \
end \
pdflatex poly_orbits_d3_n3_q2.tex \
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:
$ (ORBITER_PATH) orbiter.out -v 4 \
define G = \
linear_group -PGGL 2 8 -PGL2nConic -end \
with G do \
```

177
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 : orbit length : 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)

62 : 16 = (6, 0, 1)

Orbits of length 1:
Orbit 1: (1)

Stabilizer of orbit representative 1:
Strong generators for a group of order 1512:

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

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:
\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
\gamma^6 & 0 \\
0 & 1 \\
\end{bmatrix},
\begin{bmatrix}
\gamma^4 & 0 \\
0 & 1 \\
\end{bmatrix}
\]

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:
Generator 0 / 4 is:

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

Generator 1 / 4 is:

\[
\begin{bmatrix}
1 & 0 \\
\gamma^5 & 1
\end{bmatrix}
\]

Generator 2 / 4 is:

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

Generator 3 / 4 is:

\[
\begin{bmatrix}
1 & 0 \\
\gamma & 1
\end{bmatrix}
\]
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 $\mathcal{P}$ if for all $x, y \in \mathcal{P}$ 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 $\mathcal{P}$ 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 one 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 $V(3, 2) = \mathbb{F}_2^3$. The basis elements are listed, using the enumerator for elements on the projective geometry $\text{PG}(2, 2)$ 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 $n$. 

Figure 6.3: Subspace lattice of $V(3, 2)$
<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 15.2.</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</td>
<td>$L$</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</td>
<td>$n \ a \ b$</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:

\[
x, y \in \mathcal{P}(X) \text{ and } x \leq y \Rightarrow (y \in \mathcal{P} \rightarrow x \in \mathcal{P}).
\]

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 \( PG(3, 2) \). The following command computes the orbits of the group \( G \) generated by it:

\[
\text{PGL}_3\_2\_singer:}
\]

\[
\text{\qquad \text{\$\{ORBITER\_PATH\}orbiter.out\_v.3\_}}
\]

\[
\text{\qquad \qquad \text{-orbiter_path\$\{ORBITER\_PATH\}\_}}
\]

\[
\text{\qquad \qquad \text{-define\_G::linear_group\_-PGL\_3\_2\_-singer\_1\_-end\_}}
\]

\[
\text{\qquad \qquad \text{-with\_G::do\_}}
\]

\[
\text{\qquad \qquad \text{-group\_theoretic\_activity\_}}
\]

\[
\text{\qquad \qquad \text{-poset\_classification\_control\_}}
\]

\[
\text{\qquad \qquad \text{-problem\_label\:_PGL\_3\_2\_singer\_1\_W\_-depth\_7\_}}
\]

\[
\text{\qquad \qquad \text{-draw\_poset\_}}
\]

\[
\text{\qquad \qquad \text{-report\_}}
\]

\[
\text{\qquad \qquad \text{-end\_}}
\]

\[
\text{\qquad \text{-end\_}}
\]

\[
\text{\qquad \text{-orbits\_on\_subsets\_7\_}}
\]

\[
\text{\qquad \text{-report\_}}
\]

\[
\text{\qquad \text{-end\_}}
\]

\[
\text{\qquad pdflatex\:PGL\_3\_2\_singer\_1\_poset.tex}
\]

\[
\text{\quad open\:PGL\_3\_2\_singer\_1\_poset.pdf}
\]

Orbiter can compute orbits of groups acting in various different actions. For instance, the following example computes the orbits of \( PGL(4, 7) \) acting on the lines of \( PG(3, 7) \). All orbits on subsets of lines of size at most 3 are classified:

\[
\text{PGL\_3\_7\_lines:}
\]

\[
\text{\qquad \text{\$\{ORBITER\_PATH\}orbiter.out\_v.7\_}}
\]

\[
\text{\qquad \qquad \text{-orbiter_path\$\{ORBITER\_PATH\}\_}}
\]

\[
\text{\qquad \qquad \text{-define\_F::finite_field\_-q.7\_-end\_}}
\]

\[
\text{\qquad \qquad \text{-define\_G::linear_group\_-PGL\_4\_F\_-on_k_subspaces\_2\_-end\_}}
\]

\[
\text{\qquad \qquad \text{-with\_G::do\_}}
\]

\[
\text{\qquad \qquad \text{-group\_theoretic\_activity\_}}
\]
The following example computes the orbits of PGO(5, 2) on the power set lattice of points of $Q(4, 2)$:

```bash
PGO_5_2_on_subsets:
  $(ORBITER_PATH)orbiter.out--v.3\n  -orbiter_path$(ORBITER_PATH)\n  -define:F:-finite_field:-q.2:-end\n  -define:G:-linear_group:-PGO-5:F:-end\n  -with:G:-do\n  -group_theoretic_activity\n  -poset_classification_control\n  -problem_label:PGO_5_2\n  -depth:15\n  -report:-end\n  -draw_poset\n  -w\n  -end\n  -orbits_on_subsets:15\n  -report\n  -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:

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

The \(Q^+(3, 2)\) quadric given by the equation above consists of the nine points

\[ P_0, P_1, P_2, P_3, P_4, P_6, P_7, P_9, P_{10}. \]

The quadric is stabilized by the group \(\text{PGO}^+(4, 2)\) of order 72. The command

\[
\text{subspaces}_{\text{Op}}\_4\_2:
\]

\[
\begin{align*}
\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-\_v\_5\_\}} & \text{\texttt{\textasciitilde}} \\
\text{\texttt{\textasciitilde}} & \text{\texttt{-orbiter\_path\_\$\{ORBITER\_PATH\}\\_}} \\
\text{\texttt{-define\_G\_:linear\_group\_\_PGL\_4\_2\_orthogonal\_1\_end\_}} \\
\text{\texttt{-with\_G\_:do\_}}
\end{align*}
\]
Figure 6.6: Hasse-diagram for the orbits of the orthogonal group $\text{PGO}^+(4, 2)$ on subspaces of $\text{PG}(3, 2)$

\begin{verbatim}
\begin{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}

produces a classification of all subspaces of $\text{PG}(3, 2)$ under $\text{PGO}^+(4, 2)$. The option \texttt{-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_</td>
<td>set</td>
<td>The arc must not contain any of the given points.</td>
</tr>
<tr>
<td>point_set</td>
<td></td>
<td></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 9. 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:
> $(ORBITER_PATH)orbiter.out\-v\-4\-
> -orbiter_path$(ORBITER_PATH)\-
> -define\-finite_field\-q16\-end\-
> -define\-projective_space\-2\-F\-end\-
```
performs the classification of hyperovals in $\mathrm{PG}(2,16)$. 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

$\{0, 1, 2, 3, 52, 67, 89, 106, 126, 141, 159, 176, 184, 199, 220, 235, 245, 262\}_{144}$

$\begin{array}{ll}
0 : 0 = (1, 0, 0) & 10 : 159 = (14, 8, 1) \\
1 : 1 = (0, 1, 0) & 11 : 176 = (15, 9, 1) \\
2 : 2 = (0, 0, 1) & 12 : 184 = (7, 10, 1) \\
3 : 3 = (1, 1, 1) & 13 : 199 = (6, 11, 1) \\
4 : 52 = (3, 2, 1) & 14 : 220 = (11, 12, 1) \\
5 : 67 = (2, 3, 1) & 15 : 235 = (10, 13, 1) \\
6 : 89 = (8, 4, 1) & 16 : 245 = (4, 14, 1) \\
7 : 106 = (9, 5, 1) & 17 : 262 = (5, 15, 1) \\
8 : 126 = (13, 6, 1) & \\
9 : 141 = (12, 7, 1) & \\
\end{array}$

193
Strong generators for a group of order 144:

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\delta^4 & \delta^9 & 1
\end{bmatrix}_1, \quad
\begin{bmatrix}
1 & \delta^7 & \delta^{13} \\
\delta^8 & \delta^9 & \delta^{10} \\
\delta & \delta^6 & 1
\end{bmatrix}_3, \quad
\begin{bmatrix}
\delta^5 & \delta^5 & \delta^5 \\
\delta^5 & \delta^2 & \delta^{11} \\
\delta^5 & \delta^{14} & 1
\end{bmatrix}_0
\]

1,0,0,1,0,9,5,1,1,
1,7,6,14,5,10,2,15,1,3,
1,1,1,3,15,1,5,10,0,
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) \quad 10 : 156 = (11, 8, 1)
1 : 1 = (0, 1, 0) \quad 11 : 174 = (13, 9, 1)
2 : 2 = (0, 0, 1) \quad 12 : 186 = (9, 10, 1)
3 : 3 = (1, 1, 1) \quad 13 : 199 = (6, 11, 1)
4 : 52 = (3, 2, 1) \quad 14 : 217 = (8, 12, 1)
5 : 70 = (5, 3, 1) \quad 15 : 229 = (4, 13, 1)
6 : 83 = (2, 4, 1) \quad 16 : 256 = (15, 14, 1)
7 : 109 = (12, 5, 1) \quad 17 : 264 = (7, 15, 1)
8 : 127 = (14, 6, 1)
9 : 139 = (10, 7, 1)

Strong generators for a group of order 16320:

\[
\begin{bmatrix}
\delta^6 & 0 & 0 \\
0 & \delta^3 & 0 \\
0 & 0 & 1
\end{bmatrix}_2, \quad
\begin{bmatrix}
\delta^9 & 0 & 0 \\
0 & \delta^7 & 0 \\
0 & 0 & 1
\end{bmatrix}_1, \quad
\begin{bmatrix}
\delta^2 & 0 & 0 \\
0 & \delta^{11} & 0 \\
\delta^4 & \delta^7 & 1
\end{bmatrix}_3
\]

\[
\begin{bmatrix}
\delta^{10} & 0 & 0 \\
0 & \delta^3 & 0 \\
\delta & \delta^{11} & 1
\end{bmatrix}_3, \quad
\begin{bmatrix}
\delta & 0 & 0 \\
\delta^{12} & \delta^2 & \delta^5 \\
\delta^{14} & \delta^{10} & 1
\end{bmatrix}_1, \quad
\begin{bmatrix}
\delta^5 & 0 & 0 \\
\delta^5 & \delta^3 & \delta^6 \\
\delta^6 & \delta^8 & 1
\end{bmatrix}_0
\]

\[
\begin{bmatrix}
\delta^{12} & 1 & \delta^2 \\
\delta^4 & \delta^3 & \delta^7 \\
\delta^6 & \delta^3 & 1
\end{bmatrix}_2, \quad
\begin{bmatrix}
\delta^5 & \delta^3 & \delta^6 \\
\delta^{11} & \delta^6 & \delta^{10} \\
\delta^{10} & \delta^6 & 1
\end{bmatrix}_3
\]
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)orbiter.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:-W:-depth.6\n  ▶ ▶ ▶ ▶ ▶ ▶ -report:-end\n  ▶ ▶ ▶ ▶ ▶ -end\n  ▶ ▶ ▶ -target_size.6\n  ▶ ▶ ▶ -d.2\n  ▶ ▶ ▶ -conic_test\n  ▶ ▶ -end\n  ▶ -end
  pdflatex-nc_arcs_q16_d2.poset.tex
  open-nc_arcs_q16_d2.poset.pdf
```

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:
  ▶ $(ORBITER_PATH)orbiter.out-\v.4\n```

```
-orbiter_path:$(ORBITER_PATH)\n
-define F:finite_field:q32: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_q32_d2:-W:-depth:6\n
-draw_poset:-draw_options:-end\n
-report:-end\n
-end:\n
-target_size:6:\n
-test_nb_Eckardt_points:13\n
-conic_test:\n
-end\n
pdflatex nc_arcs_q32_d2_poset.tex
open 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:

\[\texttt{\$(ORBITER\_PATH)orbiter.out\ -v\ -define\_G:\ \ -define\_F:\ -finite\_field\ -q\_8\ -end\ -define\_P:\ -projective\_space\: 2\_F\ -end\ -with\_P\ -do\ -\ -projective\_space\_activity\ -classify\_cubic\_curves\ -q\_8\ -target\_size\_9\ -n\_3\ -d\_3\ -poset\_classification\_control\ -problem\_label\_cc\_8\ -W\ -depth\_9\ -draw\_options\ -radius\_200\ -embedded\ -end\ -recognize\"0,1,2,3,35,28\"\ -recognize\"1,2,3,51,28,61,46,71,40\"\ -draw\_poset\ -Kramer\_Mesner\_matrix\_6\_9\ -end\ -draw\_matrix\ -input\_csv\_file\_cc\_8\_KM\_6\_9.csv\ -box\_width\_50\ -bit\_depth\_8\ -end\ -pdf\_latex\_Cubic\_curves\_q8.tex\ -open\_Cubic\_curves\_q8.pdf\ -pdf\_latex\_cc\_8\_tree\_lvl\_9.tex\ -open\_cc\_8\_tree\_lvl\_9.pdf\]

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.

```
surface_4_0:
  ▶ $(\text{ORBITER\_PATH})\text{or} \text{biter.out}\cdot -v.3\cdot$
```
<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 $\text{PG}(3,q)$. 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 $X_3^3 - b^2(X_0^2 + X_1^2 + X_2^2)X_3 + \frac{b^3}{a}(a^2 + 1)X_0X_1X_2 = 0$. 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_abc0d</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 $A = a_1, \ldots, a_5$ in $\text{PG}(2,q)$ by means of the Clebsch map. Each of the $a_i$ is the rank of a point in $\text{PG}(2,q)$. Use the trihedral pair algorithm. Here, $A$ is a comma-separated string containing the numerical ranks of the $P_i$ in $\text{PG}(2,q)$.</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 $\text{PG}(2,q)$ by means of the Clebsch map. Each of the $a_i$ is the rank of a point in $\text{PG}(2,q)$. Use the two-lines algorithm. Here, $A$ is a comma-separated string containing the numerical ranks of the $P_i$ in $\text{PG}(3,q)$ and $L$ is a comma-separated string of the numerical ranks of two lines in $\text{PG}(3,q)$. 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)
Eckardt_13:
\[
\begin{align*}
pdflatex\text{-}surface\text{-}catalogue\text{-}q4\text{-}iso0\text{-}report\text{-}tex & \setcounter{equation}{13} \\
pdflatex\text{-}surface\text{-}catalogue\text{-}q4\text{-}iso0\text{-}report.pdf & \\
pdflatex\text{-}surface\text{-}catalogue\text{-}q4\text{-}iso0\text{-}with\text{-}group\text{-}tex & \\
opendefinition\text{-}projective\text{-}space\text{-}activity\text{-}end & \\
opendefinition\text{-}family\text{-}Eckardt\text{-}3\text{-}1\text{-}end & \\
opendefinition\text{-}surface\text{-}S\text{-}q13\text{-}do & \\
\text{-}report & \\
\text{-}report\text{-}with\text{-}group & \\
\text{-}end & \\
pdflatex\text{-}surface\text{-}family\text{-}Eckardt\text{-}q13\text{-}a3\text{-}b1\text{-}with\text{-}group\text{-}tex & \\
opendefinition\text{-}projective\text{-}space\text{-}activity\text{-}end & \\
opendefinition\text{-}surface\text{-}S\text{-}q13\text{-}do & \\
\text{-}cubic\text{-}surface\text{-}activity & \\
\text{-}end & \\
opdfoutput & \\
pdflatex\text{-}surface\text{-}family\text{-}Eckardt\text{-}q13\text{-}a3\text{-}b1\text{-}with\text{-}group.pdf & \\
\end{align*}
\]
creates the member of the Eckardt family described in [12] with parameters \((a, b) = (3, 1)\) over the field \(\mathbb{F}_{13}\).

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:

\[
F_{abcd\_eqn}=\begin{align*}
-(a*b*c-a*b*d-a*c*d+a*b*c*d+a*c*d-a*b*c)*&(b-d)*X0*X0*X2 \\
+(a*b*c-a*b*d-a*c*d+a*b*c*d+a*c*d-a*b*c)*&(a+b-c-d)*X0*X1*X2 \\
+(a*c-c-a*d-a*c+c+b*c*c+a*d-a*b*c)*&(b-d)*X0*X1*X3 \\
-(a*d-b*c)*(a*b*c-a*b*d-a*c+d+b*c*d+a*d-a*b*c)*&(a-c)*X1*X1*X2 \\
-(a-c)*(a*b*c-a*b*d-a*c+d+b*c*d+a*d-a*b*c)*&(a-d-b*c)*&(a*b*c-a*b*d-a*c+d+b*c*d+a*d-a*b*c)*X1*X2*X2 \\
+.&(1+1)*a*a*b*c*d-a*a*b*d-d*(1+1)*a*a*c*d*d & \\
-(1+1)*a*b*b*c*c+a*b*b*c*d+(1+1)*a*b*c*c*d+a*b*c*d & \\
-b*b*c*c*d-a*a*b*c+a*a*c*d+a*a*d+d+a*b*b*c+a*b*c*c & \\
-(1+1+1)*a*b*c*c*d+a*c*c*d+a*c*d+d+a*b*b*c*c & \\
+c*a*(a*d-b*c-a+a+b+c-d)*(b-d)*X1*X3*X3 & \\
\end{align*}
\]
The following command parses the formula and creates the surface with \((a,b,c,d) = (4,2,2,4)\):

\[
F_{abcd}: \quad \text{\textbackslash ORBITER}PATH\text{\textbackslash orbiter.out-\textbackslash v.3\textbackslash }
\]
\[
\text{\textbackslash define}\ F = \text{\textbackslash finite}\text{\textbackslash field-\textbackslash q.7\textbackslash -end}\text{\textbackslash }
\]
\[
\text{\textbackslash with}\ F = \text{\textbackslash do}\text{\textbackslash }
\]
\[
\text{\textbackslash finite}\text{\textbackslash field}\_\text{activity}\text{\textbackslash }
\]
\[
\text{\textbackslash parse_and_evaluate}\ "F_{abcd}"\ "X0,X1,X2,X3\textbackslash 
\]
\[
\text{\textbackslash report}\ "F_{abcd}\eqn"\ "a=4,b=2,c=2,d=4\textbackslash 
\]
\[
\text{\textbackslash end}
\]

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.

\[
F_{\alpha\beta\gamma\delta\textbackslash q7\_override\_group}:
\]
\[
\text{\textbackslash ORBITER}PATH\text{\textbackslash orbiter.out-\textbackslash v.3\textbackslash }
\]
\[
\text{\textbackslash define}\ F = \text{\textbackslash finite}\text{\textbackslash field-\textbackslash q.7\textbackslash -end}\text{\textbackslash }
\]
\[
\text{\textbackslash define}\ P = \text{\textbackslash projective}\text{\textbackslash space-3\textbackslash F\textbackslash -end}\text{\textbackslash }
\]
\[
\text{\textbackslash with}\ P = \text{\textbackslash do}\text{\textbackslash }
\]
\[
\text{\textbackslash projective}\text{\textbackslash space}\_\text{activity}\text{\textbackslash }
\]
\[
\text{\textbackslash define}\surface F_{2345}\_\textbackslash q7\text{\textbackslash -by_equation}\ "F_{\alpha\beta\gamma\delta}\textbackslash 
\]
\[
\"DF_{\{\alpha,\beta,\gamma,\delta\}}\textbackslash D"\ "x0,x1,x2,x3\textbackslash 
\]
\[
\text{\textbackslash report}\ "F_{\alpha\beta\gamma\delta}\eqn"\ "alpha=2,\beta=3,\gamma=4,\delta=5\textbackslash 
\]
\[
\"Dalpha=2,\beta=3,\gamma=4,\delta=5\D"\text{\textbackslash }
\]
\[
\text{\textbackslash override}\_\text{group}\_6\_2\text{\textbackslash }
\]
\[
1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4\text{\textbackslash }
\]
\[
\text{\textbackslash report}\text{\textbackslash 
\]
\[
\text{\textbackslash end}\text{\textbackslash }
\]
\[
\text{\textbackslash end}\text{\textbackslash }
\]
\[
\text{\textbackslash with}\_\text{F}_{2345}\_\text{do}\text{\textbackslash }
\]
\[
\text{\textbackslash cubic}\text{\textbackslash surface}\_\text{activity}\text{\textbackslash }
\]
\[
\text{\textbackslash report}\text{\textbackslash 
\]
\[
\text{\textbackslash report}\_\text{with}\_\text{group}\text{\textbackslash }
\]
\[
\text{\textbackslash end}
\]
\[
pdflatex\text{\textbackslash surface}\_\text{equation_F}_{\alpha\beta\gamma\delta\textbackslash q7\_report.tex}
\]
\[
open\text{\textbackslash surface}\_\text{equation_F}_{\alpha\beta\gamma\delta\textbackslash q7\_report.pdf}
\]
\[
pdflatex\text{\textbackslash surface}\_\text{equation_F}_{\alpha\beta\gamma\delta\textbackslash q7\_with\_group.tex}
\]
\[
open\text{\textbackslash surface}\_\text{equation_F}_{\alpha\beta\gamma\delta\textbackslash q7\_with\_group.pdf}
\]
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 previous 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::
▷ pdflatex quartic_curve_catalogue_q9_iso0_report.tex
▷ open-quartic.curve_catalogue_q9_iso0_report.pdf
```

The report contains the following information:

The equation

The equation of the quartic curve is:

\[
\alpha^3 X_0^3 X_1 + \alpha^4 X_0^3 X_2 + \alpha^7 X_0 X_1^3 + \alpha^6 X_1^3 X_2 + \alpha^2 X_0 X_2^3 + X_1 X_2^3
\]

\((0,0,0,8,2,4,5,7,1,0,0,0,0,0,0,0)\)

The gradient

The gradient of the quartic curve is:

\[
\alpha^7 X_1^3 + \alpha^2 X_2^3
\]

\((0,4,7,0,0,0,0,0,0,0,0)\)

\[
\alpha^3 X_0^3 + X_2^3
\]
\[ \alpha^4 X^3_0 + \alpha^6 X^3_1 \]
\[ (2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0) \]

**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 : \(P_{30} = (2, 2, 1)\)  
20 : \(P_{58} = (3, 5, 1)\)  
11 : \(P_{32} = (4, 2, 1)\)  
21 : \(P_{62} = (7, 5, 1)\)  
12 : \(P_{34} = (6, 2, 1)\)  
22 : \(P_{76} = (3, 7, 1)\)  
13 : \(P_{38} = (1, 3, 1)\)  
23 : \(P_{77} = (4, 7, 1)\)  
14 : \(P_{41} = (4, 3, 1)\)  
24 : \(P_{78} = (5, 7, 1)\)  
15 : \(P_{44} = (7, 3, 1)\)  
25 : \(P_{82} = (0, 8, 1)\)  
16 : \(P_{46} = (0, 4, 1)\)  
26 : \(P_{83} = (1, 8, 1)\)  
17 : \(P_{51} = (5, 4, 1)\)  
27 : \(P_{84} = (2, 8, 1)\)  
28 : \(P_{84} = (0, 8, 1)\)

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 : \(P_{12} = (1, 0, 1) = a_3 \cap a_4 \cap c_{23} \cap c_{24}\)
<table>
<thead>
<tr>
<th></th>
<th>( P_{13} = (2, 0, 1) = c_{34} \cap c_{36} \cap c_{45} \cap c_{56} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>( P_{15} = (4, 0, 1) = b_3 \cap b_6 \cap c_{13} \cap c_{16} )</td>
</tr>
<tr>
<td>8</td>
<td>( P_{16} = (5, 0, 1) = a_5 \cap a_6 \cap c_{25} \cap c_{26} )</td>
</tr>
<tr>
<td>9</td>
<td>( P_{18} = (7, 0, 1) = a_2 \cap b_7 \cap c_{35} \cap c_{46} )</td>
</tr>
<tr>
<td>10</td>
<td>( P_{19} = (8, 0, 1) = b_4 \cap b_5 \cap c_{14} \cap c_{15} )</td>
</tr>
<tr>
<td>11</td>
<td>( P_{20} = (0, 1, 1) = a_2 \cap b_3 \cap c_{14} \cap c_{56} )</td>
</tr>
<tr>
<td>12</td>
<td>( P_{21} = (2, 1, 1) = b_2 \cap b_4 \cap c_{26} \cap c_{46} )</td>
</tr>
<tr>
<td>13</td>
<td>( P_{22} = (3, 1, 1) = a_4 \cap b_5 \cap c_{12} \cap c_{36} )</td>
</tr>
<tr>
<td>14</td>
<td>( P_{23} = (4, 1, 1) = a_6 \cap b_1 \cap c_{23} \cap c_{45} )</td>
</tr>
<tr>
<td>15</td>
<td>( P_{26} = (7, 1, 1) = c_{16} \cap c_{25} \cap c_{34} \cap d )</td>
</tr>
<tr>
<td>16</td>
<td>( P_{27} = (8, 1, 1) = a_3 \cap a_5 \cap c_{13} \cap c_{15} )</td>
</tr>
<tr>
<td>17</td>
<td>( P_{28} = (0, 2, 1) = c_{12} \cap c_{13} \cap c_{25} \cap c_{36} )</td>
</tr>
<tr>
<td>18</td>
<td>( P_{29} = (1, 2, 1) = b_1 \cap b_5 \cap c_{16} \cap c_{56} )</td>
</tr>
<tr>
<td>19</td>
<td>( P_{31} = (3, 2, 1) = a_3 \cap a_6 \cap c_{34} \cap c_{46} )</td>
</tr>
<tr>
<td>20</td>
<td>( P_{33} = (5, 2, 1) = a_2 \cap b_4 \cap c_{24} \cap d )</td>
</tr>
<tr>
<td>21</td>
<td>( P_{35} = (7, 2, 1) = b_2 \cap b_6 \cap c_{23} \cap c_{36} )</td>
</tr>
<tr>
<td>22</td>
<td>( P_{36} = (8, 2, 1) = a_4 \cap b_3 \cap c_{15} \cap c_{26} )</td>
</tr>
<tr>
<td>23</td>
<td>( P_{37} = (0, 3, 1) = a_5 \cap b_1 \cap c_{24} \cap c_{36} )</td>
</tr>
<tr>
<td>24</td>
<td>( P_{39} = (2, 3, 1) = a_2 \cap a_6 \cap c_{12} \cap c_{16} )</td>
</tr>
<tr>
<td>25</td>
<td>( P_{40} = (3, 3, 1) = b_3 \cap b_4 \cap c_{35} \cap c_{45} )</td>
</tr>
<tr>
<td>26</td>
<td>( P_{42} = (5, 3, 1) = a_4 \cap b_2 \cap c_{13} \cap c_{56} )</td>
</tr>
<tr>
<td>27</td>
<td>( P_{43} = (6, 3, 1) = a_3 \cap b_6 \cap c_{14} \cap c_{25} )</td>
</tr>
<tr>
<td>28</td>
<td>( P_{45} = (8, 3, 1) = c_{15} \cap c_{23} \cap c_{46} \cap d )</td>
</tr>
<tr>
<td>29</td>
<td>( P_{47} = (1, 4, 1) = a_6 \cap b_2 \cap c_{14} \cap c_{35} )</td>
</tr>
<tr>
<td>30</td>
<td>( P_{48} = (2, 4, 1) = b_3 \cap b_5 \cap c_{23} \cap c_{25} )</td>
</tr>
<tr>
<td>31</td>
<td>( P_{49} = (3, 4, 1) = a_5 \cap b_6 \cap c_{56} \cap d )</td>
</tr>
<tr>
<td>32</td>
<td>( P_{50} = (4, 4, 1) = a_2 \cap a_3 \cap c_{26} \cap c_{36} )</td>
</tr>
<tr>
<td>33</td>
<td>( P_{52} = (6, 4, 1) = b_1 \cap b_4 \cap c_{13} \cap c_{34} )</td>
</tr>
<tr>
<td>34</td>
<td>( P_{54} = (8, 4, 1) = c_{12} \cap c_{15} \cap c_{24} \cap c_{45} )</td>
</tr>
<tr>
<td>35</td>
<td>( P_{55} = (0, 5, 1) = a_4 \cap a_6 \cap b_4 \cap b_6 )</td>
</tr>
<tr>
<td>36</td>
<td>( P_{56} = (1, 5, 1) = c_{13} \cap c_{26} \cap c_{45} \cap d )</td>
</tr>
<tr>
<td>37</td>
<td>( P_{59} = (4, 5, 1) = c_{24} \cap c_{25} \cap c_{46} \cap c_{56} )</td>
</tr>
<tr>
<td>38</td>
<td>( P_{60} = (5, 5, 1) = c_{12} \cap c_{14} \cap c_{23} \cap c_{34} )</td>
</tr>
<tr>
<td>39</td>
<td>( P_{61} = (6, 5, 1) = a_2 \cap a_5 \cap b_2 \cap b_5 )</td>
</tr>
<tr>
<td>40</td>
<td>( P_{63} = (8, 5, 1) = c_{15} \cap c_{16} \cap c_{35} \cap c_{36} )</td>
</tr>
<tr>
<td>41</td>
<td>( P_{64} = (0, 6, 1) = a_1 \cap b_5 \cap c_{26} \cap c_{34} )</td>
</tr>
<tr>
<td>42</td>
<td>( P_{65} = (1, 6, 1) = a_1 \cap b_4 \cap c_{25} \cap c_{36} )</td>
</tr>
<tr>
<td>43</td>
<td>( P_{66} = (2, 6, 1) = a_1 \cap b_6 \cap c_{24} \cap c_{35} )</td>
</tr>
<tr>
<td>44</td>
<td>( P_{67} = (3, 6, 1) = a_1 \cap a_2 \cap c_{13} \cap c_{23} )</td>
</tr>
<tr>
<td>45</td>
<td>( P_{68} = (4, 6, 1) = a_1 \cap b_2 \cap c_{12} \cap d )</td>
</tr>
<tr>
<td>46</td>
<td>( P_{69} = (5, 6, 1) = a_1 \cap a_3 \cap b_1 \cap b_3 )</td>
</tr>
<tr>
<td>47</td>
<td>( P_{70} = (6, 6, 1) = a_1 \cap a_4 \cap c_{16} \cap c_{46} )</td>
</tr>
<tr>
<td>48</td>
<td>( P_{71} = (7, 6, 1) = a_1 \cap a_5 \cap c_{14} \cap c_{45} )</td>
</tr>
<tr>
<td>49</td>
<td>( P_{72} = (8, 6, 1) = a_1 \cap a_6 \cap c_{15} \cap c_{56} )</td>
</tr>
</tbody>
</table>
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:

0 : $P_6 = (3, 1, 0)$
1 : $P_7 = (4, 1, 0)$
2 : $P_8 = (5, 1, 0)$
3 : $P_9 = (6, 1, 0)$
4 : $P_{10} = (7, 1, 0)$
5 : $P_{11} = (8, 1, 0)$
6 : $P_{12} = (1, 0, 1)$
7 : $P_{13} = (2, 0, 1)$
8 : $P_{15} = (4, 0, 1)$
9 : $P_{16} = (5, 0, 1)$
10 : $P_{18} = (7, 0, 1)$
11 : $P_{19} = (8, 0, 1)$
12 : $P_{20} = (0, 1, 1)$
13 : $P_{21} = (2, 1, 1)$
14 : $P_{22} = (3, 1, 1)$
15 : $P_{23} = (4, 1, 1)$
16 : $P_{26} = (7, 1, 1)$
17 : $P_{27} = (8, 1, 1)$
18 : $P_{28} = (0, 2, 1)$
19 : $P_{29} = (1, 2, 1)$
20 : $P_{31} = (3, 2, 1)$
21 : $P_{33} = (5, 2, 1)$
22 : $P_{35} = (7, 2, 1)$
44 : $P_{66} = (2, 6, 1)$
45 : $P_{67} = (3, 6, 1)$
46 : $P_{68} = (4, 6, 1)$
47 : $P_{69} = (5, 6, 1)$
48 : $P_{70} = (6, 6, 1)$
49 : $P_{71} = (7, 6, 1)$
50 : $P_{72} = (8, 6, 1)$
51 : $P_{73} = (0, 7, 1)$
52 : $P_{74} = (1, 7, 1)$
53 : $P_{75} = (2, 7, 1)$
54 : $P_{79} = (6, 7, 1)$
55 : $P_{80} = (7, 7, 1)$
56 : $P_{81} = (8, 7, 1)$
57 : $P_{85} = (3, 8, 1)$
58 : $P_{86} = (4, 8, 1)$
59 : $P_{87} = (5, 8, 1)$
60 : $P_{88} = (6, 8, 1)$
61 : $P_{89} = (7, 8, 1)$
62 : $P_{90} = (8, 8, 1)$
The lines and their points of contact are:

\[
\begin{align*}
a_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha^3 \end{bmatrix}_8 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 8 \end{bmatrix}_8, \quad P_0 = P(1, 0, 0) \times 4 \\
a_2 &= \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & 1 \end{bmatrix}_{51} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 1 \end{bmatrix}_{51}, \quad P_{53} = P(1, 8, 1) \times 4 \\
a_3 &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^6 \end{bmatrix}_{15} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 5 \end{bmatrix}_{15}, \quad P_{57} = P(2, 5, 1) \times 4 \\
a_4 &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^2 \end{bmatrix}_{17} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 7 \end{bmatrix}_{17}, \quad P_{53} = P(7, 4, 1) \times 4 \\
a_5 &= \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^7 \end{bmatrix}_{74} = \begin{bmatrix} 1 & 0 & 7 \\ 0 & 1 & 4 \end{bmatrix}_{74}, \quad P_{30} = P(2, 2, 1) \times 4 \\
a_6 &= \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^6 \end{bmatrix}_{77} = \begin{bmatrix} 1 & 0 & 7 \\ 0 & 1 & 7 \end{bmatrix}_{77}, \quad P_5 = P(2, 1, 0) \times 4 \\
b_1 &= \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^7 \end{bmatrix}_{54} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \end{bmatrix}_{54}, \quad P_{58} = P(3, 5, 1) \times 4 \\
b_2 &= \begin{bmatrix} 1 & 0 & \alpha^7 \\ 0 & 1 & \alpha^6 \end{bmatrix}_{45} = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 5 \end{bmatrix}_{45}, \quad P_{14} = P(3, 0, 1) \times 4 \\
b_3 &= \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & 1 \end{bmatrix}_{31} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \end{bmatrix}_{31}, \quad P_{62} = P(7, 5, 1) \times 4 \\
b_4 &= \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & \alpha^2 \end{bmatrix}_{67} = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 7 \end{bmatrix}_{67}, \quad P_{77} = P(4, 7, 1) \times 4 \\
b_5 &= \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & \alpha^3 \end{bmatrix}_{68} = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 8 \end{bmatrix}_{68}, \quad P_{41} = P(4, 3, 1) \times 4 \\
b_6 &= \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^2 \end{bmatrix}_{37} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 7 \end{bmatrix}_{37}, \quad P_3 = P(1, 1, 1) \times 4 \\
c_{12} &= \begin{bmatrix} 1 & 0 & \alpha^3 \\ 0 & 1 & \alpha^4 \end{bmatrix}_{82} = \begin{bmatrix} 1 & 0 & 8 \\ 0 & 1 & 2 \end{bmatrix}_{82}, \quad P_{17} = P(6, 0, 1) \times 4 \\
c_{13} &= \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^4 \end{bmatrix}_{32} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \end{bmatrix}_{32}, \quad P_{32} = P(2, 8, 1) \times 4 \\
c_{14} &= \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & 1 \end{bmatrix}_{61} = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 1 \end{bmatrix}_{61}, \quad P_{32} = P(4, 2, 1) \times 4
\end{align*}
\]
\[
\begin{align*}
c_{15} &= \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & 0 \end{bmatrix}_{60} = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 0 \end{bmatrix}_{60} \quad P_1 = P(0, 1, 0) \times 4 \\
c_{16} &= \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^6 \end{bmatrix}_{35} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 5 \end{bmatrix}_{35} \quad P_{51} = P(5, 4, 1) \times 4 \\
c_{23} &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^5 \end{bmatrix}_{16} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 6 \end{bmatrix}_{16} \quad P_{82} = P(0, 8, 1) \times 4 \\
c_{24} &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^7 \end{bmatrix}_{14} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 4 \end{bmatrix}_{14} \quad P_{25} = P(6, 1, 1) \times 4 \\
c_{25} &= \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^4 \end{bmatrix}_{72} = \begin{bmatrix} 1 & 0 & 7 \\ 0 & 1 & 2 \end{bmatrix}_{72} \quad P_{76} = P(3, 7, 1) \times 4 \\
c_{26} &= \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^3 \end{bmatrix}_{78} = \begin{bmatrix} 1 & 0 & 7 \\ 0 & 1 & 8 \end{bmatrix}_{78} \quad P_{44} = P(7, 3, 1) \times 4 \\
c_{34} &= \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^3 \end{bmatrix}_{28} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 8 \end{bmatrix}_{28} \quad P_{38} = P(1, 3, 1) \times 4 \\
c_{35} &= \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^4 \end{bmatrix}_{52} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 2 \end{bmatrix}_{52} \quad P_{24} = P(5, 4, 1) \times 4 \\
c_{36} &= \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^7 \end{bmatrix}_{24} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 4 \end{bmatrix}_{24} \quad P_{78} = P(5, 7, 1) \times 4 \\
c_{45} &= \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^6 \end{bmatrix}_{25} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 5 \end{bmatrix}_{25} \quad P_{34} = P(6, 2, 1) \times 4 \\
c_{46} &= \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha \end{bmatrix}_{53} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 3 \end{bmatrix}_{53} \quad P_{46} = P(0, 4, 1) \times 4 \\
c_{56} &= \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & 1 \end{bmatrix}_{21} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix}_{21} \quad P_4 = P(1, 1, 0) \times 4 \\
d &= \begin{bmatrix} 1 & \alpha^6 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{59} = \begin{bmatrix} 1 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{59} \quad P_2 = P(0, 0, 1) \times 4 \\
\end{align*}

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}

<table>
<thead>
<tr>
<th>28</th>
<th>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</th>
</tr>
</thead>
</table>

point types: 1^{28}
| 28 | 1 | 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 |
| 63 | 4 | 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, 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 |

Point types for points off the curve: $4^{63}$

Lines on points off the curve:
Off point $0 = P_0 = (3, 1, 0)$ lies on 4 bisecants: \{ 4, 9, 16, 17 \}
Off point $1 = P_1 = (4, 1, 0)$ lies on 4 bisecants: \{ 13, 14, 23, 25 \}
Off point $2 = P_2 = (5, 1, 0)$ lies on 4 bisecants: \{ 1, 3, 19, 24 \}
Off point $3 = P_3 = (6, 1, 0)$ 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 $5 = P_5 = (8, 1, 0)$ lies on 4 bisecants: \{ 7, 8, 18, 21 \}
Off point $6 = P_6 = (1, 0, 1)$ 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 $8 = P_{15} = (4, 0, 1)$ lies on 4 bisecants: \{ 8, 11, 13, 16 \}
Off point $9 = P_{16} = (5, 0, 1)$ lies on 4 bisecants: \{ 4, 5, 19, 20 \}
Off point $10 = P_{18} = (7, 0, 1)$ lies on 4 bisecants: \{ 1, 6, 22, 25 \}
Off point $11 = P_{19} = (8, 0, 1)$ lies on 4 bisecants: \{ 9, 10, 14, 15 \}
Off point $12 = P_{20} = (0, 1, 1)$ lies on 4 bisecants: \{ 1, 8, 14, 26 \}
Off point $13 = P_{21} = (2, 1, 1)$ lies on 4 bisecants: \{ 7, 9, 20, 25 \}
Off point $14 = P_{22} = (3, 1, 1)$ lies on 4 bisecants: \{ 3, 10, 12, 23 \}
Off point $15 = P_{23} = (4, 1, 1)$ lies on 4 bisecants: \{ 5, 6, 17, 24 \}
Off point $16 = P_{26} = (7, 1, 1)$ lies on 4 bisecants: \{ 16, 19, 21, 27 \}
Off point $17 = P_{27} = (8, 1, 1)$ lies on 4 bisecants: \{ 2, 4, 13, 15 \}
Off point $18 = P_{28} = (0, 2, 1)$ lies on 4 bisecants: \{ 12, 13, 19, 22 \}
Off point $19 = P_{29} = (1, 2, 1)$ lies on 4 bisecants: \{ 6, 10, 16, 26 \}
Off point $20 = P_{31} = (3, 2, 1)$ lies on 4 bisecants: \{ 2, 5, 21, 25 \}
Off point $21 = P_{33} = (5, 2, 1)$ lies on 4 bisecants: \{ 1, 9, 18, 27 \}
Off point $22 = P_{35} = (7, 2, 1)$ lies on 4 bisecants: \{ 7, 11, 17, 23 \}
Off point $23 = P_{36} = (8, 2, 1)$ lies on 4 bisecants: \{ 3, 8, 15, 20 \}
Off point $24 = P_{37} = (0, 3, 1)$ lies on 4 bisecants: \{ 4, 6, 18, 23 \}
Off point $25 = P_{39} = (2, 3, 1)$ lies on 4 bisecants: \{ 1, 5, 12, 16 \}
Off point $26 = P_{40} = (3, 3, 1)$ lies on 4 bisecants: \{ 8, 9, 22, 24 \}
<table>
<thead>
<tr>
<th>Point</th>
<th>Equation</th>
<th>Lies on 4 Bisecants</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>$P_{42} = (5, 3, 1)$</td>
<td>${3, 7, 13, 26}$</td>
</tr>
<tr>
<td>28</td>
<td>$P_{43} = (6, 3, 1)$</td>
<td>${2, 11, 14, 19}$</td>
</tr>
<tr>
<td>29</td>
<td>$P_{45} = (8, 3, 1)$</td>
<td>${15, 17, 25, 27}$</td>
</tr>
<tr>
<td>30</td>
<td>$P_{47} = (1, 4, 1)$</td>
<td>${5, 7, 14, 22}$</td>
</tr>
<tr>
<td>31</td>
<td>$P_{48} = (2, 4, 1)$</td>
<td>${8, 10, 17, 19}$</td>
</tr>
<tr>
<td>32</td>
<td>$P_{49} = (3, 4, 1)$</td>
<td>${4, 11, 26, 27}$</td>
</tr>
<tr>
<td>33</td>
<td>$P_{50} = (4, 4, 1)$</td>
<td>${1, 2, 20, 23}$</td>
</tr>
<tr>
<td>34</td>
<td>$P_{52} = (6, 4, 1)$</td>
<td>${6, 9, 13, 21}$</td>
</tr>
<tr>
<td>35</td>
<td>$P_{54} = (8, 4, 1)$</td>
<td>${12, 15, 18, 24}$</td>
</tr>
<tr>
<td>36</td>
<td>$P_{55} = (0, 5, 1)$</td>
<td>${3, 5, 9, 11}$</td>
</tr>
<tr>
<td>37</td>
<td>$P_{56} = (1, 5, 1)$</td>
<td>${13, 20, 24, 27}$</td>
</tr>
<tr>
<td>38</td>
<td>$P_{59} = (4, 5, 1)$</td>
<td>${18, 19, 25, 26}$</td>
</tr>
<tr>
<td>39</td>
<td>$P_{60} = (5, 5, 1)$</td>
<td>${12, 14, 17, 21}$</td>
</tr>
<tr>
<td>40</td>
<td>$P_{61} = (6, 5, 1)$</td>
<td>${1, 4, 7, 10}$</td>
</tr>
<tr>
<td>41</td>
<td>$P_{63} = (8, 5, 1)$</td>
<td>${15, 16, 22, 23}$</td>
</tr>
<tr>
<td>42</td>
<td>$P_{64} = (0, 6, 1)$</td>
<td>${0, 10, 20, 21}$</td>
</tr>
<tr>
<td>43</td>
<td>$P_{65} = (1, 6, 1)$</td>
<td>${0, 9, 19, 23}$</td>
</tr>
<tr>
<td>44</td>
<td>$P_{66} = (2, 6, 1)$</td>
<td>${0, 11, 18, 22}$</td>
</tr>
<tr>
<td>45</td>
<td>$P_{67} = (3, 6, 1)$</td>
<td>${0, 1, 13, 17}$</td>
</tr>
<tr>
<td>46</td>
<td>$P_{68} = (4, 6, 1)$</td>
<td>${0, 7, 12, 27}$</td>
</tr>
<tr>
<td>47</td>
<td>$P_{69} = (5, 6, 1)$</td>
<td>${0, 2, 6, 8}$</td>
</tr>
<tr>
<td>48</td>
<td>$P_{70} = (6, 6, 1)$</td>
<td>${0, 3, 16, 25}$</td>
</tr>
<tr>
<td>49</td>
<td>$P_{71} = (7, 6, 1)$</td>
<td>${0, 4, 14, 24}$</td>
</tr>
<tr>
<td>50</td>
<td>$P_{72} = (8, 6, 1)$</td>
<td>${0, 5, 15, 26}$</td>
</tr>
<tr>
<td>51</td>
<td>$P_{73} = (0, 7, 1)$</td>
<td>${2, 7, 16, 24}$</td>
</tr>
<tr>
<td>52</td>
<td>$P_{74} = (1, 7, 1)$</td>
<td>${4, 8, 12, 25}$</td>
</tr>
<tr>
<td>53</td>
<td>$P_{75} = (2, 7, 1)$</td>
<td>${3, 6, 14, 27}$</td>
</tr>
<tr>
<td>54</td>
<td>$P_{76} = (3, 7, 1)$</td>
<td>${17, 20, 22, 26}$</td>
</tr>
<tr>
<td>55</td>
<td>$P_{78} = (4, 7, 1)$</td>
<td>${5, 10, 13, 18}$</td>
</tr>
<tr>
<td>56</td>
<td>$P_{81} = (5, 7, 1)$</td>
<td>${1, 11, 15, 21}$</td>
</tr>
<tr>
<td>57</td>
<td>$P_{85} = (3, 8, 1)$</td>
<td>${14, 16, 18, 20}$</td>
</tr>
<tr>
<td>58</td>
<td>$P_{86} = (4, 8, 1)$</td>
<td>${3, 4, 21, 22}$</td>
</tr>
<tr>
<td>59</td>
<td>$P_{87} = (5, 8, 1)$</td>
<td>${10, 11, 24, 25}$</td>
</tr>
<tr>
<td>60</td>
<td>$P_{88} = (6, 8, 1)$</td>
<td>${5, 8, 23, 27}$</td>
</tr>
<tr>
<td>61</td>
<td>$P_{89} = (7, 8, 1)$</td>
<td>${2, 9, 12, 26}$</td>
</tr>
<tr>
<td>62</td>
<td>$P_{90} = (8, 8, 1)$</td>
<td>${6, 7, 15, 19}$</td>
</tr>
</tbody>
</table>
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 \( P\Gamma L(4,q) \) of \( PG(3,q) \). The approach described in [12] relies on Schlafli’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 \( PG(3,4) \), 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:

\[
\text{surface classify q4:}
\]
\[
\text{\textasciitilde (ORBITER PATH)orbiter.out -v 5 -
\text{\textasciitilde define F:finite_field:q4: end
\text{\textasciitilde define P:projective_space:3:F: end
\text{\textasciitilde with P:do
\text{\textasciitilde projective_space_activity
\text{\textasciitilde classify_surfaces_with_double_sixes:Surf27:-W:-end
\text{\textasciitilde with Surf27:do
\text{\textasciitilde classification_of_cubic_surfaces_with_double_sixes_activity
\text{\textasciitilde report:-end
\text{\textasciitilde end
\text{\textasciitilde print_symbols
\text{\textasciitilde pdflatex Surfaces_q4.tex
\text{\textasciitilde open Surfaces_q4.pdf
\]

The \texttt{-report} option creates a latex report. After some redactions, the report contains the following elements.

---

The semilinear group

The Action

Group action \( P\Gamma L(4,4) \) of degree 85
The group is a matrix group.

The base action is on projective space \( PG(3,4) \)
\( q = 4 \)
\p = 2
\e = 2
\n = 3
Number of points = 85
Number of lines = 357
Number of lines on a point = 21
Number of points on a line = 5
;

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)
\q = 4
\p = 2
\e = 2
\n = 3
Number of points = 85
Number of lines = 357
Number of lines on a point = 21
Number of points on a line = 5
;

The group stabilizing the fixed line

The Action

Group action \text{PGL}(4, 4)\text{OnWedgeres100} 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>4</td>
<td>10</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>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>2</td>
<td>{ 0, 3, 56, 80, 92 }</td>
<td>(360, 15360)</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>3</td>
<td>{ 0, 3, 56, 80, 93 }</td>
<td>(120, 46080)</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----------------------</td>
<td>--------------</td>
<td>-----</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

215
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:

\[
\begin{align*}
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
, & &
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
\omega^2 & 0 & \omega & 0 \\
0 & \omega^2 & 0 & 1 \\
\end{pmatrix}
, & &
\begin{pmatrix}
1 & 0 & 0 & 0 \\
\omega^2 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & \omega^2 & 1 \\
\end{pmatrix}
, \\
\begin{pmatrix}
1 & 0 & 0 & 0 \\
\omega^2 & \omega & 0 & 0 \\
0 & \omega^2 & \omega & 0 \\
0 & 0 & 1 & 1 \\
\end{pmatrix}
, & &
\begin{pmatrix}
1 & 0 & 0 & 0 \\
\omega^2 & \omega & 0 & 0 \\
0 & \omega & 0 & 0 \\
\omega^2 & 1 & \omega^2 & 1 \\
\end{pmatrix}
, & &
\begin{pmatrix}
1 & 1 & 0 & 0 \\
\omega^2 & 0 & 0 & 0 \\
0 & 0 & \omega^2 & \omega^2 \\
0 & 0 & 1 & 0 \\
\end{pmatrix}
\end{align*}
\]
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:
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:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_1,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}_1,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
\omega^2 & 0 & \omega & 0 \\
\omega & \omega & \omega & 1
\end{bmatrix}_1,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega^2 & 0 & \omega & 0 \\
\omega & \omega & \omega & 1
\end{bmatrix}_1,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & 0 & \omega & 0 \\
\omega & \omega & \omega & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}_1
\]

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:

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

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

The overall number of objects is: 38080

**The Group** PΓL(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:
Strong generators for a group of order 51840:

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

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

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

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,
1,0,0,0,0,1,0,0,1,1,1,0,1,1,0,1,0,
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,

The overall number of objects is: 38080

**Surface 4#0**

**The equation**

The equation of the surface is:

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

(0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 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:

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

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

1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 3, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 3, 2, 2, 0, 0, 0, 0, 2, 0, 1, 0, 3, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 2, 2, 0, 0, 2, 0, 2, 2, 1, 1, 0, 1, 3, 1, 2, 1, 0, 2, 0, 3, 2, 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, 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 & 0 & 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^2 & 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^2 & 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}
\[
\ell_{16} = c_{16} = \begin{bmatrix} 1 & 0 & \omega^2 & 0 \\ 0 & 1 & 0 & \omega \end{bmatrix}_{71} = \begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}_{71} = \text{Pl}(3, 2, 0, 0, 3, 1)_{299}
\]

\[
\ell_{17} = c_{23} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{85} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{85} = \text{Pl}(1, 1, 1, 1, 0, 0)_{16}
\]

\[
\ell_{18} = c_{24} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{122} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{122} = \text{Pl}(0, 1, 1, 1, 1, 1)_{202}
\]

\[
\ell_{19} = c_{25} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{110} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{110} = \text{Pl}(1, 0, 1, 1, 1, 1)_{199}
\]

\[
\ell_{20} = c_{26} = \begin{bmatrix} 1 & 0 & \omega & 0 \\ 0 & 1 & 1 & \omega^2 \end{bmatrix}_{55} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 3 \end{bmatrix}_{55} = \text{Pl}(2, 3, 2, 0, 2, 1)_{244}
\]

\[
\ell_{21} = c_{34} = \begin{bmatrix} 1 & \omega & 0 & 1 \\ 0 & 0 & 1 & \omega^2 \end{bmatrix}_{145} = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 3 \end{bmatrix}_{145} = \text{Pl}(0, 3, 2, 3, 2, 1)_{271}
\]

\[
\ell_{22} = c_{35} = \begin{bmatrix} 1 & 0 & \omega & 1 \\ 0 & 1 & 1 & \omega^2 \end{bmatrix}_{139} = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 3 \end{bmatrix}_{139} = \text{Pl}(2, 0, 2, 3, 2, 1)_{267}
\]

\[
\ell_{23} = c_{36} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{26} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{26} = \text{Pl}(1, 1, 1, 0, 1, 1)_{180}
\]

\[
\ell_{24} = c_{45} = \begin{bmatrix} 1 & \omega^2 & 0 & 0 \\ 0 & 0 & 1 & \omega \end{bmatrix}_{81} = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}_{81} = \text{Pl}(0, 0, 3, 2, 3, 1)_{332}
\]

\[
\ell_{25} = c_{46} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{100} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{100} = \text{Pl}(0, 1, 1, 0, 0, 0)_{6}
\]

\[
\ell_{26} = c_{56} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{1} = \text{Pl}(1, 0, 1, 0, 0, 0)_{3}
\]

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:

\[ E_{56} = a_5 \cap b_6 \cap c_{56} = P_0 = P_0 = P(1,0,0,0) = P(1,0,0,0), \ T = 0 \]
\[ E_{51} = a_5 \cap b_1 \cap c_{15} = P_1 = P_1 = P(0,1,0,0) = P(0,1,0,0), \ T = 4 \]
\[ E_{46} = a_4 \cap b_6 \cap c_{46} = P_2 = P_2 = P(0,0,1,0) = P(0,0,1,0), \ T = 20 \]
\[ E_{41} = a_4 \cap b_1 \cap c_{14} = P_3 = P_3 = P(0,0,0,1) = P(0,0,0,1), \ T = 84 \]
\[ E_{32} = a_3 \cap b_2 \cap c_{23} = P_4 = P_4 = P(1,1,1,1) = P(1,1,1,1), \ T = 27 \]
\[ E_{52} = a_5 \cap b_2 \cap c_{25} = P_5 = P_5 = P(1,1,0,0) = P(1,1,0,0), \ T = 1 \]
\[ E_{54} = a_5 \cap b_4 \cap c_{45} = P_6 = P_6 = P(\omega,1,0,0) = P(2,1,0,0), \ T = 2 \]
\[ E_{53} = a_5 \cap b_3 \cap c_{35} = P_7 = P_7 = P(\omega^2,1,0,0) = P(3,1,0,0), \ T = 3 \]
\[ E_{36} = a_3 \cap b_6 \cap c_{36} = P_8 = P_8 = P(0,1,0,1) = P(0,1,0,1), \ T = 5 \]
\[ E_{16} = a_1 \cap b_6 \cap c_{16} = P_9 = P_9 = P(\omega,0,1,0) = P(2,0,1,0), \ T = 10 \]
\[ E_{26} = a_2 \cap b_6 \cap c_{26} = P_{10} = P_{10} = P(\omega^2,0,1,0) = P(3,0,1,0), \ T = 15 \]
\[ E_{14,23,56} = c_{14} \cap c_{23} \cap c_{56} = P_{11} = P_{11} = P(0,1,1,0) = P(0,1,1,0), \ T = 9 \]
\[ E_{13,24,56} = c_{13} \cap c_{24} \cap c_{56} = P_{12} = P_{12} = P(1,1,1,0) = P(1,1,1,0), \ T = 6 \]
\[ E_{65} = a_6 \cap b_5 \cap c_{56} = P_{13} = P_{13} = P(\omega,1,1,0) = P(2,1,1,0), \ T = 12 \]
\[ E_{12,34,56} = c_{12} \cap c_{34} \cap c_{56} = P_{14} = P_{14} = P(\omega^2,1,1,0) = P(3,1,1,0), \ T = 18 \]
\[ E_{15,23,46} = c_{15} \cap c_{23} \cap c_{46} = P_{15} = P_{23} = P(1,0,0,1) = P(1,0,0,1), \ T = 21 \]
\[ E_{43} = a_3 \cap b_1 \cap c_{13} = P_{16} = P_{26} = P(0,1,0,1) = P(0,1,0,1), \ T = 25 \]
\[ E_{15,24,36} = c_{15} \cap c_{24} \cap c_{36} = P_{17} = P_{27} = P(1,1,0,1) = P(1,1,0,1), \ T = 22 \]
\[ E_{21} = a_2 \cap b_1 \cap c_{12} = P_{18} = P_{30} = P(0,0,1,1) = P(0,0,1,1), \ T = 46 \]
\[ E_{15,26,34} = c_{15} \cap c_{26} \cap c_{34} = P_{19} = P_{21} = P(1,0,0,1) = P(1,0,0,1), \ T = 24 \]
\[ E_{61} = a_6 \cap b_1 \cap c_{16} = P_{20} = P_{34} = P(0,\omega^2,0,1) = P(0,3,0,1), \ T = 67 \]
\[ E_{15} = a_1 \cap b_5 \cap c_{15} = P_{21} = P_{35} = P(1,\omega^2,0,1) = P(1,3,0,1), \ T = 23 \]
\[ E_{22} = a_2 \cap b_2 \cap c_{24} = P_{22} = P_{38} = P(0,0,1,1) = P(0,0,1,1), \ T = 41 \]
\[ E_{13,25,46} = c_{13} \cap c_{25} \cap c_{46} = P_{23} = P_{39} = P(1,0,1,1) = P(1,0,1,1), \ T = 26 \]
\[ E_{14,25,36} = c_{14} \cap c_{25} \cap c_{36} = P_{24} = P_{42} = P(0,1,1,1) = P(0,1,1,1), \ T = 30 \]
\[ E_{25} = a_6 \cap b_2 \cap c_{26} = P_{25} = P_{47} = P(\omega,1,1,1) = P(2,2,1,1), \ T = 53 \]
\[ E_{26} = a_2 \cap b_5 \cap c_{25} = P_{26} = P_{48} = P(\omega^2,\omega,1,1) = P(3,2,1,1), \ T = 80 \]
\[ E_{16,25,34} = c_{16} \cap c_{25} \cap c_{34} = P_{27} = P_{51} = P(\omega,\omega^2,1,1) = P(2,3,1,1), \ T = 55 \]
\[ E_{12} = a_1 \cap b_2 \cap c_{12} = P_{28} = P_{32} = P(\omega^2,\omega^2,1,1) = P(3,3,1,1), \ T = 79 \]
\[ E_{43} = a_4 \cap b_3 \cap c_{34} = P_{29} = P_{53} = P(0,0,\omega,1) = P(0,0,2,1), \ T = 62 \]
\[ E_{12,35,46} = c_{12} \cap c_{35} \cap c_{46} = P_{30} = P_{54} = P(1,0,\omega,1) = P(1,0,2,1), \ T = 36 \]
\[ E_{35} = a_3 \cap b_5 \cap c_{35} = P_{31} = P_{59} = P(\omega,\omega,1,1) = P(2,1,2,1), \ T = 49 \]
\[ E_{63} = a_6 \cap b_3 \cap c_{36} = P_{32} = P_{60} = P(\omega^2,1,\omega,1) = P(3,1,2,1), \ T = 76 \]
\[ E_{14,26,35} = c_{14} \cap c_{26} \cap c_{35} = P_{33} = P_{61} = P(0,\omega,\omega,1) = P(0,2,2,1), \ T = 51 \]
\[ E_{23} = a_2 \cap b_3 \cap c_{23} = P_{34} = P_{62} = P(1,\omega,\omega,1) = P(1,2,2,1), \ T = 39 \]
\[ E_{13} = a_1 \cap b_3 \cap c_{13} = P_{35} = P_{67} = P(\omega,\omega^2,\omega,1) = P(2,3,2,1), \ T = 50 \]
\[ E_{16,24,35} = c_{16} \cap c_{24} \cap c_{35} = P_{36} = P_{68} = P(\omega^2,\omega^2,\omega,1) = P(3,3,2,1), \ T = 74 \]
\[ E_{45} = a_4 \cap b_5 \cap c_{45} = P_{37} = P_{69} = P(0,0,\omega^2,1) = P(0,0,3,1), \ T = 83 \]
\[ E_{64} = a_6 \cap b_4 \cap c_{46} = P_{38} = P_{70} = P(1,0,\omega^2,1) = P(1,0,3,1), \ T = 31 \]
\[ E_{12,36,45} = c_{12} \cap c_{36} \cap c_{45} = P_{39} = P_{75} = P(\omega,\omega,\omega^2,1) = P(2,1,3,1), \ T = 59 \]
\[ E_{34} = a_3 \cap b_4 \cap c_{34} = P_{40} = P_{76} = P(\omega^2,1,\omega^2,1) = P(3,1,3,1), \ T = 71 \]
41: \( E_{24} = a_2 \cap b_4 \cap c_{24} = P_{41} = P_{79} = P(\omega, \omega, \omega^2, 1) = P(2, 2, 3, 1), \ T = 58 \)
42: \( E_{13,26,45} = c_{13} \cap c_{26} \cap c_{45} = P_{42} = P_{80} = P(\omega^2, \omega, \omega^2, 1) = P(3, 2, 3, 1), \ T = 70 \)
43: \( E_{14} = a_1 \cap b_4 \cap c_{14} = P_{43} = P_{81} = P(0, \omega^2, \omega^2, 1) = P(0, 3, 3, 1), \ T = 72 \)
44: \( E_{16,23,45} = c_{16} \cap c_{23} \cap c_{45} = P_{44} = P_{82} = P(1, \omega^2, \omega^2, 1) = P(1, 3, 3, 1), \ T = 33 \)
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^{10}
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 : \( E_{56}, E_{31}, E_{15,24,36}, E_{16,25,34}, E_{12}, E_{14,26,35}, E_{23}, E_{45}, E_{64} \)

\[
\vdots
\]

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 = \( E_{23}, E_{31}, E_{12} \)

\[
\vdots
\]

239 : 239 = 119,1 = \( E_{12,36,45}, E_{14,26,35}, E_{13,25,46} \)
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  \straightup\arrow -define.F.-finite_field.-q.4.-end.-\n  \straightup\arrow -define.P.-projective_space.3.F.-end.-\n  \straightup\arrow -with.P.-do.-\n  \straightup\arrow -projective_space_activity.-\n  \straightup\arrow \\arrow -control.six_arcs.-problem_label.sixarcs_q4.-end.-\n  \straightup\arrow \\arrow -classify_surfaces_through_arcs_and_two_lines.-\n  \straightup\arrow \\arrow -end\n  pdflatex.surfaces.arc_lifting_4.tex\nopen.surfaces.arc_lifting_4.pdf
```

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></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,

```bash
surface_recognize_q7_abcd_2_3_3_4:
  $ (ORBITER_PATH) orbiter.out -v.3-
  define F::finite_field::q7::end-
  define P::projective_space::3 F::end-
  with P::do-
  projective_space_activity::
  do classify_surfaces_with_double_sixes Surf::W::end-
  do end-
  do with Surf::do-
  do do classification_of_cubic_surfaces_with_double_sixes_activity::
  do do recognize::
```
identifies the surface (cf. Table 4.3)
\[ X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 + X_1X_2X_3 = 0 \] (7.1)
in the classification of surfaces over the field \( \mathbb{F}_7 \). 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

\[
\begin{bmatrix}
1 & 4 & 4 & 0 \\
6 & 0 & 0 & 0 \\
6 & 2 & 0 & 1 \\
7 & 0 & 4 & 0
\end{bmatrix}
\] (7.2)

Orbiter can compute isomorphism between two given surfaces. Both surfaces must have 27 lines. For instance, the command

```
surface_isomorph_16:

> $(ORBITER_PATH)orbiter.out -v.3 \n> -define F -finite_field -q 16 -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 \n> -classification_of_cubic_surfaces_with_double_sixes_activity \n> -isomorphism_testing \n> -q 16 -by_coefficients \n> "1,5,1,8,1,9,1,10,1,11,1,12,6,14,6,15,7,18,7,19" -end \n> -q 16 -by_coefficients \n> "13,6,3,8,3,11,13,13,1,19" -end \n> -end \n> -end \n> -print_symbols
```

228
computes an isomorphism between two cubic surfaces with 27 lines
\[
X_0^2X_2 + X_1^2X_2 + X_1^2X_3 + X_0X_2^2 + X_1X_2^2 + X_2^2X_3 + \delta^{13}X_1X_3^2 + \delta^{12}X_2X_3^2 = 0
\]
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
\[
X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 + X_1X_2X_3 = 0.
\]

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:

```
poly_orbits_d3n3q2.csv:
  · $(\text{ORBITER \ PATH})\text{orbiter.out}$
  · -v 4
  · -define G -linear_group -PGL 4 2 -end
  · -with G -do
  · -group_theoretic_activity
  · · -orbits_on_polynomials 3
  · · -orbits_on_polynomials_draw_tree 6
  · -end
```

To investigate the properties of these surfaces, the following commands can be used:

```
poly_orbits_d3n3q2F2.csv:
poly_orbits_d3n3q2.csv
  · $(\text{ORBITER \ PATH})\text{orbiter.out}$
  · -v 4
  · -define F -finite_field -q 2 -end
  · -define P -projective_space 3 F -end
  · -with P -do
  · -projective_space_activity
  · -table_of_cubic_surfaces_compute_properties
  · · poly_orbits_d3n3q2.csv 2.0
  · -end
```

```
Dickson_q2_analyze:
poly_orbits_d3n3q2F2.csv
  · $(\text{ORBITER \ PATH})\text{orbiter.out}$
  · -v 4
  · -define F -finite_field -q 2 -end
  · -define P -projective_space 3 F -end
  · -with P -do
  · -projective_space_activity
  · -cubic_surface_properties_analyze
  · · poly_orbits_d3n3q2F2.csv 2
  · -end
  · pdflatex poly_orbits_d3n3q2F2_report.tex
  · open poly_orbits_d3n3q2F2_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)orbniter.out-v.4\$

$\$-\$-\$-\$-\$-\$-define F: finite_field: \$q.4\$ - end\$

$\$-\$-\$-\$-\$-\$-define P: projective_space: 3 F - end\$

$\$-\$-\$-\$-\$-\$-with P - do\$

$\$-\$-\$-\$-\$-\$-projective_space_activity\$

$\$-\$-\$-\$-\$-\$-table_of_cubic_surfaces_compute_properties\$

$\$-\$-\$-\$-\$-\$-poly_orbits_d3_n3_q2.csv: 2.0\$

$\$-\$-\$-\$-\$-\$-end\$

Dickson_q4_analyze: poly_orbits_d3_n3_q2_F4.csv

$\$(ORBITER_PATH)orbniter.out-v.4\$

$\$-\$-\$-\$-\$-\$-define F: finite_field: \$q.4\$ - end\$

$\$-\$-\$-\$-\$-\$-define P: projective_space: 3 F - end\$

$\$-\$-\$-\$-\$-\$-with P - do\$

$\$-\$-\$-\$-\$-\$-projective_space_activity\$

$\$-\$-\$-\$-\$-\$-cubic_surface_properties_analyze\$

$\$-\$-\$-\$-\$-\$-poly_orbits_d3_n3_q2_F4.csv: 2.0\$

$\$-\$-\$-\$-\$-\$-end\$

$\$pdflatex-poly_orbits_d3_n3_q2_F4_report.tex$

$\$open-poly_orbits_d3_n3_q2_F4_report.pdf$
7.6 ATLAS and Tables

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

cubic_surfaces_tables_17::
▷ $(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::
▷ $(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

Applications

8.1 Number Theory

In Table 8.1, some number theoretic commands are shown. For instance,

\[
\text{inverse}\mod a: \\
\text{▷ } (\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_2\_\text{inverse}\mod 18059241\_58014043
\]

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

\[
\text{jacobi}\_a: \\
\text{▷ } (\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v\_5\_\text{jacobi}\_2221\_7817
\]
<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 $fname$</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 $x_{n+1} = ax_n + b \pmod{p}$</td>
</tr>
</tbody>
</table>

Table 8.1: 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 `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:

$$\begin{align*}
\left( \frac{2221}{7817} \right) &= \left( \frac{2221}{2221} \right) \cdot (-1)^{\frac{2221-1}{2} \cdot \frac{7817-1}{2}} \\
&= \left( \frac{7817}{2221} \right) \\
&= \left( \frac{7817}{2221} \right) \\
&= \left( \frac{1154}{2221} \right) \\
&= \left( \frac{2}{2221} \right) \cdot \left( \frac{577}{2221} \right) \\
&= (-1)^{\frac{2221^2 - 1}{2} \cdot \frac{577}{2221} \cdot \frac{2221}{577}} \\
&= (-1) \cdot \left( \frac{577}{2221} \right) \\
&= (-1) \cdot \left( \frac{2221}{577} \right) \cdot (-1)^{\frac{577-1}{2} \cdot \frac{2221-1}{2}} \\
&= (-1) \cdot \left( \frac{2221}{577} \right) \\
&= (-1) \cdot \left( \frac{490}{577} \right)
\end{align*}$$
\[
\begin{align*}
&= (-1) \cdot \left( \frac{2}{577} \right) \cdot \left( \frac{245}{577} \right) \\
&= (-1) \cdot (-1)^{\frac{577^2 - 1}{8}} \cdot \left( \frac{245}{577} \right) \\
&= (-1) \cdot \left( \frac{245}{577} \right) \\
&= (-1) \cdot \left( \frac{577}{245} \right) \cdot (-1)^{\frac{245 - 1}{2} \cdot \frac{577 - 1}{2}} \\
&= (-1) \cdot \left( \frac{577}{245} \right) \\
&= (-1) \cdot \left( \frac{87}{245} \right) \\
&= (-1) \cdot \left( \frac{245}{87} \right) \cdot (-1)^{\frac{87 - 1}{2} \cdot \frac{245 - 1}{2}} \\
&= (-1) \cdot \left( \frac{245}{87} \right) \\
&= (-1) \cdot \left( \frac{71}{87} \right) \\
&= (-1) \cdot \left( \frac{87}{71} \right) \cdot (-1)^{\frac{71 - 1}{2} \cdot \frac{87 - 1}{2}} \\
&= \left( \frac{87}{71} \right) \\
&= \left( \frac{16}{71} \right) \\
&= \left( \frac{2}{71} \right)^4 \cdot \left( \frac{1}{71} \right) \\
&= \left( (-1)^{\frac{71^2 - 1}{8}} \right)^4 \cdot \left( \frac{1}{71} \right) \\
&= \left( \frac{1}{71} \right) \\
&= 1
\end{align*}
\]

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 `square_root_mod` command from Section 3.1 to compute a square root of 2221 modulo 7817 and verify this fact. The command

```
sqrt_mod_7817:
```

```
> $(ORBITER_PATH)orbiter.out.-v.2.-square_root_mod.2221.7817
```

yields that 7634 is a square root. Indeed,

\[
7634^2 \equiv 2221 \mod 7817.
\]
### 8.2 Representation Theory

Orbiter has some commands for representations of finite groups. Table 8.2 lists the commands available to classify arcs. The command

```plaintext
representation_on_polynomials_of_degree_3:
  $(ORBITER_PATH)orbiter.out:-v.4:\n  -define:G:-linear_group:-PGL.4.3:-end:\n  -with:G:-do:\n  -group_theoretic_activity:\n  -representation_on_polynomials.3:\n  -end:\n  $(ORBITER_PATH)orbiter.out:-v.2:\n  -loop:L.0-9:1:-draw_matrix:\n  -input_csv_file:PGL.4.3_rep.3.%L.csv:\n  -box_width:40:-bit_depth:24:-partition:3:20:20:-end:\n  -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 8.1 shows the representing matrices for a generating set of size 9.
Figure 8.1: 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 8.3: Cryptographic Commands

### 8.3 Cryptography

In Table 8.3, some cryptographic commands are shown. In Table 8.3, 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:
- \$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2\$
- \$-\text{define}\cdot\text{F}\cdot-\text{finite}\_\text{field}\cdot-q.11\cdot-\text{end}\$
- \$-\text{with}\cdot\text{F}\cdot-\text{do}\$
- \$-\text{finite}\_\text{field}\_\text{activity}\$
- \$-\text{EC}\_\text{add}\cdot1.3\cdot"1,4"\cdot"1,4"\cdot-\text{end}\$

adds the point \((1,4)\) on the curve \(y^2 = x^3 + x + 3 \mod 11\) to itself. The command

EC_cyclic_subgroup:
- \$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2\$
- \$-\text{define}\cdot\text{F}\cdot-\text{finite}\_\text{field}\cdot-q.11\cdot-\text{end}\$
- \$-\text{with}\cdot\text{F}\cdot-\text{do}\$
- \$-\text{finite}\_\text{field}\_\text{activity}\$
- \$-\text{EC}\_\text{cyclic}\_\text{subgroup}\cdot1.3\cdot"1,4"\cdot-\text{end}\$

238
<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 8.4: Finite Field Activities related to Cryptography
Figure 8.2: 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:
  $(ORBITER_PATH)orbiter.out-v.2\$
  -define:F:-finite_field=-q.199:-end\$
  -with:F:-do\$
  -finite_field_activity\$
  -EC_points:"EC_5_7_q199".5:7:-end
  $(ORBITER_PATH)orbiter.out-v.2\$
  -draw_matrix--input_csv_file:EC_5_7_q199_points_xy.csv\$
  -box_width:10:-bit_depth:24\$
  -partition:2:199:199:-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 8.2. Both the $x$-axis and the $y$-axis are indexed by the field elements from 0 to 198.

The command
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_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
```

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
```
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 \(\mathbb{F}_q\): 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 \(-p/2 \leq a_i \leq p/2\). So, in an extension to the convention for field elements in \(\mathbb{F}_q\), 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

\[
\begin{align*}
\text{NTRU}_N &= 7 \\
\text{NTRU}_P &= 3 \\
\text{NTRU}_Q &= 41 \\
\text{NTRU}_D &= 2 \\
\text{NTRUE_XN1} &= "-1,0,0,0,0,0,0,1," \\
\text{ALICE PRIVATE F} &= "-1,0,1,1,-1,0,1" \\
\text{ALICE PRIVATE G} &= "0,-1,-1,0,1,0,1"
\end{align*}
\]

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 \(\mathbb{F}_p[x]/(x^n - 1)\),

2. \(F_q(x)\) the inverse of \(f(x)\) in \(\mathbb{F}_q[x]/(x^n - 1)\).

To this end, we can use the `extended_gcd_for_polynomials` command from Table 8.1. The following two makefile commands do the job:

\[
\begin{align*}
\text{ALICE PRIVATE F} &= "-1,0,1,1,-1,0,1" \\
\text{ALICE PRIVATE G} &= "0,-1,-1,0,1,0,1"
\end{align*}
\]
NTRU_Alice1:
  ➤ $(ORBITER_PATH)orbiter.out-v.2-
  ➤   -define:F:-finite_field-q$(NTRU_Q)-end-
  ➤   -with:F:-do-
  ➤   -finite_field_activity-
  ➤   -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"

NTRU_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}+X^{2}+X+1
ALICE_PRIVATE_FP="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:
  ➤ $(ORBITER_PATH)orbiter.out-v.2-
  ➤   -define:F:-finite_field-q$(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"

The public key is assigned to the makefile variable 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).

BOB_MESSAGE="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 $(NTRU_P) -end·
▷ -with F::do·
```
-finite_field_activity:\n-polynomial_reduce_mod_p\cdot$(ALICE\_C2)\cdot$-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:
\> $(\text{ORBITER\_PATH})\text{orbiter.out} -v 2\:\backslash
\> \> -define F \cdot -finite_field \cdot -q \cdot$(NTRU\_Q)\cdot$-end\:\backslash
\> \> -with F \cdot -do\:\backslash
\> \> -finite_field_activity\:\backslash
\> \> -polynomial\_mult\_mod\cdot$(ALICE\_PRIVATE\_FP)\:\backslash
\> \> \> $(ALICE\_C3)\cdot$$(NTRUE\_XN1)\cdot$\backslash
\> \> -end

#C(X)=2X^{5}+X^{3}+X^{2}+2X+1  
ALICE\_C4=\"1,2,1,0,2\"

NTRU\_decrypt5:
\> $(\text{ORBITER\_PATH})\text{orbiter.out} -v 2\:\backslash
\> \> -define F \cdot -finite_field \cdot -q \cdot$(NTRU\_P)\cdot$-end\:\backslash
\> \> -with F \cdot -do\:\backslash
\> \> -finite_field_activity\:\backslash
\> \> -polynomial\_center\_lift\cdot$(ALICE\_C4)\cdot$-end

#A(X)=-X^{5}+X^{3}+X^{2}-X+1  
#plaintext\_BOB\_MESSAGE

Decryption produces Bob’s message to Alice.
Chapter 9
Coding Theory

9.1 Introduction

In Table 9.1, global coding theoretic commands of Orbiter are shown. The commands

```
Hamming_space_4_2.distance_matrix:
```

```
$\text{(ORBITER_PATH)}\text{orbiter.out} -\text{Hamming\_space\_distance\_matrix\_4\_2}
```

```
$\text{(ORBITER_PATH)}\text{orbiter.out} -\text{v\_2\_draw\_matrix}\�
```

```
\text{\triangleright} \text{\triangleright} -\text{input\_csv\_file}\text{Hamming\_n4\_q2.csv}\�
```

```
\text{\triangleright} \text{\triangleright} -\text{box\_width}\text{\_20\_bit\_depth}\text{\_24\_partition}\text{\_4\_16\_16\_end}
```

```
\text{\open} -\text{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 9.1. Table 9.2 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\"
```

```
\text{\code}_5\text{\_2\_3\_diagram:}
```

```
\text{\triangleright} \text{\triangleright} \text{(ORBITER_PATH)}\text{orbiter.out} -\text{v\_2\_code\_diagram}\text{\_code}_5\text{\_2\_3}\�
```

```
\text{\triangleright} \text{\triangleright} \text{(CODE\_5\_2\_3\_CODEWORDS)}\text{\_5\_metric\_balls}\text{\_1}
```

```
\text{\triangleright} \text{(ORBITER_PATH)}\text{orbiter.out} -\text{v\_2\_draw\_matrix}\�
```

```
\text{\triangleright} \text{\triangleright} -\text{input\_csv\_file}\text{\code}_5\text{\_2\_3\_diagram\_01\_5\_4.csv}\�
```

```
\text{\triangleright} \text{\triangleright} -\text{box\_width}\text{\_25\_bit\_depth}\text{\_24\_partition}\text{\_4\_8\_4\_end}
```

The Hamming graph $H(5,2)$ can be created with the following command:

```
Hamming_5\_2\_graph:
```

```
\text{\triangleright} \text{(ORBITER_PATH)}\text{orbiter.out} -\text{v\_2}\�
```

247
<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 r_1 ... r_k</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 9.1: Global Coding Theoretic Commands
Figure 9.1: 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 9.2: Coding Theoretic Activities
Figure 9.2: 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 9.2.
9.2 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 PG(2, 2). To compute the dual, we need to compute the nullspace of this matrix. The following command does that:

```plaintext
SIMPLEX_CODE_GENERATOR="\n 1,0,1,0,1,0,1,\n 0,1,1,0,0,1,1,\n 0,0,0,1,1,1"

Hamming generator:
   $(ORBITER_PATH)orbiter.out:-v.2-
  -define:F:-finite_field:-q:2:-end-
  -define:v:-vector-field:F:-format:3-
  -dense:$(SIMPLEX_CODE_GENERATOR)-
  -end-
  -with:F:-do-
  -finite_field_activity-
  -nullspace:v-
  -end
   pdflatex.nullspace_3_7.tex
  open-nullspace_3_7.pdf
```

This produces the following output:

```
Input matrix:
\[
\begin{bmatrix}
 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

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

Basis for Perp:
\[
\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]
```
Figure 9.3: The Hamming code

Suppose we want to look at the codewords of the Hamming code in the Hamming space. The following command will produce Figure 9.3.

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="\\
1,0,0,0,1,1,\\
0,1,0,0,1,0,1,\\
0,0,1,0,1,1,0,\\
0,0,0,1,1,1"\\

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 

$$(1, 0, 0, 7, 7, 0, 1).$$

Suppose we want to establish the MacWilliams relations for the Hamming code. The following command creates the matrix of Kravtchuck numbers:

```
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:

$$\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
7 & 5 & 3 & 1 & -1 & -3 & -5 & -7 \\
21 & 9 & 1 & -3 & -3 & 1 & 9 & 21 \\
35 & 5 & -5 & -3 & 3 & 5 & -5 & -35 \\
35 & -5 & -5 & 3 & 3 & -5 & -5 & 35 \\
21 & -9 & 1 & 3 & -3 & -1 & 9 & -21 \\
7 & -5 & 3 & -1 & -1 & 3 & -5 & 7 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1
\end{array}$$

The Hamming code is cyclic. To see this, we need to consider the action of the Singer cycle on the set of points of $\text{PG}(2, 2)$. The following command creates the Singer cycle:

```
Hamming.singer:
▷ $(ORBITER PATH)orbiter.out -v 3 \n▷ -define G -linear_group -PGL 3 2 -singer 1 -end \n▷ -with G -do \n▷ -group_theoretic_activity \n▷ -report \n▷ -orbits_on_points \n▷ -end
▷ pdflatex PGL_3_2_Singer_3_2_1_report.tex
▷ open 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

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

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:

\begin{verbatim}
SIMPLEX_CODE_GENMA_CYCLIC="\\
1,0,0,1,1,1,0,\\
0,1,0,0,1,1,1,\\
0,0,1,1,1,0,1"

Hamming_cyclic_generator:
$\langle$(ORBITER_PATH)orbiter.out-v.2\rangle
\> -define:F:-finite_field:-q:2:-end:\\
\> -define:v:-vector:-format:3:-field:F:\\
\> -dense:$\langle$(SIMPLEX_CODE_GENMA_CYCLIC)\rangle:\\
\> -end:\\
\> -with:F:-do:-finite_field_activity:\\
\> -nullspace:v:\\
\> -end\\
\> pdflatex:NULLSPACE_3_7.tex
\end{verbatim}

254
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:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$
9.3 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, 8, 9, 10, 11, 132, 913, 1460, 1750, 1898, 2518, 2787, 2874, 3320, 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
```
9.4 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-polygonal_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-polygonal_division_file_13.pdf
```

This creates the following output:

```
<table>
<thead>
<tr>
<th>1010110100110101011110000101111100</th>
<th>1101 = 1101111100011111110110000101111100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101</td>
<td>10101100110101010111000101111100</td>
</tr>
</tbody>
</table>
```
1101
====
111110100110101010111000010111100
1101
====
1010100110101010111000010111100
1101
====
111100110101010111000010111100
1101
====
10001101010111000010111100
1101
====
101110101010111000010111100
1101
====
110101010111000010111100
1101
====
101010111000010111100
1101
====
11110111000010111100
1101
====
100111000010111100
1101
====
10011000010111100
1101
====
1001000010111100
1101
====
100000010111100
1101
====
10100010111100
1101
====
1110010111100
1101
====
11010111100
1101
====
101010111000010111100
1101
====
11110111000010111100
1101
====
100111000010111100
1101
====
1001000010111100
1101
====
100000010111100
1101
====
10100010111100
1101
====
1110010111100
1101
====
11010111100
1101
====
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)orbiner.out -encode_text_5bits:
  ▶ ▶ "Hithere"."text.csv"
  ▶ ▶ $(ORBITER_PATH)orbiner.out -v.2:\n  ▶ ▶ -define:F:-finite_field:-q:2:-end:\n  ▶ ▶ -with:F:-do:\n  ▶ ▶ -finite_field_activity:\n  ▶ ▶ -polynomial_division_from_file_all_k_bit_error_patterns:\n  ▶ ▶ ▶ text.csv.13.1-end\n  ▶ ▶ 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:

```
: 0101011010101101101:1010101011110000101111100 0 : 01010110101010101101011100010111101 : 100 : 4 : X^{-2}
1 : 0101011010101011011010111100101111110 : 111 : 7 : X^{-2} + X + 1
2 : 0101011010101010101110001011100010111100 : 001 : 1 : 1
3 : 01010110101010111100010111000010111100 : 000 : 0 : 0
4 : 010101101011011011101011100010111100 : 010 : 2 : X
5 : 01010110101010101110001011110110011100 : 010 : 2 : X^{-2} + X
6 : 010101101010111110010111000010111101 : 011 : 3 : X + 1
7 : 01010110101101110101011110000101111100 : 100 : 4 : X^{-2}
8 : 0101011010101011010111000101111100 : 111 : 7 : X^{-2} + X + 1
9 : 0101011010110101011000101100111100 : 001 : 1 : 1
10 : 0101011010101111110101110000101111100 : 000 : 0 : 0
11 : 010101101010111111010111000101111100 : 010 : 2 : X
```
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\$
   -define:F:-finite_field:-q:2:-end:\$
   -with:F:-do:-finite_field_activity:\$
   -find_CRC_polynomials:3:128:10:-end
```

The program finds 244 polynomials. The execution time is about 1 minute.
9.5 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\cdot\cdot\cdot-vector\cdot\cdot\cdot-dense$$(REED_MULLER_3_1_BASIS_IN_BINARY)\$-end\$
  \$-linear_code_through_basis\-8$\$
  pdflatex\cdot code_n8_k4_q2.tex
  open\cdot 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 9.4.

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:
  $(ORBITER_PATH)orbiter.out\-v\-2\$-code_diagram\"RM_3_1\"$
  $(REED_MULLER_3_1_CODEWORDS)\cdot8$
  -metric_balls\-1
  $(ORBITER_PATH)orbiter.out\-v\-2\$-draw_matrix$
```
Figure 9.5: Boolean function representation of RM$_{3,1}$ in $H(8,2)$

produces a representation of the code as boolean function in the Hamming space $H(8,2)$, shown in Figure 9.5. The different codewords are given different colors.
9.6 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

\[
\text{lcm}(m_{\beta_{\mathbb{F}_q}^1}, m_{\beta_{\mathbb{F}_q}^2}, \ldots, m_{\beta_{\mathbb{F}_q}^{d-1}}).
\]

To create the polynomial \( m_{\beta_{\mathbb{F}_q}^a} \), we consider the \( q \)-cyclotomic set of \( a \) modulo \( n \), which is

\[
\{ aq^i \mod n \mid i \in \mathbb{Z} \}.
\]

Suppose we want to make a BCH-code of length 21 over \( \mathbb{F}_8 \). In Section 3.4, 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_cyclo\text{-}tomic\_mod\_21\_q8:

The output is shown in Figure 9.6. We will try BCH-codes with minimum distances 3, 5 and 7. Here is distance 3:

F_8\_BCH\_code\_d3:

The code is described in a latex output file:

\[
\begin{align*}
\text{BCH-code:} \\
& n = 21, \; k = 17, \; d_0 = 3, \; q = 8, \\
& g(x) = m_1m_2 = X^4 + 4X^3 + 4X^2 + 3X + 4 \\
& \text{Chosen cyclotomic sets:} \\
& \{ 1, 8 \} \\
& \{ 2, 16 \}
\end{align*}
\]
Figure 9.6: The 8-cyclotomic sets modulo 21

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:

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

And now for \(d = 5\):

F_8_BCH_code_d5:

\[
\text{\$\text{(ORBITER_PATH)}\text{orbiter.out} -v.3\}\}
\text{\$\text{-define} F\text{-finite_field} -q 8 -override_polynomial 11\text{-end}\}
\text{\$\text{-with} F\text{-do} \text{-finite_field_activity} \text{-make_BCH_code 21 5\text{-end}}}
\text{\$pdflatex BCH\text{-codes q8 n21 d5.tex}}
\text{\$ open BCH\text{-codes q8 n21 d5.pdf}}
\]

The output file is:

\[
\text{BCH-code:} \\
\text{n = 21, k = 14, d_0 = 5, q = 8,} \\
g(x) = m_1 m_2 m_3 m_4 = X^7 + 3X^6 + 3X^5 + 2X^4 + X^3 + 2X^2 + X + 2 \\
\text{Chosen cyclotomic sets:} \\
\text{\{ 1, 8 \}} \\
\text{\{ 2, 16 \}} \\
\text{\{ 3 \}} \\
\text{\{ 4, 11 \}} \\
\text{The generator polynomial has degree 7}
\]
-dense "2,1,2,1,2,3,3,1"
-sparse "2,0,1,1,2,2,1,3,2,4,3,5,3,6,1,7"

The generator matrix is:
\[
\begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1
\end{bmatrix}
\]

Finally, \( d = 7 \):

```
F_8.BCH_code_d7:
▷ $(ORBITER_PATH) orbiter.out -v.3 \n▷ ▷ -define F: -finite_field -q:8 -override_polynomial:11 -end \n▷ ▷ -with F: -do: -finite_field_activity: -make_BCH_code:21:7 -end
▷ pdflatex BCH_codes_q8_n21_d7.tex
▷ open BCH_codes_q8_n21_d7.pdf
```

The output file is:

```
BCH-code:

\( n = 21, k = 11, d_0 = 7, q = 8 \),
\( g(x) = m_1 m_2 m_3 m_4 m_5 m_6 = X^{10} + X^{9} + 2X^{8} + 5X^{7} + 2X^{6} + 4X^{4} + 6X^{3} + 5X^{2} + 6X + 6 \)

Chosen cyclotomic sets:
{ 1, 8 }
{ 2, 16 }
{ 3 }
```
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:

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

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.

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.
> -cycloctomic_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 9.7.

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
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:
  \$\text{ORBITER\_PATH}\text{orbiter.out\text{-v.3}}\$
  \$\text{-define F\text{-finite_field\text{-q256\text{-end}}}\$
  \$\text{-with F\text{-do\text{-finite_field\_activity\text{-nth_roots\_771\text{-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:
  \$\text{ORBITER\_PATH}\text{orbiter.out\text{-v.3}}\$
  \$\text{-define F\text{-finite_field\text{-q256\text{-end}}}\$
  \$\text{-with F\text{-do\text{-finite_field\_activity\text{-make\_BCH\_code\_771\_16\text{-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.
9.7 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

\begin{verbatim}
CODE_RS_6_4_7="\n621000-\n06210-\n00621-\n000621"
RREF_RS_8_weight_enumerator:
\> $(ORBITER_PATH)orbiner.out-\-v.2-\n\> \> -define:F-\-finite_field-\q.8-\-end-\n\> \> -define:v-\-vector-\-format.5-\-field:F-\n\> \> \> -compact:$\$\(CODE_RS_8)\$-\n\> \> \> -end\n\> \> \> -with:F-\-do-\n\> \> \> \-finite_field_activity-\n\> \> \> \> \-weight_enumerator-\-v-\n\> \> \> \> \-end
\end{verbatim}

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 \( F_8 \) is the cyclic code generated by

\[(X - \alpha)(X - \alpha^2) = X^2 + 6X + 5.\]

The associated cyclic generator matrix is

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

We use the makefile variable \texttt{CODE\_RS\_8} to hold this generator matrix. The following command computes the weight enumerator

\texttt{CODE\_RS\_8}="\$
561000\$
0561000\$
0056100\$
0005610\$
0000561"

\texttt{RREF\_RS\_8\_weight\_enumerator:}
\begin{verbatim}
 $(ORBITER\_PATH)orbiter.out\ -v\ .2\ \n $define F: finite_field = q8\ -end\n $define v: vector = format\( =\) field F\n $compact = $(CODE\_RS\_8)\n $end\n $with = F\ -do\n $finite_field_activity\n $weight\_enumerator\ v\n $end
\end{verbatim}

which turns out to be

\[y^7 + 245x^3 y^4 + 1225x^4 y^3 + 5586x^5 y^2 + 12838x^6 y + 12873x^7.\]

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.

\texttt{RS\_8\_field\_reduction:}
\begin{verbatim}
 $(ORBITER\_PATH)orbiter.out\ -v\ .2\ \n $define F: finite_field = q8\ -end\n $with = F\ -do\n\end{verbatim}
The reduced matrix is shown in Figure 9.8. Let us compute the weight enumerator of the reduced code. The command

\begin{verbatim}
RS_8_reduced="
010001100000000000000000
001110010000000000000000
110011001000000000000000
000010001100000000000000
000001110010000000000000
000110011001000000000000
000000010001100000000000
000000001110010000000000
000000110011001000000000
000000000111010000000000
000000110011001000000000"
\end{verbatim}
computes the weight enumerator of the binary code. It is

\[ \begin{align*}
1 & y^{21} + 28 x^5 y^{18} + 84 x^4 y^{17} + 273 x^5 y^{16} + 924 x^6 y^{15} + 1956 x^7 y^{14} + \\
& 2982 x^8 y^{13} + 4340 x^9 y^{12} + 5796 x^{10} y^{11} + 5796 x^{11} y^{10} + 4340 x^{12} y^9 + \\
& 2982 x^{13} y^8 + 1956 x^{14} y^7 + 924 x^{15} y^6 + 273 x^{16} y^5 + 84 x^{17} y^4 + 28 x^{18} y^3 + \\
& 1 x^{21}
\end{align*} \]

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_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:40-\!-bit_depth:24\$
  ▶ ▶ -partition:4:"15":"21"\$
  ▶ ▶ -end

We compute the weight enumerator

CODE_21_15_4_weight Enumerator:
  ▶ $(ORBITER_PATH)orbiter.out-\!-v.2\$
  ▶ ▶ -define:F:-finite_field:-q:2-\!-end\$
  ▶ ▶ -define:v:-vector:-format:15:-field:F\$
  ▶ ▶ ▶ -compact:$(CODE_21_15_4)\$
  ▶ ▶ -end\$
  ▶ ▶ -with:F:-do\$
  ▶ ▶ -finite_field_activity\$
  ▶ ▶ -weight Enumerator:v\$
  ▶ ▶ -end

which turns out to be

\[1y^{21} + 221x^4 y^{17} + 1600x^6 y^{15} + 6498x^8 y^{13} + 10912x^{10} y^{11} + 9250x^{12} y^9 + 3584x^{14} y^7 + 669x^{16} y^5 + 32x^{18} y^3 + 1x^{20} y.\]

This shows that this code is a \([21, 15, 4]\)_2. It is optimal.
9.8 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 -
  -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 -
  -define:F:-finite_field:-q.2:-end-
  -define:P:-projective_space:8.F:-end-
  -with:P:-do-
  -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 1 1 0 0 0 0 0
1 1 1 1 1 0 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
```

275
To compute the minimum distance of the code, we do:

```
CODE_GV_N15_K6="\n111111111100000\n111110000010000\n111001100001000\n11010101000100\n101010110000010\n101101001000001"
```

```
GV_n15_k6_weight Enumerator:
▷ $(ORBITER_PATH)orbiter.out:-v.2:\n▷ ▷ -define:F:-finite_field:-q:2:-end:\n▷ ▷ -define:v:-vector:-format:6:-field:F:\n▷ ▷ ▷ -compact:$(CODE_GV_N15_K6)\n▷ ▷ -end:\n▷ ▷ -with:F:-do:\n▷ ▷ -finite_field_activity\n▷ ▷ ▷ -weight Enumerator:v\n▷ ▷ -end
```

The weight enumerator is

\[ \frac{1}{y^{15}} + 27x^6 y^9 + 24x^8 y^7 + 9x^{10} y^5 + 3x^{12} y^{3}. \]

From this, we see that the code has minimum distance 6, which is better than predicted.
9.9 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 $\text{PG}(r-1, q)$. 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 $\text{PG}(r-1, q)$ for which any $d-1$-subset (if any) is independent. Under the correspondence, the action of $\text{PGL}(r, q)$ 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 $\text{PGL}(r, q)$ 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

codes_8_4_4:
▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot6\cdot$
▷ ▷ $-\text{orbiter\_path}\cdot(\text{ORBITER\_PATH})\cdot$
▷ ▷ $-\text{define}\cdot G\cdot$
▷ ▷ $-\text{linear\_group}\cdot-PGL\cdot4\cdot2\cdot-end\cdot$
▷ ▷ $-\text{with}\cdot G\cdot-do\cdot$
▷ ▷ $-\text{group\_theoretic\_activity}\cdot$
▷ ▷ $-\text{poset\_classification\_control}\cdot$
▷ ▷ ▷ $-\text{problem\_label}\cdot codes_8_4_4\cdot$
▷ ▷ ▷ $-\text{draw\_poset}\cdot$
▷ ▷ ▷ $-\text{draw\_options}\cdot Embedded\cdot radius\cdot250\cdot$
▷ ▷ ▷ $-\text{line\_width}\cdot1\cdot0\cdot spanning\_tree\cdot end\cdot$
▷ ▷ ▷ $-\text{report}\cdot end\cdot$
▷ ▷ $-end\cdot$
▷ ▷ $-\text{linear\_codes}\cdot3\cdot8\cdot$
▷ ▷ $-end$
▷ $\text{pdflatex\_codes_8_4_4\_poset\_lvl_8.tex}$
▷ $\text{open\_codes_8_4_4\_poset\_lvl_8.pdf}$
▷ $\text{pdflatex\_codes_8_4_4\_poset.tex}$
▷ $\text{open\_codes_8_4_4\_poset.pdf}$

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 $\text{PGL}(4, 2)$ on the poset $\Lambda_{3,3}(2)$. Using poset classification, Orbiter then produces the poset of orbits shown in Figure 9.9. In this diagram, the numbers stand for Orbiter ranks of points in $\text{PG}(3, 2)$. 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 9.9: 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 10

Combinatorics

10.1 Introduction

In Table 10.1, Orbiter commands for combinatorics are shown.

For instance, the command

Char_Sym_4:
$ (\text{ORBITER\_PATH}) \text{orbi} \text{ter.out} -v.2 -\text{character\_table\_symmetric\_group\_4}$

computes the character table of the symmetric group Sym(4):

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-character_table_symmetric_group</td>
<td>$n$</td>
<td>Computes the character table of Sym($n$) using the algorithm of Burnside.</td>
</tr>
<tr>
<td>-make_elementary_symmetric_functions</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>-Dedekind_numbers</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>
</tbody>
</table>

Table 10.1: Combinatorial Commands

281
The character table of Sym(4) is the matrix

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

The command

elementary_symmetric_functions_4:

\$\text{orbiter.out} -\text{make\_elementary\_symmetric\_functions\_4}\$

creates the elementary symmetric functions in 4 variables. The output is:

\[
\begin{align*}
k=1: & \quad x_0 + x_1 + x_2 + x_3 \\
k=2: & \quad x_0 x_1 + x_0 x_2 + x_0 x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 \\
k=3: & \quad x_0 x_1 x_2 + x_0 x_1 x_3 + x_0 x_2 x_3 + x_1 x_2 x_3 \\
k=4: & \quad x_0 x_1 x_2 x_3
\end{align*}
\]
10.2 Diophantine Systems

Diophantine systems of equations and inequalities arise frequently in Combinatorics. In Table 10.2, Orbiter commands for creating and solving diophantine systems are shown. In Table 10.3, 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:

\[
\text{TEST\_SYSTEM}="\\\n0,1,0,1,0,0,\\n0,0,1,0,1,0,\\n1,0,1,0,0,0,\\n0,1,0,1,0,1,\\n1,0,0,0,0,1,\\n1,0,1,0,0,0,\\n0,1,0,0,1,1"
\]

\[
\text{TEST\_RHS}="1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"
\]

There are two commands to solve a diophantine system: \(-\text{solve}\_\text{mckay}\) and \(-\text{solve}\_\text{DLX}\). 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 $F_q$.</td>
</tr>
</tbody>
</table>

Table 10.2: Orbiter Commands to create Diophantine systems
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-print</code></td>
<td></td>
<td>Print the system.</td>
</tr>
<tr>
<td><code>-solve_mckay</code></td>
<td></td>
<td>Solve the system using McKay’s pos-solve.</td>
</tr>
<tr>
<td><code>-solve_DLX</code></td>
<td></td>
<td>Solve the system using Knuth’s dancing links.</td>
</tr>
<tr>
<td><code>-solve_standard</code></td>
<td></td>
<td>Solve the system using the standard solver.</td>
</tr>
<tr>
<td><code>-draw</code></td>
<td></td>
<td>Produce a drawing of the coefficient matrix of the system.</td>
</tr>
<tr>
<td><code>-draw_as_bitmap</code></td>
<td><code>w b</code></td>
<td>Produce a bitmap drawing of the coefficient matrix of the system, using boxes of <code>w</code> pixels with. Set the color bit-depth to <code>b</code> (<code>b = 8</code> or <code>b = 24</code>). The output is a <code>bmp</code>-file.</td>
</tr>
<tr>
<td><code>-perform_column_reductions</code></td>
<td></td>
<td>Eliminate variables which must be zero and write a reduced system.</td>
</tr>
<tr>
<td><code>-test_single_equation</code></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><code>-project_to_single_equation_and_solve</code></td>
<td><code>i j</code></td>
<td>Solve the system assuming the <code>j</code>th solution to the restricted system consisting of the <code>i</code>th row.</td>
</tr>
<tr>
<td><code>-project_to_two_equations_and_solve</code></td>
<td><code>i j r m</code></td>
<td>Solve the system assuming any solution to the restricted system consisting of the <code>i</code>th and the <code>j</code>-th row whose number is congruent to <code>r</code> mod <code>m</code>.</td>
</tr>
</tbody>
</table>

Table 10.3: Orbiter activities for Diophantine systems

285
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
10.3 Combinatorial Linear Spaces

A linear space is a pair \((S, \mathcal{L})\) where \(S\) is a set and \(\mathcal{L}\) is a set of subsets of \(S\) such that each set \(L \in \mathcal{L}\) satisfies \(|L| \geq 2\) and moreover for any two \(a, b \in S\) there is exactly one element \(L \in \mathcal{L}\) 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 \(L \in \mathcal{L}\) of size \(i\). The vector \((a_2, \ldots, a_n)\) is the line type of \((S, \mathcal{L})\). The equation

\[
\binom{n}{2} = \sum_{j=2}^n a_j \binom{j}{2}
\] (10.1)

is satisfied. The equation can be used to generate all possible line types of a putative linear space. Here is an example. For \(|S| = 6\), (10.1) becomes

\[
x_0 \binom{6}{2} + x_1 \binom{5}{2} + x_2 \binom{4}{2} + x_3 \binom{3}{2} + x_4 \binom{2}{2} = \binom{6}{2}.
\]

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

\[
\begin{bmatrix}
15 & 10 & 6 & 3 & 1 & 15
\end{bmatrix}.
\]

The Orbiter command

\[
\begin{align*}
\text{linsp6:} & \quad $(ORBITER\_PATH)orbiter.out\cdot-v\cdot4\cdot$
\end{align*}
\]

solves the system using McKay’s program possolve [49]. The program finds 15 solutions, written to the file \text{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

\[p_j = \#j\text{-lines though } P.\]
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:
  $(ORBITER_PATH)orbiter.out -v 4\n  -define A -vector -format 1 -dense "6,4,3" -end\n  -define D -diophant\n  -label linsp30_pt_types\n  -coefficient_matrix A\n  -RHS "29,29,1" -x_bounds "0,1,0,27,0,24" -end\n  -with D -do\n  -diophant_activity -solve mckay -end
```

we determine the possibilities

\[
(p_7, p_5, p_4) = \begin{pmatrix}
1 & 5 & 1 \\
1 & 2 & 5 \\
0 & 5 & 3 \\
0 & 2 & 7
\end{pmatrix}
\]

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:

\[
\begin{align*}
b_0 + b_1 + b_2 + b_3 &= 30 \\
b_0 + b_1 &= 7 \\
5b_0 + 2b_1 + 5b_2 + 2b_3 &= 135 = 27 \cdot 5 \\
b_0 + 5b_1 + 3b_2 + 7b_3 &= 96 = 24 \cdot 4 \\
10b_0 + b_1 + 10b_2 + b_3 &\leq 351 = \binom{27}{2} \\
10b_1 + 3b_2 + 21b_3 &\leq 276 = \binom{24}{2}
\end{align*}
\]

Using the Orbiter commands
we determine the possibilities

\[
(b_0, b_1, b_2, b_3) = \begin{cases} 
2 & 5 & 23 & 0 \\
3 & 4 & 22 & 1 \\
4 & 3 & 21 & 2 \\
5 & 2 & 20 & 3 \\
6 & 1 & 19 & 4 \\
7 & 0 & 18 & 5 
\end{cases}
\]
10.4 Classification of Configurations and Geometries

A partial linear space is a set system on a fixed set $V$. We write $L = (V, B)$, where $B$ is a set of distinct subsets of $V$, called lines. The members of $V \cup B$ 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 $V \cup B$ such that each $C_i$ either is a subset of $V$ or a subset of $B$. 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 $V \cup B$ 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 $P$. 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 \textit{-geometry\_builder} option can answer these kinds of questions.

The command

geo\_10\_3:
\begin{verbatim}
  \$ (ORBITER\_PATH) orbiter.out -v 2 \\
  -define Test_lines -set -loop 4 -1 -1 -end \\
  -geometry_builder -V 10 -B 10 -TDO 3 -fuse 1 \\
  -fname GEO\_10\_3 \\
  -test Test_lines \\
  -end
\end{verbatim}
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 `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 88 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 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 44 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 44 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 33 37 41 46 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 37 41 46 48 52 54 57 62 66 68 73 75 79 84 86 89 96 97 99
0 1 2 10 13 14 20 25 26 31 33 37 41 46 48 52 54 57 62 66 69 73 76 79 84 86 89 95 97 99
0 1 2 10 13 14 20 25 26 31 33 37 41 46 48 52 54 57 62 66 68 73 76 79 84 86 89 95 97 99
-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 83 103 104 119
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 `-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 `fname_GEO` option. Here, the `fname_GEO` option sets the output file to `10_3`. The `-search_tree` option then creates the file `10_3_tree.txt`. 

291
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  -tree_draw 10_3_tree.txt \n  -draw_options -embedded -radius 50 \n  -xin 10000 -yin 10000 \n  -xout 1000000 -yout 500000 \n  -nodes_empty \n  -scale 0.5 -line_width 0.3 -end \n  $ (ORBITER PATH)orbiter.out -v 2 \n  -draw_options -embedded -radius 50 \n  -xin 10000 -yin 10000 \n  -xout 1000000 -yout 500000 \n  -nodes_empty \n  -scale 0.5 -line_width 0.3 -end \n  -tree_draw 10_3_tree.txt \n  pdflatex 10_3_tree_draw.tex \n  open 10_3_tree_draw.pdf
```

The resulting tree is shown in Figure 10.1.

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
```
Figure 10.1: Search tree for the classification of 10₃ configurations

There is a unique graph with these properties. It is the Petersen graph. Its automorphism group is Sym(5) of order 120.

We can classify configurations with a given girth. For instance, while there are 245342 isomorphism classes of configurations 15₃, 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 15₃:

```
geo_15_3:
  $(ORBITER_PATH)orbiter.out -v 2 \
  -define Test_lines -set -loop 4 16 1 -end \n  -geometry builder \n  -V 15 -B 15 -TDO 3 \n  -fuse 1 -fname GEO 15 3 \n  -test Test_lines \n  -end
```

This command takes about 8 minutes of time to complete. The next command classifies the 15₃ with girth 4. Only one configuration arises, the Cremona Richmond, with automorphism group Sym(6) of order 720.

```
geo_15_3_g4:
  $(ORBITER_PATH)orbiter.out -v 2 \
  -define Test_lines -set -loop 4 16 1 -end \n```

293
-geometry_builder\]
\> \> -V15-B15-TDO3\]
\> \> -fuse1-fname_GEO15_3.g4\]
\> \> -girth4\]
\> \> -test.Test_lines\]
\> \> -end
10.5 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 $\text{PG}(2, q)$. 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:
  $$(\text{ORBITER\_PATH})\text{orbiter.out} -v \cdot 8 \cdot \$$
  $-create\_design -q \cdot 3 \cdot -family \cdot \text{PG} \_2 \cdot q \cdot $$
  $\text{-end}$$

creates the design $\text{PG}(2, 3)$ 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:

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

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

\[
\begin{array}{c|c|c}
\rightarrow & 13_1 & \\
\hline
13_0 & 4 & 13_1 \\
\hline
\end{array}
\]

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

\[M_{t,k}(G) \cdot \mathbf{x} = \lambda \mathbf{1},\]

where \(\mathbf{x}\) is a column vector of zeros and ones and \(\mathbf{1}\) is the column vector of all ones. The length of \(\mathbf{x}\) 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 \(\mathbf{x}\) 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 $7$-$\left(33,8,10\right)$, 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-Mesner matrix has 32 rows and 97 columns. The matrix is stored in the csv-file

```
KM_PGGL_2_32_KM_7_8.csv.
```

The second command produces the graphical representation of the matrix shown in Figure 10.2 (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_PGGL_2_32:
  $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.3\cdot$
  $\cdot$-define-G:-linear\_group:-PGL\_2\_32:-end\cdot$
  $\cdot$-with-G:-do\cdot$
  $\cdot$-group\_theoretic\_activity\cdot$
  $\cdot$-poset\_classification\_control\cdot$
  $\cdot$-problem\_label:KM\_PGGL\_2\_32:-W:-depth\_8\cdot$
  $\cdot$-Kramer\_Mesner\_matrix\_7\_8\cdot$
  $\cdot$-draw\_poset\cdot$
  $\cdot$-draw\_options:-embedded:-sideways:-radius\_50\cdot$
  $\cdot$-scale\_0.5:-line\_width\_0.3:-end\cdot$
  $\cdot$-end\cdot$
```

297
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.
10.6 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.2.1
  ▷   -geometry_builder.1
  ▷   ▷ -V.9.-B.12.1
  ▷   ▷ -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.2.1
  ▷   -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 commnds, 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 $Sym(9)$. The following command does that:

```
AG_2_3_BLOCKS="0,13,22,27,35,41,47,53,55,59,71,76"
```

The number of designs is $|Sym(9)|/432 = 362880/432 = 840$. To find all large sets, we establish the block-disjointness graph on this set of designs and find all cliques of size 7:

```
LS_AG_2_3_disjoint_sets_graph_and_cliques:
  $(ORBITER\_PATH) orbiter.out -v.20\n  -define D -design -list_of_blocks\n  9.3:$ (AG_2_3_BLOCKS) -end\n  -define Sym9 -permutation_group -symmetric_group 9 -end\n  -define T -design_table D:"AG_2_3".Sym9
```

The files $AG_2_3_design_table_disjoint_sets_sol.txt$ and $AG_2_3_design_table_disjoint_sets_sol.csv$ 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:

```
LS_AG_2_3_export_solutions:
  $(ORBITER\_PATH) orbiter.out -v.20\n  -define D -design -list_of_blocks 9.3\n  $(AG_2_3_BLOCKS) -end\n  -define Sym9 -permutation_group -symmetric_group 9 -end\n  -define T -design_table D:"AG_2_3".Sym9\n  -with D -do
```
The final step to classify the large sets up to isomorphism will be discussed in Section 14.4.
10.7 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 $q_1^{d_1} \times q_2^{d_2}$ in product action.

Finite projective planes often arise in this context. However, not all examples are projective planes. Orbiter 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

$$\tau_1 : \mathbb{Z}_3 \to \mathbb{Z}_3, \ x \mapsto x + 1 \mod 3, \ \tau_2 : \mathbb{Z}_7 \to \mathbb{Z}_7, \ y \mapsto y + 1 \mod 7.$$

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:

```
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="\nb_orbits_on_blocks.1\"-depth.5\"
\nDD_PP4:
\> $(\text{ORBITER\_PATH})\text{orbiter.out.-v.6.}\"
\> \> -Delandtsheer_Doyen-$(PP4)$-$(PP4\_GROUP1)$-$(PP4\_MASK1)$-
\> \> \> -end-

DD_PP4_system:
\> $(\text{ORBITER\_PATH})\text{orbiter.out.-v.4.}\"
\> \> -define.D-diophant.-label.PP4-
\> \> \> -problem_of_Steiner_type.10-PP4_pair_covering.csv-
\> \> \> -has_sum.1-
\> \> \> -end-
\> \> \> -with.D-do-
\> \> \> \> -diophant_activity.-solve.mckay-
\> \> \> \> -end
```

302
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 exactly one solution. This must be the PG(2, 4) design. Since there are no more designs, isomorphism testing is not needed.
10.8 Tactical Decompositions

Table 10.4 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

```
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`

```
<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

```
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 ) ( f\text{name} )</td>
<td>Use solutions to system ( i ) from file ( f\text{name} ).</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>( f\text{name} )</td>
<td>Specify the input TDO-file for refinement.</td>
</tr>
</tbody>
</table>

Table 10.4: TDO refinement options
Next, we can compute all coarsest column-tactical refinements of the decomposition. To this end, the command

```
max_arc_16_4_refine:
  ▷ $(ORBITER_PATH)orbiter.out-\text{-v}4-\text{-tdo}\_refinement\·
  ▷ ▷ -\text{-input}\_file-max\_arc\_q16\_r4\_tdo-\text{-dual}\_is\_linear\_space-\text{-end}
```

is used. Because the incidence structure is a projective plane, the dual is a linear space also. Hence the option `-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 `max_arc_q16_r4r.tdo` is produced. Note the added letter `r` at the end of the file name (r for refinement). We can use the following command to display the decomposition stack in the file:

```
max_arc_16_4r_print:
  ▷ $(ORBITER_PATH)orbiter.out-\text{-v}4-\text{-tdo}\_print-max\_arc\_q16\_r4r\_tdo·
```

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
```

```
col_scheme at level 3 :  
is 1 x 2  
    | 221_{ 1} 52_{ 2} |
=================================
273_{ 0} | 17 17
```

```
306
```
col_scheme at level 4:
is 2 x 2

<table>
<thead>
<tr>
<th>(221{1}) &amp; (52{2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>52{0}</td>
</tr>
<tr>
<td>221{3}</td>
</tr>
</tbody>
</table>

extra_col_scheme at level 3:
is 1 x 2

<table>
<thead>
<tr>
<th>(221{1}) &amp; (52{2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>273{0}</td>
</tr>
</tbody>
</table>
Chapter 11

Finite Geometry

11.1 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 $Q = 4$):

```
\text{desarguesian\_spread\_in\_PG\_3\_2:}
\text{▷ \$(\text{ORBITER\_PATH})\text{orbiter.out\_v.3}\$}
\text{▷ \text{▷ \text{-define\_FQ\_finite\_field\_q\_4\_end}\$}
\text{▷ \text{▷ \text{-define\_Fq\_finite\_field\_q\_2\_end}\$}
\text{▷ \text{▷ \text{-with\_FQ\_and\_Fq\_do\_finite\_field\_activity}\$}
\text{▷ \text{▷ \text{-cheat\_sheet\_desarguesian\_spread\_2\_end}}
\text{▷ pdfLaTeX\_Desarguesian\_Spread\_3\_2.tex}
\text{▷ open\_Desarguesian\_Spread\_3\_2.pdf}
```

The cheat sheet contains the following spread:

| Spread element 0 is $(1,0)$: $1 \ 0 \ 0 \ 0$
| Spread element 1 is $(0,1)$: $0 \ 0 \ 0 \ 1$
| Spread element 2 is $(1,1)$: $0 \ 1 \ 0 \ 0$

309
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 $(3, 1) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{22}$

Spread elements by rank: $(0, 34, 9, 17, 22)$.

The following command creates the Desarguesian plane-spread in PG(5, 2):

```
desarguesian_spread_in_PG_5_2:
  $(\text{ORBITER\_PATH})\text{orbiter.out}$\text{-v.3}\text{
  -$\text{-define}\text{\text{-FQ\text{-finite}\text{-field}}\text{-q\text{\text{-}\text{-8}}\text{-end}}}$\text{
  -$\text{-define}\text{-Fq\text{-finite}\text{-field}}\text{-q\text{\text{-2}}\text{-end}}$\text{
  -$\text{-with}\text{-and}\text{-Fq\text{-do}\text{-finite\text{-field}\text{-activity}}}$\text{
  -$\text{-cheat\text{-sheet\text{-desarguesian\text{-spread\text{-2}}-end}}$\text{
  $\text{pdflatex}\text{-Desarguesian\_Spread\_5\_2.tex}$\text{
  $\text{open}\text{-Desarguesian\_Spread\_5\_2.pdf}$
```

Spread element 0 is $(1, 0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{0}$

Spread element 1 is $(0, 1) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}_{1394}$

Spread element 2 is $(1, 1) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}_{189}$

Spread element 3 is $(2, 1) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}_{671}$

Spread element 4 is $(3, 1) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}_{562}$

Spread element 5 is $(4, 1) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{1040}$
Spread element 6 is $(5,1) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

Spread element 7 is $(6,1) = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$

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:
  ▶ $(\text{ORBITER\_PATH})$\text{orbiter.out-v.6}\
  ▶ ▶ -orbiter_path$(\text{ORBITER\_PATH})$\
  ▶ ▶ ▶ -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
  ◀#pdflatex-spreads_4_2_poset_detailed_lvl_17.tex
  ◀#open-spreads_4_2_poset_detailed_lvl_17.pdf
```

classifies the line-spreads of PG(3,4) under the action of PΓL(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 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:

311
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:

\[
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
1 & \omega & 0 & 1 \\
\omega^2 & \omega^2 & \omega & 1
\end{bmatrix}_0,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \omega & \omega^2 \\
0 & 0 & 0 & 1
\end{bmatrix}_1,
\begin{bmatrix}
\omega & 1 & \omega & \omega \\
\omega^2 & 0 & 0 \\
\omega & \omega & \omega^2 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}_0
\]

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:

\[
\begin{bmatrix}
\omega & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & \omega^2 \\
0 & 0 & 0 & 1
\end{bmatrix}_0,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \omega & \omega^2 \\
0 & 0 & 0 & 0
\end{bmatrix}_1,
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & \omega^2 \\
0 & 0 & \omega & 1
\end{bmatrix}_0,
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega^2 & \omega & \omega^2 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}_1,
\begin{bmatrix}
\omega & 1 & \omega & \omega \\
\omega^2 & \omega & 1 & \omega^2 \\
0 & 0 & 1 & \omega^2 \\
0 & 0 & 0 & 1
\end{bmatrix}_0
\]
There are 0 extensions
Number of generators 7

\[ \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & \omega & 1 \\
0 & 1 & \omega & \omega^2 \\
\omega & 1 & 1 & 1
\end{bmatrix}_0 \]

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

\textbf{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:

\[ \begin{bmatrix}
\omega & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
0 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_1, \begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
0 & \omega^2 & 0 & 0 \\
\omega & 0 & \omega & 1 \\
\omega^2 & 0 & 0 & 1
\end{bmatrix}_0, \begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega & 0 & 1 & 1 \\
\omega^2 & 0 & \omega^2 & 1
\end{bmatrix}_0, \begin{bmatrix}
\omega & \omega & \omega & \omega \\
\omega^2 & 0 & \omega^2 & 0 \\
\omega^2 & 0 & \omega^2 & \omega \\
0 & \omega^2 & \omega^2 & 1
\end{bmatrix}_1, \begin{bmatrix}
0 & 0 & \omega^2 & 1 \\
1 & 0 & \omega^2 & 1 \\
1 & 0 & \omega & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}_1, \begin{bmatrix}
0 & \omega^2 & \omega^2 & 0 \\
0 & 0 & 0 & \omega^2 \\
1 & 0 & 1 & \omega^2 \\
\omega & 1 & \omega & 1
\end{bmatrix}_0 \]

1,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,3,1,
1,0,0,0,0,1,0,0,3,0,3,2,1,0,0,2,0,
1,0,0,0,3,1,0,0,3,0,2,2,1,0,1,2,0,
1,1,1,1,2,0,2,0,2,0,2,1,0,2,2,3,0,
1,0,3,1,1,3,1,0,1,0,2,2,0,0,0,1,1,
0,1,1,0,0,0,1,2,0,2,1,3,2,3,2,0,
There are 0 extensions
Number of generators 6
| OCN | |Aut| | Name          |
|-----|---|---|----------------|
| 0   |   |1200|Hall spread    |
| 1   |   |81600|Desarguesian spread |
| 2   |   |576 |Semifield spread |

Table 11.1: Spreads in PG(3, 4) in the Orbiter Catalogue

| OCN | |Aut| | Name |
|-----|---|---|-----|
| 0   |   |1008|     |
| 1   |   |1008|     |
| 2   |   |1728|     |
| 3   |   |216 |     |
| 4   |   |360 |     |
| 5   |   |288 |     |
| 6   |   |3600|     |
| 7   |   |244800|     |

Table 11.2: Spreads in PG(7, 2) in the Orbiter Catalogue

The three spreads in PG(3, 4) can be distinguished by their stabilizer orders. Table 11.1 lists the line spreads in PG(3, 4) according to their orbiter catalogue number (OCN). Table 11.2 lists the solid spreads in PG(7, 2) according to their orbiter catalogue number (OCN).
11.2 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)}\text{orbiter.out}\cdot-v.3\backslash
\text{define}\cdot F\cdot-\text{finite}\_field\cdot q.4\cdot-\text{end}\backslash
\text{define}\cdot \text{PGGL4}\cdot-\text{linear}\_\text{group}\cdot \text{PGGL4}\cdot F\cdot-\text{end}\backslash
\text{define}\cdot \text{PGGL5}\cdot-\text{linear}\_\text{group}\cdot \text{PGGL5}\cdot F\cdot-\text{end}\backslash
\text{with}\cdot \text{PGGL4}\cdot-\text{and}\cdot \text{PGGL5}\cdot-\text{do}\backslash
\text{group}\_\text{theoretic}\_\text{activity}\backslash
\text{Andre_Bruck_Bose_construction}\cdot 0\cdot \text{"TP16-4-HALL"}.\backslash
\text{end}\backslash
$\text{(ORBITER\_PATH)}\text{orbiter.out}\cdot-v.2\cdot-\text{draw}\_\text{matrix}\backslash
\text{input}\_\text{csv}\_\text{file}\cdot \text{TP16-4-HALL}\_\text{incma.csv}\backslash
\text{box}\_\text{width}\cdot 6\cdot-\text{bit}\_\text{depth}\cdot 8\cdot-\text{partition}\cdot 6\cdot 273\cdot 273\cdot \text{end}\backslash
\text{open}\cdot \text{TP16-4-HALL}\_\text{incma}\_\text{draw.bmp}\backslash
\text{pdflatex}\cdot \text{TP16-4-HALL}\_\text{report.tex}\backslash
\text{open}\cdot \text{TP16-4-HALL}\_\text{report.pdf}
```

creates the Hall plane of order 16. Remember from Table 11.1 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:
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:
\[
\begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 1 & \omega^2
\end{bmatrix}
\]

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:
\[
\begin{bmatrix}
1 & 0 & \omega^2 & 1 \\
0 & 1 & \omega & 1
\end{bmatrix}
\]

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

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

317
\[
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega & \omega^2 & 0 & 0 \\
0 & \omega^2 & \omega & 0 \\
\omega^2 & 0 & \omega & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
, \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
1 & 0 & \omega^2 & 0 \\
\omega & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
, \quad
\begin{bmatrix}
1 & \omega^2 & \omega & 1 \\
0 & \omega^2 & 0 & 1 \\
\omega & 1 & \omega^2 & 0 \\
0 & 1 & \omega^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

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

Tactical decomposition schemes:

<table>
<thead>
<tr>
<th></th>
<th>80₁</th>
<th>192₂</th>
<th>₁₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>256₀</td>
<td>5</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>₅₃</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>192₂</th>
<th>₁₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>256₀</td>
<td>16</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>₅₃</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>
11.3 Packings

A packing of PG(3, q) is a set of pairwise line-disjoint spreads of 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 11.3 list Orbiter commands related to the construction of packings with assumed symmetry.

Table 11.4 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 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 PG(3, 5). Let us construct these packings. We start by making a table of all regular packings:

spread_table_PG_3_5_regular:

There are 155,000 packings. In the command, we rely on the classification of spreads in 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

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

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</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</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</td>
<td>descr</td>
<td>Proceed on to long orbits using Table 11.4.</td>
</tr>
<tr>
<td>-spread_tables_prefix</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 11.3: 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_from_file</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>-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 11.4: Orbiter commands related to the construction of packings with assumed symmetry related to picking long orbits
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:

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:

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:
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!

```
PG_3_5_assume_31_fpc0_lo_cliques:
  $(ORBITER_PATH)orbiter.out -v 2
  -define G -graph -load_from_file H31_fpc0_lo.graph -end
  -with G -do
    -graph_theoretic_activity
    -find_cliques -target_size 1 -end
  -print_symbols
```

There are exactly 8 cliques of size 1. The next command builds the packings arising from these 8 cliques:

```
PG_3_5_assume_31_read:
  $(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
    -H "H31" $(PGL_4_5_SUBGROUP_31_ME) -end
    -N "H31" $(PGL_4_5_SUBGROUP_31_ME) -end
  -end
  -define PW -packing_choose_fixed_points PW 0 -end
  -define L -packing_long_orbits PW -end
  -orbit_length 31 -clique_size 1
  -read_solutions
  -end
```

The next command classifies the 8 packings up to isomorphism, using Nauty:

```
PG_3_5_assume_31_classify:
  $(ORBITER_PATH)orbiter.out -v 2
```
There are exactly 2 isomorphism classes of packings. These are of course the examples found by Prince and generalized by Penttila and Williams. The packings are invariant under a group of order 93.
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-catalogue</td>
<td>$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 11.6 for possibilities for $F$.</td>
</tr>
</tbody>
</table>

Table 11.5: Commands for creating BLT-sets

### 11.4 BLT-Sets

A BLT-set of $Q(4, q)$ is a set of $q + 1$ points 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 11.5 shows options to create known BLT-sets. Table 11.6 shows options for known families or sporadic sets. For instance, the command

```
BLT_11.0:
  > $(ORBITER_PATH)orbiter.out:-v.2:\
  >   > -define:F:-finite_field:-q.11:-end:\
  >   > -define:O:-orthogonal_space:0.5:F:-end:\
  >   > -with:O:do:-orthogonal_space_activity:\
  >   >   > -create_BLT_set:-catalogue:0:-end:\
  >   > -end
  > #pdflatex:0.1.6.2_report.tex:
  > #open:0.1.6.2_report.pdf
```

creates the BLT-set #0 in $Q(4, 11)$. The command

```
BLT_11_Mondello:
  > $(ORBITER_PATH)orbiter.out:-v.2:\
  >   > -define:F:-finite_field:-q.11:-end:\
  >   > -define:O:-orthogonal_space:0.5:F:-end:\
  >   > -with:O:do:-orthogonal_space_activity:\
```

325
<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>$q \equiv \pm 1 \mod 10$</td>
<td>Mondello BLT-set due to Penttila [51].</td>
</tr>
<tr>
<td>Mondello</td>
<td></td>
<td>Fisher BLT-set [26].</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>$q = p^e, e &gt; 1$</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 11.6: Families of BLT-sets

creates the Mondello BLT-set in $Q(4, 11)$. Orbiter creates the following report:

The quadratic form is:

$$X_0^2 + X_1X_2 + X_3X_4 = 0$$

The BLT-set is:
<table>
<thead>
<tr>
<th>i</th>
<th>Rank</th>
<th>Point</th>
<th>(a, b, c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>846</td>
<td>(1, 6, 4, 10, 3)</td>
<td>(22, 11, 1)</td>
</tr>
<tr>
<td>1</td>
<td>851</td>
<td>(1, 5, 7, 10, 3)</td>
<td>(22, 110, 1)</td>
</tr>
<tr>
<td>2</td>
<td>1234</td>
<td>(1, 5, 1, 7, 7)</td>
<td>(37, 11, 1)</td>
</tr>
<tr>
<td>3</td>
<td>613</td>
<td>(1, 6, 10, 5, 1)</td>
<td>(73, 110, 1)</td>
</tr>
<tr>
<td>4</td>
<td>1307</td>
<td>(1, 1, 3, 8, 5)</td>
<td>(59, 36, 1)</td>
</tr>
<tr>
<td>5</td>
<td>1418</td>
<td>(1, 3, 9, 6, 10)</td>
<td>(95, 36, 1)</td>
</tr>
<tr>
<td>6</td>
<td>1022</td>
<td>(1, 9, 5, 10, 2)</td>
<td>(99, 96, 1)</td>
</tr>
<tr>
<td>7</td>
<td>835</td>
<td>(1, 2, 6, 3, 3)</td>
<td>(99, 36, 1)</td>
</tr>
<tr>
<td>8</td>
<td>950</td>
<td>(1, 10, 8, 2, 9)</td>
<td>(95, 96, 1)</td>
</tr>
<tr>
<td>9</td>
<td>789</td>
<td>(1, 8, 2, 4, 4)</td>
<td>(59, 96, 1)</td>
</tr>
<tr>
<td>10</td>
<td>611</td>
<td>(1, 7, 7, 5, 1)</td>
<td>(73, 11, 1)</td>
</tr>
<tr>
<td>11</td>
<td>1236</td>
<td>(1, 4, 4, 7, 7)</td>
<td>(37, 110, 1)</td>
</tr>
</tbody>
</table>

Plane intersection type is $4^{18}$ $3^{148}$

Plane invariant is too big (18 planes)

\[
\begin{array}{c|c|c}
\rightarrow & 18_{1} & \downarrow 18_{1} \\
12_{0} & 6 & 12_{0} 4 \\
\end{array}
\]

\[
C_{0} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}_{12}
\]

\[
C_{1} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}_{18}
\]

\[
\begin{array}{c|c|c}
\rightarrow & 18_{1} & \downarrow 18_{1} \\
12_{0} & 6 & 12_{0} 4 \\
\end{array}
\]

\[
C_{0} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}_{12}
\]

\[
C_{1} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}_{18}
\]

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 $Q(4, 13)$:

```
BLT_13_deep_14:
  $\$(ORBITER\_PATH)orbit\_out\_v.2\$
  $\$-define\_F\_finite\_field\_q.13\_end\$
  $\$-define\_O\_orthogonal\_space\_0.5\_F\_end\$
  $\$-with\_O\_do\_orthogonal\_space\_activity\$
  $\$-BLT\_set\_starter\_14\_problem\_label\_BLT\_q13\_W\_depth\_14\_end\$
```
end
Chapter 12

Graph Theory

12.1 Creating Graphs

Table 12.1 shows some Orbiter commands to create graphs.

For instance, the command

Cycle_13:
> $(ORBITER_PATH)oriter.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)oriter.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><code>-load</code></td>
<td>filename</td>
<td>Read a graph from file in Orbiter format.</td>
</tr>
<tr>
<td><code>-load_csv_no_border</code></td>
<td>filename</td>
<td>Read a graph from a csv file. Ignore the first row and first column.</td>
</tr>
<tr>
<td><code>-load_dimacs</code></td>
<td>filename</td>
<td>Read a graph from file in dimacs format.</td>
</tr>
<tr>
<td><code>-edge_list</code></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><code>-edges_as_pairs</code></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><code>-cycle</code></td>
<td>n</td>
<td>Cycle graph on n vertices.</td>
</tr>
<tr>
<td><code>-Hamming</code></td>
<td>n q</td>
<td>Hamming graph $H(n,q)$</td>
</tr>
<tr>
<td><code>-Johnson</code></td>
<td>n k s</td>
<td>Johnson graph</td>
</tr>
<tr>
<td><code>-Paley</code></td>
<td>q</td>
<td>Paley graph</td>
</tr>
<tr>
<td><code>-Sarnak</code></td>
<td>p q</td>
<td>Lubotzky-Phillips-Sarnak graph [45]</td>
</tr>
<tr>
<td><code>-Schlaefli</code></td>
<td>q</td>
<td>Schlaefli graph</td>
</tr>
<tr>
<td><code>-Shrikhande</code></td>
<td></td>
<td>Shrikhande graph</td>
</tr>
<tr>
<td><code>-Winnie_Li</code></td>
<td>q i</td>
<td>Winnie-Li graph [44]</td>
</tr>
<tr>
<td><code>-Grassmann</code></td>
<td>n k q r</td>
<td>Grassmann graph</td>
</tr>
<tr>
<td><code>-coll_orthogonal</code></td>
<td>$\epsilon$ d q</td>
<td>Collinearity graph of $O^\epsilon(d,q)$</td>
</tr>
<tr>
<td><code>-triherited_pair_disjointness_graph</code></td>
<td></td>
<td>Triheral pair disjointness graph</td>
</tr>
<tr>
<td><code>-non_attacking_queens_graph</code></td>
<td>n</td>
<td>Create the graph for non-attacking queens on a $n \times n$ chess board.</td>
</tr>
<tr>
<td><code>-subset</code></td>
<td>label labeltex subset</td>
<td>Define vertex coloring with two colors based on a subset of vertices.</td>
</tr>
<tr>
<td><code>-disjoint_sets_graph</code></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><code>-orbital_graph</code></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><code>-collinearity_graph</code></td>
<td>inc-matrix</td>
<td>Collinearity graph of the given incidence matrix.</td>
</tr>
<tr>
<td><code>-chain_graph</code></td>
<td>P1 P2</td>
<td>Chain graph with respect to the partitions P1 and P2.</td>
</tr>
<tr>
<td><code>-Cayley_graph</code></td>
<td>$G$ gens</td>
<td>Cayley graph with respect to group $G$ and generating set gens.</td>
</tr>
</tbody>
</table>

Table 12.1: Orbiter commands to define graphs

330
Chain_232:
▷ $(ORBITER\_PATH)\text{orbiter.out}\cdot-v\cdot2\cdot$
▷ ▷ -define\(P1\cdot-vector\cdot-dense\cdot2,3,2\cdot-end\cdot$
▷ ▷ -define\(P2\cdot-vector\cdot-dense\cdot2,3,2\cdot-end\cdot$
▷ ▷ -define\(Gamma\cdot-graph\cdot$
▷ ▷ ▷ -chain_graph\(P1\cdotP2\cdot$
▷ ▷ -end

creates the chain graph with respect to the partitions \((2,3,2)\) and \((2,3,2)\).

The command

Paley_13_graph:
▷ $(ORBITER\_PATH)\text{orbiter.out}\cdot-v\cdot2\cdot$
▷ ▷ -define\(Gamma\cdot-graph\cdot-Paley\cdot13\cdot-end\cdot$

creates the Paley graph on 13 vertices.

The command

triheiral_pair_graph:
▷ $(ORBITER\_PATH)\text{orbiter.out}\cdot-v\cdot2\cdot$
▷ ▷ -define\(Gamma\cdot$
▷ ▷ ▷ -graph\cdot-triheiral_pair_disjointness_graph\cdot$
▷ ▷ -end

creates the graph of trihedral pairs. Two vertices are adjacent if the associated trihedral pairs are line-disjoint.

The command

small_graph:
▷ $(ORBITER\_PATH)\text{orbiter.out}\cdot-v\cdot2\cdot$
▷ ▷ -define\(G\cdot-graph\cdot-edges\_as\_pairs\cdot5\cdot"0,1,0,2,0,3,0,4,1,3,1,4,2,4\cdot"\cdot-end
creates a graph by listing the edges in pairs. In this case, the graph

is created.

The command

```
petersen:
$\text{define}\cdot G\cdot -\text{graph}\cdot -\text{Johnson}\cdot 5\cdot 2\cdot 0\cdot -\text{end}
```

creates the Petersen graph.

The command

```
Johnson\_6\_2\_0:
$\text{define}\cdot G\cdot -\text{graph}\cdot -\text{Johnson}\cdot 6\cdot 2\cdot 0\cdot -\text{end}
```

creates the Johnson graph $J(6, 2, 0)$.

The command

```
Hamming\_graph\_3:
$\text{define}\cdot G\cdot -\text{graph}\cdot -\text{Hamming}\cdot 3\cdot 2\cdot -\text{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 12.2: Orbiter commands to modify graphs

In Section 14.7, 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-
  -define gens=-vector=-file-
  -load_csv_no_border-
  -file halljanko315.csv-
  -end-
  -define Gamma=-graph-
  -orbital_graph G.3-
  -end-
```

Table 12.2 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-
```
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 $\mathbb{F}_{11}$. This means that the map $x \mapsto ax + b \text{ mod } 11$ is encoded as a pair $(a, b)$.

Cayley $\mathbb{Z}_{11}$ 1mod3:

```
$($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".1\-1\"1,1".\-
   -end\-
   -define\-Gamma\-graph\-
   -Cayley\-graph\-G\-S\-
   -end.
```

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:

```
$($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.
```

And one more example, using the same group but a different connection set. This graph is called the star graph:

Cayley Sym4_star:

```
$($ORBITER_PATH)orbiter.out-\-v\-2\-
```

334
-define S:vector: dense:"1,0,2,3,2,1,0,3,3,1,2,0"-end-
-define G:permutation_group:symmetric_group:4-
-define Gamma:graph-
-define Cayley_graph:G:S-
-define end-
<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 12.4.</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 14.7.</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 12.3: Graph Theoretic Activities

12.2 Graph Theoretic Activities

Graph theoretic activities allow us to perform tasks on graphs. Table 12.3 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

```
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
```

336
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:
▷ $(\text{ORBITER\_PATH})\text{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 - \\
▷ $(\text{ORBITER\_PATH})\text{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_9_eigenvalues.tex - \\
▷ #open Cycle_9_eigenvalues.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:
▷ $(\text{ORBITER\_PATH})\text{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:}

\begin{verbatim}
\$(ORBITER\_PATH)orbiter.out\ -v\ 2 \-
\define\ G\ -graph\ -Johnson\ 5\ 2\ 0\ -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
\end{verbatim}

\begin{verbatim}
\$(ORBITER\_PATH)orbiter.out\ -v\ 2\ -draw\ matrix\ \\
\-input\ csv\ file\ Johnson\ 5\ 2\ 0.csv\ \\
\-box\ width\ 40\ -bit\ depth\ 24\ -partition\ 4\ "10"\ "10"\ -end
\end{verbatim}

\texttt{dot\ -Tpng\ Johnson\ 5\ 2\ 0.gv\ -o 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:}

\begin{verbatim}
\$(ORBITER\_PATH)orbiter.out\ -v\ 2 \-
\define\ G\ -graph\ -edges\ as\ pairs\ 5\ "0,1,0,2,0,3,0,4,1,3,1,4,2,4"\ -end\ \\
\-with\ G\ -do\ \\
\-graph\ theoretic\ activity\ -export\ csv\ -end\ \\
\end{verbatim}
draws the small graph created in Section 12.1 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:

```
P5G_2_collinearity_graph::O5_2.incidence_matrix.csv.
> $(ORBITER_PATH)orbiter.out-v.3-
> -define:Inc-vector-file:O5_2.incidence_matrix.csv-end-
> -define:Gamma-graph-collinearity_graph:Inc-end-
> -with:Gamma-do-
> -graph_theoretic_activity-
> -properties-
> -end
```

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 us consider again the Cayley graphs from Section 12.1. Here is a command that draws the first graph:

```
Cayley_Z11_1mod3_draw:
> $(ORBITER_PATH)orbiter.out-v.2-
> -draw_options-xin:2000000-
> -yin:2000000-embedded-radius:20000-end-
> -define:F:finit_field:q:11-end-
> -define:S:vector-dense-
> "1,1,1,4,1,7,1,10"-end-
> -define:G:linear_group:AGL:1:F-
> -define:G:subgroup_by_generators:"Z11":11:1:"1,1"-
> -end-
> -define:Gamma:-graph-
> -Cayley_graph:G:S-
```
The drawing is shown in Figure 12.1. Let us draw the Cayley graph of Sym(5) with respect to the Coxeter generators:

```plaintext
Cayley_Sym5_coxeter_draw:
$($ORBITER\_PATH)\text{or}b\text{itr.out}\cdot-v.2\$
$-$draw_options:\-xin.1000000\-yin.1000000\$
$-$embedded\-radius.10000\-nodes_empty\-end\$
$-$define\:S\-vector\-dense\$
$>$"1,0,2,3,4,0,2,1,3,4,0,1,3,2,4,0,1,2,4,3"\-end$
$-$define\:G\-permutation_group\-symmetric\_group\:5\$
$\$-end$
$>$-define\:Gamma\-graph$
$>$-Cayley_graph\:G\:S$
$\$-end$
$>$-with\:Gamma\-do$
$>$-graph\:theoretic\_activity\-draw\-end$
```

pdflatex\:Cayley_graph\_AGL\_1\_11\_draw.tex
open\:Cayley_graph\_AGL\_1\_11\_draw.pdf
Figure 12.2: The Cayley graph of Sym(5) w.r.t. the Coxeter generators

The drawing is shown in Figure 12.2.
<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 12.4: Options for classifying graphs

### 12.3 Classification of Graphs and Tournaments

Table 12.4 lists the Orbiter commands to classify graphs and tournaments. The following command classifies all graphs on 4 vertices:

```shell
graph_classify_4:
   ▶ $(ORBITER_PATH)orbiter.out-v.2\n   ▶ ▶ -define GC -graph_classification\n   ▶ ▶ ▶ -n 4\n   ▶ ▶ ▶ -poset_classification_control\n   ▶ ▶ ▶ ▶ -problem label graphs_v4-depth 6-draw poset\n   ▶ ▶ ▶ ▶ -draw_options -radius 250-embedded-end\n   ▶ ▶ ▶ -end\n   ▶ ▶ -end\n   ▶ ▶ -with GC do\n   ▶ ▶ -graph_classification_activity\n   ▶ ▶ ▶ -draw_options -embedded -radius 400\n   ▶ ▶ ▶ ▶ -line width 2-scale 0.15-end\n   ▶ ▶ ▶ -draw_graphs at level 3\n   ▶ ▶ ▶ -end\n   ▶ ▶ -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:

```shell
tournament_classify_4:
   ▶ $(ORBITER_PATH)orbiter.out-v.2\n   ▶ ▶ -define GC -graph_classification\n   ▶ ▶ ▶ -n 4-tournament\n```

342
Figure 12.3: The four isomorphism types of tournaments on 4 vertices

The next command classifies all cubic graphs on 8 vertices:

```plaintext
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.-draw_options.at_level.12-
  -draw_options.-radius.400.-draw_options.-line_width.2.-scale.0.15.-end-
  -draw_graphs_at_level.12-
  -print_symbols
```

Figure 12.3 shows the resulting list of 4 tournaments.
<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>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 12.5: Clique Finding Options

### 12.4 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 12.3 can be used to find all cliques in a graph. Additional options for this command are shown in Table 12.5. For instance, the cliques of size 3 in the graph `graph_v5_e7.colored_graph` from Section 12.1 can be found using

```
small_graph_cliques:
  $$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.10\cdot$
  $$(\text{ORBITER\_PATH})\text{define}\cdot G\cdot \text{graph}\cdot \text{load\_from\_file}\cdot \text{graph\_v5\_e7\_colored\_graph}\cdot \text{-end}\cdot$
  $$(\text{ORBITER\_PATH})\text{-with}\cdot G\cdot \text{-do}\cdot$
  $$(\text{ORBITER\_PATH})\text{-graph\_theoretic\_activity}\cdot \text{-find\_cliques}\cdot \text{-target\_size}\cdot 3\cdot \text{-end}\cdot$
```

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:

```
Paley_13_aut:
  $$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2\cdot$
  $$(\text{ORBITER\_PATH})\text{-define}\cdot \text{Gamma}\cdot \text{-graph}\cdot \text{-Paley}\cdot 13\cdot \text{-end}\cdot$
  $$(\text{ORBITER\_PATH})\text{-with}\cdot \text{Gamma}\cdot \text{-do}\cdot$
  $$(\text{ORBITER\_PATH})\text{-graph\_theoretic\_activity}\cdot \text{-automorphism\_group}\cdot$
```

344
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
-define G permutation group
-bsgs Paley_13 "Paley_13".13.78 "0,1".3 gens
```

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
-define G permutation group
-bsgs Paley_13 "Paley_13".13.78 "0,1".3 gens
-Gamma graph Paley_13
-with G do
-group theoretic activity
-poset classification control
-\W
-problem label Paley13 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
-with Gamma do
-graph theoretic activity find cliques target size 3
```

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 12.1. 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)orborder.out -v -6 -
  -define Gamma -graph -
  -load -
  -define Group_Perm315_Orbital_3.colored_graph -
  -end -
  -with Gamma -do -
  -graph_theoretic_activity -
  -define gens -vector -
  -find_cliques -target_size 5 -
  -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)orborder.out -v -6 -
  -define Gamma -graph -
  -load -
  -define Group_Perm315_Orbital_3.colored_graph -
  -end -
  -define gens -vector -
  -file halljanko315_gens.csv -
  -end -
  -define G -permutation_group -
  -bsgs halljanko315 "File\ halljanko315" -
  315:1209600 "0,1,42,95" -gens -end -
  -with G -do -
  -group_theoretic_activity -
  -poset_classification_control -
  -problem_label HJ64_cliques -
  -clique_test Gamma -
  -depth 5 -
  -end -
  -orbits_on_subsets 5 -
  -report -
  -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\$
▷ ▷ -define:Inc:-vector:-file:0_5_2.incidence_matrix.csv:-end:\$
▷ ▷ -define:Gamma:-graph:-collinearity_graph:Inc:-end:\$
▷ ▷ -with:Gamma:-do:\$
▷ ▷ -graph_theoretic_activity:\$
▷ ▷ ▷ -find_cliques:-target_size:3:-end:\$
▷ ▷ -end
```

There are 15 cliques of size 3. They correspond to the lines in the geometry.
Chapter 13

Combinatorial Objects

13.1 Combinatorial Objects

Sometimes, it is useful to have a unified behavior of different types of objects. This eliminates the need to memorize different versions of a command for different but similar objects. Combinatorial objects are an example for this situation. Many different types of objects exist, but they all share common functionality such as the possibility of computing the canonical form. So, in Orbiter, a combinatorial object is an abstraction of different types of objects. By treating different classes of objects under one umbrella, it becomes possible to apply activities for these different classes of objects in a unified fashion. Most notably, classification by canonical forms has a unified interface, but varying implementations.

Table 13.1 summarizes the commands that can be used to define combinatorial object(s). The objects are coded. They can be given in the command line, or they can be stored in a file. The commands can be repeated and mixed, as long as the objects are of the same type. The file types will be discussed in more detail in the next section. Here are some examples. First, we create the Hirschfeld surface from a set object. The set-object in turn is created from a makefile variable containing the Orbiter point ranks of the surface:

```plaintext
HIRSCHFELD_SURFACE_Q4_SET_OF_POINTS="0,1,2,3,4,5,6,7,8,9,\n10,11,12,13,14,23,26,27,30,31,34,35,38,39,42,47,48,51,52,\n53,54,59,60,61,62,67,68,69,70,75,76,79,80,81,82"

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:
  >   -end
```

349
<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_packing</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:

```
HYPEROVAL_16_144="0, 1, 2, 3, 52, 67, 89, 106, 126, 141, 159, 176, 184, 199, 220, 235, 245, 262"
HYPEROVAL_16_16320="0, 1, 2, 3, 52, 70, 83, 109, 127, 139, 156, 174, 186, 199, 217, 229, 256, 264"
```

```
hyperoval_16_create:
  > $(ORBITER_PATH)oribter.out -v 2\n  > -define C -combinatorial_objects\n  > -set_of_points $(HYPEROVAL_16_16320)\n  > -set_of_points $(HYPEROVAL_16_144)\n  > -end\n```

Next, we read the points of an elliptic curve from file (see Section 4.2):

```
EC_read:: elliptic_curve_b1_c3_q11.txt
  > $(ORBITER_PATH)oribter.out -v 4\n  > -define C -combinatorial_objects\n  > -file_of_points elliptic_curve_b1_c3_q11.txt\n  > -end
```

Next, we read a packing, using a previously defined table of spreads, stored in a csv file.

```
PG_3_5_assume_31_read:
  > $(ORBITER_PATH)oribter.out -v 2\n  > -define C -combinatorial_objects\n  > -file_of_packings_through_spread_table\n  > -H31_packings.csv\n  > -SPREAD_TABLES_5_REG/spread_25_spreads.csv\n  > -5\n  > -end
```

The following command reads a file of large sets of designs:

```
LS_AG_2_3_read:
  > $(ORBITER_PATH)oribter.out -v 2\n  > -define C -combinatorial_objects\n  > -file_of_designs\n  > solutions.csv 9 84 3 12\n  > -end
```

The next command reads incidence geometries from a file containing the flags:
geo_7_3_read:
▷ $(ORBITER_PATH)orbiter.out \-v \cdot 10 \cdot \$
▷ ▷ -draw_incidence_structure_description \-
▷ ▷ ▷ -width \cdot 60 \-with \cdot 10 \cdot 6 \-end \-
▷ ▷ ▷ -define \cdot \cdot C \cdot \cdot \cdot -combinatorial_objects \-
▷ ▷ ▷ ▷ 7 \cdot 3 \cdot inc \cdot 7 \cdot 7 \cdot 21 \-
▷ ▷ -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 \cdot 10 \-
▷ ▷ -draw_incidence_structure_description \-
▷ ▷ ▷ -width \cdot 60 \-with \cdot 10 \cdot 6 \-end \-
▷ ▷ ▷ -define \cdot \cdot C \cdot \cdot \cdot -combinatorial_objects \-
▷ ▷ ▷ ▷ -file_of_incidence_geometries_by_row_ranks \-
▷ ▷ ▷ ▷ Desargues_path_lex_least.inc \cdot 10 \cdot 10 \cdot 3 \-
▷ ▷ -end
13.2 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 13.1. 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 13.2. 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 13.3). 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
```
<table>
<thead>
<tr>
<th>ℓ₀</th>
<th>ℓ₁</th>
<th>ℓ₂</th>
<th>ℓ₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>P₁</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>P₂</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P₃</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P₄</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>P₅</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 13.2: The incidence matrix of the Pasch configuration

<table>
<thead>
<tr>
<th>ℓ₀</th>
<th>ℓ₁</th>
<th>ℓ₂</th>
<th>ℓ₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>P₁</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>P₂</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>P₃</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>P₄</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>P₅</td>
<td>20</td>
<td>21</td>
<td>22</td>
</tr>
</tbody>
</table>

Figure 13.3: 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_pasc_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)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
Chapter 14
Canonical Forms with Nauty

14.1 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 14.1: Orbiter commands related to canonical labelings

<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>

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 14.1 list Orbiter commands related to canonical labelings of combinatorial objects.
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.

```
EC_canon: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\ 
  -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\ 
  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
```

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:
  -file_of_points Hirschfeld_surface_q4.txt:
  -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:
  -max_TDO_depth 10:
  -end:
  -report:
  -show_TDO:
  -end:
  -end
```

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):
```

$$
\begin{bmatrix}
1 & 0 & 0 \\
0 & 10 & 0 \\
0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
1 & 1 & 8 \\
5 & 9 & 5 \\
8 & 1 & 1
\end{bmatrix}.
$$
▷▷▷ -set_of_points:$(HYPEROVAL_{16\_144})\$
▷▷▷ -end\$
▷▷▷ -define:F:-finite_field:-q.16:-end\$
▷▷▷ -define:P:-projective_space:2:F:-end\$
▷▷▷ -with:C:-do\$
▷▷▷ -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
▷ pdflatex:hyperoval_q16_classification.tex
▷ open:hyperoval_q16_classification.pdf
▷ $(ORBITER\_PATH)\$\text{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
▷ open:hyperoval_q16_object0\_TDA_flag_orbits_draw.bmp
▷ $(ORBITER\_PATH)\$\text{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
▷ open:hyperoval_q16_object1\_TDA_flag_orbits_draw.bmp
14.3 Canonical Forms of Incidence Geometries

Let us consider 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 gometry.
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 gometry.

We will also be producing drawings of the incidence matrices of geometries. In these draw-
ings, 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
PG(2, 2), which is a configuration $7_3$.

```
geo_7_3.c:
  $ (ORBITER_PATH)orbiter.out -v 10 \n  -draw incidence structure description \n  -width 60 -with 10.6 -end \n```
A bitmap drawing is produced, as shown in Figure 14.1. The command also produces the following report of the geometry:

<table>
<thead>
<tr>
<th>Rep</th>
<th># Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1 168 0</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 94123.

```
AG_2_3_c::AG_2_3.inc
  $(ORBITER_PATH)orbiter.out -$v-2\$
  $define C=:-combinatorial_objects$\$
  $define C=:-file_of_incidence_geometries$\$
  $define C=:-AG_2_3.inc-9-12-36$\$
  $define C=:-end$\$
```
Figure 14.2: The affine plane AG(2, 3) is a configuration $9_4^{12_3}$.

A bitmap drawing is produced, shown in Figure 14.2. 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>#</th>
<th>Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>432</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:

\[
\begin{array}{c|c}
9_0 & 4 \\
\hline
9_0 & 12_1 \\
\hline
9_0 & 3 \\
\end{array}
\]
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 infact 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 combinatorial_objects command. The classification algorithm can print a report which lists the transversal and all elements in it in latex form.

geo_10_3.c:
  > $(ORBITER_PATH)orbiter.out,-v,2,\n  > -define:C,-combinatorial_objects,\n  > -file_of_incidence_geometries:10_3.inc,10.10.30,\n  > -end,\n  > -with:C,-do,\n  > -combinatorial_object_activity,\n  > -canonical_form,\n  > -classification_prefix:10_3,\n  > -save_ago,\n  >
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>#</th>
<th>Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>6</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 )

369
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:
{1}

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:

The following command computes the canonical form for the three triangle free configurations

371
found by Abdullah Alazemi. These configurations have 24 points, 24 lines, each line consists of 3 points and each point is on 3 lines.

\begin{verbatim}
TFC_24_3_c:
  > echo $(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
\end{verbatim}

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 14.3. Because the geometry is flag transitive, there is only one color being used for the incidence. In fact, all incidences are in black.
Figure 14.3: A flag transitive $24_3$ configuration
14.4 Canonical Forms of Objects from Design Theory

In Secton 10.6, some large sets of $\text{AG}(2, 3)$ were constructed. The final isomorphism classification is performed using the Nauty interface. A list of combinatorial objects is created, and the `-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 $\text{AG}(2, 3)$.

```
LS_AG_2_3_solutions_classify:
  ▶ $(\text{ORBITER\_PATH})\text{orbiter}\text{.out}\cdot-v\cdot2\cdot$
  ▶ ▶ -draw\_incidence\_structure\_description\cdot$
  ▶ ▶ ▶ -width\cdot30\cdot-with\cdot10\cdot3\cdot-end\cdot$
  ▶ ▶ -define\cdotC\cdot-combinatorial\_objects\cdot$
  ▶ ▶ ▶ -file\_of\_designs\cdot$
  ▶ ▶ ▶ solutions\_csv\cdot9\cdot84\cdot3\cdot12\cdot$
  ▶ ▶ -end\cdot$
  ▶ ▶ -with\cdotC\cdot-do\cdot$
  ▶ ▶ -combinatorial\_object\_activity\cdot$
  ▶ ▶ ▶ -canonical\_form\cdot$
  ▶ ▶ ▶ -save\_ago\cdot$
  ▶ ▶ ▶ -classification\_prefix\cdotlarge\_sets\_of\_AG\_2\_3\cdot$
  ▶ ▶ ▶ -end\cdot$
  ▶ ▶ ▶ -report\"classification\"\cdot$
  ▶ ▶ ▶ -end
  ▶ pdflatex large\_sets\_of\_AG\_2\_3\_classification\_tex
  ▶ open large\_sets\_of\_AG\_2\_3\_classification\_pdf
```

It turns out that there are exactly two isomorphism types, with automorphism groups of order 54 and 42, respectively.
14.5 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\$

-define:F\_\_finite_field\_\_q.2\_\_end

-define:genma\_\_vector\_\_field\_\_format.2

-define:dense:$(CODE_N3_K2_Q2_GENMA)

-end

-define:P\_\_projective_space.1\_\_F\_\_end

-with:P\_\_do

-projective_space_activity

-cannotical_form_of_code

"3_2".genma\_\_save_ago\_\_label."3_2"

-classification_prefix."3_2"

-end

-end

pdflatex.3_2_classification.tex

open.3_2_classification.pdf

$\$(ORBITER\_PATH)orbiter.out\_\_v.2\_\_-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
```

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>#</th>
<th>Ago</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

375
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}
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:
g₁ = (1, 2) of order 2 and with 4 fixed points.
g₂ = (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:
g₁ = \[
\begin{bmatrix}
1 & 0 \\
1 & 1 \\
\end{bmatrix}
\] = \[
\begin{bmatrix}
10 \\
11 \\
\end{bmatrix}
\] of order 2 and with 1 fixed points.
g₂ = \[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\end{bmatrix}
\] = \[
\begin{bmatrix}
01 \\
10 \\
\end{bmatrix}
\] 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.

```plaintext
Hamming_5_2_with_5_2_3_code:
  > $(ORBITER_PATH)orbiter.out --v.2\$
  >   -define G:graph:Hamming_5_2:"
  >   > -subset:code_5_2_3:code_5_2_3"
  >   > $(CODE_5_2_3_CODEWORDS):-end:\n  >   > -with G:-do:\n  >   > -graph_theoretic_activity:-export_csv:-end:\n  >   > -with G:-do:\n  >   > -graph_theoretic_activity:-export_graphviz:-end:\n  >   > -with G:-do:\n  >   > -graph_theoretic_activity:-save:-end:\n  >   > -with G:-do:\n```

377
The group has order 32. For the graph theoretic commands, see Section 12.1.

The command

\begin{verbatim}
CODE_RM_3_1_GENMA="\
1111111\n0101010\n0011001\n0000111"

RM_3_1_group:
\end{verbatim}

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:

\begin{verbatim}
REED_MULLER_3_1_CODEWORDS="0,255,170,85,204,51,102,\
153,240,15,90,165,60,195,150,105"

RM_3_1_group_and_diagram:
\end{verbatim}
The drawing in Figure 14.4 is created.

The command

```
RS_6_4_7_group:
  $(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)$
  $end$
```
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

```bash
GV_n15_k6_d5_group:
  ➤ $(ORBITER_PATH) orbiter.out -v.20 \n  ➤   -define F -finite_field -q.2 -end \n  ➤   -define genma -vector -field F -format 6 \n  ➤   ➤ -compact $(CODE GV N15 K6) \n  ➤   -end \n  ➤   -define P -projective_space 5 F -end \n  ➤   -with P -do \n  ➤   -projective_space_activity \n  ➤   ➤ -canonical_form_of_code \n  ➤   ➤   "GV_n15_k6_d5".genma -save ago -label "GV_n15_k6_d5" \n  ➤   ➤   -classification_prefix "GV_n15_k6_d5" -end \n  ➤   ➤ -end \n  pdflatex GV_n15_k6_d5_classification.tex \n  open GV_n15_k6_d5_classification.pdf
```

computes the automorphism group of the Gilbert-Varshamov code from Section 9.8. It has order 12.
14.6  Canonical Forms of General Codes

The command

```plaintext
HAMMING_CODE_CODEWORDS="0,·67,·37,·102,·22,·85,·51,·112,·15,·76,·42,·105,·25,·90,·60,·127"

Hamming_graph_7_with_Hamming_code:
▷ $(ORBITER_PATH)orbiter.out:-v.2:\n▷  ▷ -define=G:-graph.=Hamming:7:2:\n▷  ▷  ▷ -subset="Hamming_code":"\|with\|Hamming\|code".\n▷  ▷  ▷ $(HAMMING_CODE_CODEWORDS)-end:\n▷  ▷  ▷ -with=G:-do:\n▷  ▷  ▷ -graph_theoretic_activity:-export_csv.-end:\n▷  ▷  ▷ -with=G:-do:\n▷  ▷  ▷ -graph_theoretic_activity:-export_graphviz.-end:\n▷  ▷  ▷ -with=G:-do:\n▷  ▷  ▷ -graph_theoretic_activity:-save.-end:\n▷  ▷  ▷ -with=G:-do:\n▷  ▷  ▷ -graph_theoretic_activity:-automorphism_group.-end
▷ pdflatex.Hamming_7_2_Hamming_code_report.tex
▷ open-Hamming_7_2_Hamming_code_report.pdf
```

computes the set stabilizer of the Hamming code inside the automorphism group of the Hamming graph. The group has order $2688 = 16 \cdot 168$. 

382
14.7 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:

```
Cycle_13_aut:
  $(ORBITER_PATH)orbiter.out\-v\-2\-define\-Gamma\-graph\-cycle\-13\-end\\
  \-with\-Gamma\-do\\
  \-graph\-theoretic\-activity\-automorphism\-group\-end\\
```

The output is two files: The first one, `Cycle_13_group.makefile` is a makefile containing an Orbiter command to create the automorphism group:

```
Cycle_13:
  $(ORBITER_PATH)orbiter.out\-v\-2\\
  \-define\-gens\-vector\-file\-Cycle\-13\-gens.csv\-end\\
  \-define\-G\-permutation\-group\\
  \-bsgs\-Cycle\-13\-"Cycle\-13"\-13\-26\-0\-5\-2\-gens\-end\\
```

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).

```
Chain_232_aut:
  $(ORBITER_PATH)orbiter.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
  pdflatex\-chain\-graph\-report.tex
  open\-chain\-graph\-report.pdf
```
The following report is written:

The automorphism group of chain_graph has order 1152 and is generated by:
Strong generators for a group of order 1152:

\[(12, 13),\]
\[(3, 4),\]
\[(2, 3),\]
\[(10, 11),\]
\[(9, 10),\]
\[(5, 6),\]
\[(7, 8),\]
\[(0, 1),\]
\[(0, 12)(1, 13)(2, 9)(3, 10)(4, 11)(5, 7)(6, 8)\]

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 -load_dimacs command. For instance, the following command computes the automorphism group of the Levi graph of the desarguesian projective plane:

```
JK_graph_pp16_1:
  $(ORBITER_PATH)orbiter.out -v 2
  -define:Gamma -graph -load_dimacs
  ../JUNTITILA/KASKI/benchmarks/pp/pp16-1
  -end
  -with:Gamma -do
  -graph_theoretic_activity -save -end
  -with:Gamma -do
  -graph_theoretic_activity -automorphism_group -end
```
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 \texttt{firstpathnode}, \( \text{nb_backtrack2} \) is the number of calls to \texttt{othernode}, \( \text{nb_backtrack3} \) is the number of calls to \texttt{processnode}, \( \text{nb_backtrack4} \) is the number of calls to \texttt{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:

```
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\ 
```

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.

```
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
```
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: \\
▷ ▷ ▷ -load_csv_no_border: \\
▷ ▷ ▷ halljanko315.csv: \\
▷ ▷ -end: \\
▷ ▷ -with:G:-do: \\
▷ ▷ ▷ -graph_theoretic_activity: -automorphism_group: \\
▷ ▷ -end: \\
▷ ▷ -with:G:-do: \\
▷ ▷ ▷ -graph_theoretic_activity: -properties: \\
▷ ▷ -end
```

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: \\
▷ ▷ ▷ halljanko315 gens.csv: -end: \\
▷ ▷ -define:G:-permutation_group: \\
▷ ▷ ▷ -bsgs:halljanko315: "File\halljanko315" : \\
▷ ▷ ▷ 315 \cdot 1209600 :: 0,1,2 :: 6 : gens: \\
▷ ▷ -end: \\
▷ ▷ -with:G:-do: \\
▷ ▷ ▷ -group_theoretic_activity: \\
▷ ▷ ▷ ▷ -poset_classification_control: \\
▷ ▷ ▷ ▷ -W: \\
▷ ▷ ▷ ▷ -problem_label:HJ_orbits: \\
▷ ▷ ▷ ▷ -depth:2: \\
▷ ▷ ▷ ▷ -end: \\
▷ ▷ ▷ -orbits_on_subsets:2: \\
```

386
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:
  > $(ORBITER_PATH)orbiter.out -v.2 \
  > -define gens -vector -file \
  >   halljanko315.gens -end \
  > -define G -permutation_group \
  > -bsgs halljanko315 "File\halljanko315" \
  > -define Gamma -graph \
  > -orbital_graph G 3 \
  > -end \
  > -with Gamma -do \
  >   graph_theoretic_activity \
  > -properties \
  > -end \
  > -with Gamma -do \
  >   graph_theoretic_activity \
  > -save \
  > -end
```

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.
  > $(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
```

The group is \(\text{PGO}(5,2)\) 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 15

Interfaces

15.1 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 15.2.

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 15.1. 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>-input_csv_file</td>
<td>csv-file</td>
<td>Specify the input csv file</td>
</tr>
<tr>
<td>-partition</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>-box_width</td>
<td>$w$</td>
<td>Use $w$ pixels per matrix entry.</td>
</tr>
<tr>
<td>-bit_depth</td>
<td>$d$</td>
<td>Use color bit depth of $d$ bits ($d = 8$ or $d = 24$).</td>
</tr>
<tr>
<td>-invert_colors</td>
<td></td>
<td>Use an inverted color scheme.</td>
</tr>
</tbody>
</table>

Table 15.1: Commands to Create Bitmap Graphics
F_7_tables:
  - $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot3\$
  - $-\text{define} F\cdot-\text{finite\_field}\cdot-q\cdot7\cdot-\text{end}\$
  - $-\text{with} F\cdot-\text{do}\cdot-\text{finite\_field\_activity}\$
  - $-\text{-\_cheat\_sheet\_GF}\$
  - $-\text{end}$
- $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot2\$
  - $-\text{draw\_matrix}\$
  - $-\text{-\_input\_csv\_file\_GF\_q7\_addition\_table.csv}\$
  - $-\text{-\_box\_width}40\$
  - $-\text{-\_bit\_depth}24\$
  - $-\text{-\_partition}3\cdot7\cdot7\$
  - $-\text{end}$
- open GF_q7_addition_table_draw.bmp

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 15.1. 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_cyclecic_inca:
  - $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot2\$
  - $-\text{-\_define} F\cdot-\text{-\_finite\_field}\cdot-q\cdot4\cdot-\text{-\_end}\$
  - $-\text{-\_define} P\cdot-\text{-\_projective\_space}\cdot2\cdotF\cdot-\text{-\_end}\$
  - $-\text{-\_with} P\cdot-\text{-\_do}\cdot-\text{-\_projective\_space\_activity}\$
  - $-\text{-\_cheat\_sheet\_for\_decomposition\_by\_element\_PG}\$
```

390
Figure 15.1: 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 15.2. 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 \(-\text{draw} \_\text{poset}\) option in the poset classification control command package, see Table 6.2. The posets are stored in a file with extension \( .\text{layered} \_\text{graph} \). These files can be drawn using the \(-\text{draw} \_\text{layered} \_\text{graph}\) command. The commands in Table 15.2 and Table 15.3 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 15.2: Drawing Options for Layered Graph Files (Part 1)
Figure 15.2: 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 15.3: Drawing Options for Layered Graph Files (Part 2)
Figure 15.3: The first basic orbit of $\text{PGL}(4,2)$ as a subgroup of $\text{PGO}^+(6,2)$

PGL\_4\_2\_Wedge\_4\_0\_detached\_graphical\_output:
\[ $(\text{ORBITER} \_\text{PATH})\text{orbiter.out}; -v; 12; \]
\[ -\text{define} \cdot G; -\text{linear} \cdot \text{group}; -\text{PGL} \cdot 4 \cdot 2; \]
\[ -\text{wedge} \cdot \text{detached}; -\text{end}; \]
\[ -\text{with} \cdot G; -\text{do}; \]
\[ -\text{group} \cdot \text{theoretic} \cdot \text{activity}; \]
\[ -\text{report}; \]
\[ -\text{end} \]
\[ \text{pdflatex}; \text{PGL} \cdot 4 \cdot 2 \cdot \text{Wedge} \cdot 4 \cdot 0 \cdot \text{detached} \cdot \text{report} \cdot \text{tex}; \]
\[ \text{open}; \text{PGL} \cdot 4 \cdot 2 \cdot \text{Wedge} \cdot 4 \cdot 0 \cdot \text{detached} \cdot \text{report} \cdot \text{pdf}; \]

produces a report about this group action. Figure 15.3 shows the first basic orbit in the stabilizer chain of the group in that action.

The command

schreier\_tree\_graphical\_output:
\[ $(\text{ORBITER} \_\text{PATH})\text{orbiter.out}; -v; 4; \]
\[ -\text{draw} \cdot \text{options}; \]
\[ -\text{yout}; 500000; \]
\[ -\text{radius}; 15; -\text{nodes} \cdot \text{empty}; \]
\[ -\text{line} \cdot \text{width}; 0.5; -\text{y} \cdot \text{stretch}; 0.25; \]
\[ -\text{end}; \]
\[ -\text{define} \cdot G; -\text{linear} \cdot \text{group}; -\text{PGL} \cdot 4 \cdot 2; -\text{end}; \]
\[ -\text{with} \cdot G; -\text{do}; \]
\[ -\text{group} \cdot \text{theoretic} \cdot \text{activity}; \]
\[ -\text{orbits} \cdot \text{on} \cdot \text{polynomials}; 3; \]
Figure 15.4: A Schreier tree in the action on polynomials

draws the 6th Schreier tree in the classification of orbits of PGL(4,2) on homogeneous polynomials of degree 3 in 4 variables. The drawing is shown in Figure 15.4. This particular orbit has length 420, so there are 420 nodes in the tree.
15.2 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 15.3). 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 15.4-15.5 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 15.6 and 15.7 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 15.8 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 15.7 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:
  $\$(ORBITER\_PATH)orbiter.out\,-v\,2\,-povray\,$
  \(-\text{round}\,0\,-\text{nb\_frames}\,\text{\_default}\,30\,$
  \(-\text{output}\_\text{mask}\,\%d\,\%03d\,\text{.pov}\,$
  \(-\text{video}\_\text{options}\,-W\,1024\,-H\,768\,$
  \(-\text{global}\_\text{picture}\_\text{scale}\,0.5\,$
  \(-\text{default}\_\text{angle}\,75\,$
  \(-\text{clipping}\_\text{radius}\,2.7\,$
  \(-\text{end}\,$
  \(-\text{scene}\_\text{objects}\,$
  \(-\text{obj}\_\text{file}\,\text{cube\_centered}\,\text{.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\,$
<table>
<thead>
<tr>
<th>Option</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-do_not_rotate</td>
<td></td>
<td>No rotation. By default, the animation consists of a full rotation around a vertical axis.</td>
</tr>
<tr>
<td>-rotate_about_z_axis</td>
<td></td>
<td>Rotate around z-axis.</td>
</tr>
<tr>
<td>-rotate_about_111</td>
<td></td>
<td>Rotate around (1,1,1)-axis (default).</td>
</tr>
<tr>
<td>-rotate_about_custom_axis</td>
<td>axis</td>
<td>Rotate around a custom axis. The axis is specified as a vector of length 3.</td>
</tr>
<tr>
<td>-boundary_none</td>
<td></td>
<td>Remove the clipping.</td>
</tr>
<tr>
<td>-boundary_box</td>
<td></td>
<td>Clip using a box shape.</td>
</tr>
<tr>
<td>-boundary_sphere</td>
<td></td>
<td>Clip using a sphere (default).</td>
</tr>
<tr>
<td>-font_size</td>
<td>s</td>
<td>Set font size to s.</td>
</tr>
<tr>
<td>-stroke_width</td>
<td>s</td>
<td>Set text depth to s.</td>
</tr>
<tr>
<td>-omit_bottom_plane</td>
<td></td>
<td>Remove the bottom plane.</td>
</tr>
<tr>
<td>-W</td>
<td>w</td>
<td>Set output dimension to w pixels wide.</td>
</tr>
<tr>
<td>-H</td>
<td>h</td>
<td>Set output dimension to h pixels height.</td>
</tr>
<tr>
<td>-nb_frames</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>-zoom</td>
<td>r $a_s$ $a_t$ $c_s$ $c_t$</td>
<td>Set zoom angle and clipping with in round r to change from $a_s$ to $a_t$ and from $c_s$ to $c_t$, respectively.</td>
</tr>
<tr>
<td>-pan</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>-pan_reverse</td>
<td>$r$ $F$ $T$ $C$</td>
<td>Same as -pan, but camera movement is in opposite order.</td>
</tr>
<tr>
<td>-no_background</td>
<td></td>
<td>Remove background.</td>
</tr>
<tr>
<td>-no_bottom_plane</td>
<td>$r$</td>
<td>Remove bottom plane in round $r$.</td>
</tr>
<tr>
<td>-camera</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 15.4: 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 praeamble g text l fname</td>
<td>In round r, produce running latex text text with start value s, sustain s and gravity g. Put praeamble 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 15.5: 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 $i_1$</td>
</tr>
<tr>
<td>-triangular_face_given_by_three_lines</td>
<td>$i_1 \ i_2 \ i_3$</td>
<td>Create a triangular face given 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 15.6: 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>(i_1) (i_2)</td>
<td>Create a line through two points</td>
</tr>
<tr>
<td>-line_through_point_with_direction</td>
<td>(x) (y) (z) (u_x) (u_y) (u_z)</td>
<td>Create a line through a point ((x, y, z)) with a given direction ((u_x, u_y, u_z))</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 15.7: 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 15.8: 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:

```
v -1 -1 -1
v 1 -1 -1
v -1 1 -1
v 1 1 -1
```

402
The monkey saddle is a cubic surface, given by the equation

\[ z = x^3 - 3xy^2 \]

The next example plots the surface known 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,-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.0.8 -
  -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" -
  -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
```
Here is one of the frames that are created:

![Eckardt surface animation frame](image)

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 Schl¨afli labeling of the lines is indicated.

**Eckardt:**

```
mv-makefile_animation.PO
```

```bash
mv $(ORBITER_PATH)orbiter.out -v2 -povray
mv -round0 -nb_frames_default.30
mv -output_mask-Eckardt_%d_03d.pov
mv -video_options-W1024-H768
mv -global_picture_scale0.9
mv -default_angle75
mv -clipping_radius2.4
mv -camera0."1,1,1"."-3,1,3"."0.12,0.12,0.12"
mv -end
mv -scene_objects
mv -Hilbert_Cohn_Vossen_surface
mv -group_of_things"0"
mv -cubics0."texture{pigment{White*0.5.transmit0.5}}"."finish{ambient0.4.diffuse0.5.roughness0.001.reflection0.1.specular.8}"."14,19,23"
mv -lines1.0.02."texture{pigment{color.Red}}"
```
Figure 15.5 shows the final product.
Figure 15.5: The Eckardt surface
The Endrass octic [25] is the algebraic surface given by the equation

\[ x_8 = 64 (-w^2 + x^2) (-w^2 + y^2) ((x+y)^2 - 2 w)^2 ((x-y)^2 - 2 w)^2 - \left(-4 \left(1 + \sqrt{2}\right) (x^2 + y^2)^2 + 8 \right) \]

\[ + 2 \left(2 + 7 \sqrt{2}\right) w^2) (x^2 + y^2) - 16 z^4 + 8 \left(1 - 2 \sqrt{2}\right) z^2 w^2 - \left(1 + 12 \sqrt{2}\right) w^4)^2 \]

The following Orbiter command creates a povray graphics of the octic, shown in Figure 15.6:

```pov
ENDRASS_OCTIC.LEX165="-93.2548,0,0,0,0,0,0,0,-309.019,0,0,527.529,0,395.647,\n0,0,0,0,0,0,0,0,-687.529,0,0,1582.59,0,1186.94,0,0,0,0,-1055.06,0,\n-1582.59,0,-593.47,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-309.019,\n0,0,1582.59,0,1186.94,0,0,0,0,-2110.12,0,-3165.17,0,-1186.94,0,0,0,0,0,\n0,874.039,0,1560.63,0,1677.92,0,343.362,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\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,-93.2548,0,0,527.529,0,395.647,\n0,0,0,0,-1055.06,0,-1582.59,0,-593.47,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\n1677.92,0,343.362,0,0,0,0,0,0,0,-256.0,0,-468.077,0,-789.019,0,\n-525.726,0,0,0.941125"

endrass8:
▷ $(\text{ORBITER\_PATH})\text{or}b\text{i}t\text{e.out}\cdot-v\cdot2\cdot-povray\cdot$
▷ ▷ -round\_0:nb\_frames\_default:30$
▷ ▷ -output_mask\_endrass\_octic\_d\_%03d.pov$
▷ ▷ -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-\_phong\_1\}\}"
▷ ▷ ▷ -lines:1:0.15\:"texture\{pigment\{\_color\_Green\}\}$
finish\{\_diffuse\_0.9-\_phong\_1\}\}"
▷ ▷ ▷ -lines:2:0.15\:"texture\{pigment\{\_color\_Blue\}\}$
finish\{\_diffuse\_0.9-\_phong\_1\}\}"
▷ ▷ ▷ -lines:3:0.05\:"texture\{pigment\{\_color\_Black\}\}$
finish\{\_diffuse\_0.9-\_phong\_1\}\}"
```

407
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 15.9: Prepare frames commands

15.3 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 15.9. For instance, the command

```sh
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 -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].

409
15.4 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 \( r \sin(at + c), \ r \sin(bt) \) 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:

\[
\begin{align*}
\sin(x) & \mapsto \text{push } x \ \text{sin}, \\
\add{a}{b} & \mapsto \text{push } a \ \text{push } b \ \text{add}, \\
\times{a}{b} & \mapsto \text{push } a \ \text{push } b \ \text{mult}.
\end{align*}
\]

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 \( a = 3, \ b = 2, \ c = \pi/2 \) and \( r = 7 \), the function is computed using

\texttt{lissajous:}
\begin{verbatim}
  \$\{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{-code_end-}
\end{verbatim}

The sequence

\[
\text{push } t \ \text{push } a \ \text{push } c \ \text{add} \ \text{sin} \ \text{push } r \ \text{mult}
\]

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.003d.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:0.09:"texture{pigment{color:Yellow}}" -
  ▶ ▶ -lines:1.09:"texture{pigment{color:Yellow}}" -
  ▶ ▶ -lines:2.09:"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:4.0.0."texture{pigment{color:Red}}" -
  finish:{diffuse:0.9:phong:1}" -
  ▶ ▶ -plane_by_dual_coordinates:"0,0,1,0" -
  ▶ ▶ -group_of_things:"0" -
  ▶ ▶ -planes:5."texture{pigment{color:Blue*0.5:transmit:0.5}}" -
transmit:0.5" -
  ▶ ▶ -scene_objects_end -
```

411
Figure 15.7: Lissajous figure

The plot is shown in Figure 15.7.

We can turn it into a 3D plot by using the \( t \) value for the \( z \) coordinate. The function is computed using the command

\[
lissajous_3d:\$
\begin{align*}
-\text{povray_end} \\
-\text{rm}-\text{rf-POV} \\
\text{mkdir-POV} \\
\text{mv} \cdot \text{lissajous}_0 \_\ast .\text{pov-POV} \\
\text{mv} \cdot \text{makefile} \_\text{animation-POV}
\end{align*}
\]

The code to produce the 3D plot is

\[
lissajous_3d \_\text{plot}: \$
\begin{align*}
-\$(\text{ORBITER_PATH}) \text{orbiter.out} -v.2 \backslash \\
-\text{smooth_curve}" \text{lissajous}_3d" \cdot 0.07 \cdot \text{2000} \cdot 0.0 \cdot 0.18.85 \backslash \\
-\text{const} \cdot a.3 \cdot b.2 \cdot c.1.57 \cdot r.7 \cdot -\text{const_end} \backslash \\
-\text{var} \cdot t \cdot -\text{var_end} \backslash \\
-\text{code} \backslash \\
-\text{push} \cdot t \cdot \text{push} \cdot a \cdot \text{mult} \cdot \text{push} \cdot c \cdot \text{add} \cdot \text{sin} \cdot \text{push} \cdot r \cdot \text{mult} \cdot \text{return} \backslash \\
-\text{push} \cdot t \cdot \text{push} \cdot b \cdot \text{mult} \cdot \text{sin} \cdot \text{push} \cdot r \cdot \text{mult} \cdot \text{return} \backslash \\
-\text{push} \cdot t \cdot \text{return} \backslash \\
-\text{code_end} \backslash \\
\end{align*}
\]

The code to produce the 3D plot is

\[
lissajous_3d \_\text{plot}: \$
\begin{align*}
-\$(\text{ORBITER_PATH}) \text{orbiter.out} -v.2 -\text{povray} \backslash \\
-\text{round} \cdot 0 -\text{nb_frames} \_\text{default} \cdot 30 \backslash \\
\end{align*}
\]
The 3D curve is shown in Figure 15.8.
Figure 15.8: Lissajous Spacecurve
Chapter 16

Miscellaneous

16.1 Miscellaneous

Table 16.1 list miscellaneous Orbiter commands. The command \texttt{-csv\_file\_select\_rows} can be used to select rows from a csv file. The command \texttt{-csv\_file\_select\_cols} can be used to select columns from a csv file. The command \texttt{-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 $\alpha^0, \alpha^2, \alpha^4$.

\begin{verbatim}
misc_select:
▷ $(\text{ORBITER\_PATH})$orbit.out-\text{-v.3-}
▷ ▷ -\text{define} F -\text{finite} field-q.7-\text{-end-}
▷ ▷ -\text{with} F -\text{do} -\text{finite} field \text{activity} -\text{cheat} sheet \text{GF} -\text{-end} 
▷ $($(\text{ORBITER\_PATH})$orbit.out-\text{-v.4-}$\text{-csv\_file\_select\_rows\_and\_cols-}$
▷ ▷ \text{GF}\_q7\_multiplication\_table\_reordered.csv-
▷ ▷ "0,2,4" "0,2,4".
\end{verbatim}

The even powers of $\alpha$ create a multiplicative subgroup. Figure 16.1 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

\begin{verbatim}
Row,C0,C1,C2,C3,C4,C5
0,1,3,2,6,4,5
1,3,2,6,4,5,1
2,2,6,4,5,1,3
3,6,4,5,1,3,2
4,4,5,1,3,2,6
\end{verbatim}

415
<table>
<thead>
<tr>
<th>Option</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-csv_file_select_rows</td>
<td>fname $R$</td>
<td>Selects rows listed in $R$ from the csv-file fname.</td>
</tr>
<tr>
<td>-csv_file_select_cols</td>
<td>fname $R$</td>
<td>Selects columns listed in $R$ from the csv-file fname.</td>
</tr>
<tr>
<td>-csv_file_select_rows_and_cols</td>
<td>fname $R$ $C$</td>
<td>Selects rows listed in $R$ and columns listed in $C$ from the csv-file fname.</td>
</tr>
<tr>
<td>-csv_file_join</td>
<td>fname col-label</td>
<td>Joins csv file fname according to column with label col-label. This option is given once for each file that should be joined.</td>
</tr>
<tr>
<td>-csv_file_latex</td>
<td>fname</td>
<td>Produces a latex table from the given csv-file.</td>
</tr>
<tr>
<td>-store_as_csv_file</td>
<td>fname $m$ $n$ $L$</td>
<td>Stores the data in $L$ to a csv file. The data is an $m \times n$ matrix in row-major ordering.</td>
</tr>
</tbody>
</table>

Table 16.1: Miscellaneous Orbiter Commands

Figure 16.1: 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
```
## 16.2 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 17

Orbiter on Windows

17.1 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.
4. Install additional software using your own GNU/Linux distribution package manager.
5. Invoke Windows applications using a Unix-like command-line shell.
6. 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 are 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](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

![Command execution](image)

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/

- Remember: all Ubuntu commands must begin with either
  
  - wsl or wsl.exe
Makefile

- To edit file in vim, click “!”
- 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 18

The Makefile

18.1 The Makefile

1 #MY_PATH=../orbiter
2 MY_PATH="~/DEV.22/orbiter"
3 #MY_PATH=/scratch/betten/COMPILE/orbiter
4
5 # uncomment exactly one of the following two lines.
# uncomment the first if you want to run orbiter through docker.
6 # uncomment the second if you have an installed copy of orbiter and you want to run it directly.
7 #ORBITER_PATH=docker run -it --volume ${PWD}:/mnt --workdir /mnt
8 ORBITER_PATH=$(MY_PATH)/src/apps/orbiter/
9
10 # additional configurations for when you want to compile automatically generated code
11
12 SRC=$(MY_PATH)/src
13 MY_CPP=g++
14 MY_CC=gcc
15 CPPFLAGS=-Wall -I. -I../../DEV.22/orbiter/src/lib -std=c++14
16 LIB=$(SRC)/lib/liborbiter.a -lpthread
17 LFLAGS=-lm -Wl,-rpath=/usr/local/gcc-8.2.0/lib64
18
19 # End of configuration part
20
21 # additional configurations for when you want to compile automatically generated code
22
23 # End of configuration part
```
31 GINAC_PATH=$(/path/to/ginac)
32 SANDBOX_PATH=$(/path/to/sandbox)
33
34 update:
35  cd $(ORBITER_PATH); make clean;
36  cd $(MY_PATH); make clean; git pull; make
37
38 update_all:
39  cd $(MY_PATH); make clean; git pull; make
40
41 sandbox:
42  $(SANDBOX_PATH)/sandbox.out
43

# Makefile Variables

50 MAGMA_PATH=/usr/local/magma

V7_VANDERMONDE_EXTENDED="\n1,0,0,0,0,0,1,0,0,0,0,0,0,0,\n1,1,1,1,1,1,0,1,0,0,0,0,0,0,\n1,2,4,1,2,4,1,0,0,1,0,0,0,0,\n1,3,2,6,4,5,1,0,0,0,1,0,0,0,\n1,4,2,1,4,2,1,0,0,0,0,1,0,0,\n1,5,4,6,2,3,1,0,0,0,0,0,1,0,\n1,6,1,6,1,6,1,0,0,0,0,0,0,0,1"

# Co3 from Conway et al., 1985 (ATLAS)
# order=495766656000
# Co3 from the paper by Suleiman and Wilson 1997

CONWAY_GEN1="\n1101110010000001010000\n1111010111101000001011\n000001000000010001010\n1111001101100100010110\n010101000000010011010\n0000010000000010010101\n0100000000000100010101"
```
CONWAY

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

ENDRASS_SPARSE="
6,0,4,2,7,5,9,6,20,6,23,1,25,3,30,1,32,3,34,4,56,6,59,1,61,6,66,
2,68,6,70,3,77,2,79,6,83,6,120,2,123,5,125,3,130,1,132,3,134,3,141,
2,143,6,147,3,156"
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,\n0,0,1,0,0,0,0,0,\n1,0,0,0,0,0,0,0,\n0,0,1,0,0,0,0,0,\n0,1,0,1,1,0,0,0,\n0,0,0,0,1,1,0,0,\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,\n0,0,0,0,0,1,0,0,\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"

Ree_gen2="16,3,11,5,16,22,20,24,6,18,24,7,1,26,9,23,17,18,23,20,13,\"
169 9,7,2,15,17,5,11,3,3,6,21,4,24,16,25,8,6,24,21,12,7,\
170 24,15,2,13,11,14,24"
171
172
173 HIRSCHFELD_SURFACE_Q4_SET_0F_POINTS="0,1,2,3,4,5,6,7,8,9,\
174 10,11,12,13,14,23,26,27,30,31,34,35,38,39,42,47,48,51,52,\
175 53,54,59,60,61,62,67,68,69,70,75,76,79,80,81,82"
176
177 HYPEROVAL_16_144="0,1,2,3,52,67,89,106,126,\
178 141,159,176,184,199,220,235,245,262"
179 HYPEROVAL_16_16320="0,1,2,3,52,70,83,109,127,\
180 139,156,174,186,199,217,229,256,264"
181
ELEMENTARY SYMMETRIC_4_4="x0*x1*x2*x3"

CODE_5_2_3_CODEWORDS="0,7,25,30"

SURFACE_F7_15LINES_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="1,5,-1,16,-2,17,1,10,2,18,1,8,1,9,-1,11,-1,19,-1,14"

CODE_RS_6_4_7="\\
621000-\\
062100-\\
006210-\\
000621"

CODE_RS_10_8_11="\\
7,2,1,0,0,0,0,0,0,0,\\
0,7,2,1,0,0,0,0,0,0,\\
0,0,7,2,1,0,0,0,0,0,\\
0,0,0,7,2,1,0,0,0,0,\\
0,0,0,0,7,2,1,0,0,0,\\
0,0,0,0,0,7,2,1,0,0,\\
0,0,0,0,0,0,7,2,1,0,
0,0,0,0,0,0,0,7,2,1-
#

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"
D4="1,3,1,4,1,11,1,9,1,10,1,14,1,17,1,18,1,19"
D5="1,3,1,4,1,8,1,10,1,17,1,18,1,19"
D6="1,2,1,3,1,5,1,7,1,10,1,11,1,19"
D7="1,2,1,3,1,5,1,7,1,10,1,11,1,19"
D8="1,3,1,4,1,5,1,7,1,10,1,11,1,19"
D9="1,3,1,4,1,8,1,10,1,14,1,15"
D10="1,4,1,7,1,8,1,9,1,10,1,13,1,15,1,16,1,17"
D11="1,4,1,7,1,8,1,10,1,13,1,14,1,15,1,16,1,17"
D12="1,4,1,6,1,7,1,8,1,9,1,10,1,15,1,16,1,17"
D13="1,6,1,8,1,9,1,10,1,15,1,16,1,17"
D14="1,6,1,7,1,10,1,11,1,14,1,15"
D15="1,4,1,5,1,6,1,8,1,12,1,14,1,16"
D16="1,4,1,7,1,15,1,12"
D17="1,4,1,7,1,8,1,9,1,11,1,13,1,15,1,16,1,17,1,18,1,19"
D18="1,4,1,7,1,8,1,11,1,13,1,14,1,15,1,16,1,17,1,18,1,19"
D19="1,5,1,6,1,7,1,8,1,9,1,10,1,15,1,16,1,17"
D20="1,4,1,5,1,6,1,7,1,8,1,10,1,13,1,14,1,15,1,16,1,17"
D21="1,5,1,6,1,8,1,10,1,13,1,14,1,15,1,16,1,17"
D22="1,4,1,5,1,7,1,9,1,10,1,11,1,13,1,14,1,15,1,18,1,19"
D23="1,4,1,7,1,9,1,11,1,13,1,14,1,15,1,18,1,19"
D24="1,5,1,8,1,9,1,10,1,13,1,14,1,15,1,18,1,19"
D25="1,5,1,10,1,13,1,11,1,9,1,14,1,15,1,18,1,19"
D26="1,5,1,6,1,8,1,10,1,15,1,18,1,19"
D27="1,4,1,5,1,7,1,10,1,11,1,12,1,13,1,14,1,17,1,18"
D28="1,4,1,5,1,6,1,7,1,10,1,11,1,14,1,15,1,17,1,18"
D29="1,4,1,7,1,13,1,11,1,9,1,15,1,17,1,18"
D30="0,1,1,1,2,1,3,1,7,1,10,1,11,1,16"
D31="1,0,1,1,1,2,1,5,1,7,1,11,1,13,1,14,1,15,1,17,1,18,1,19"
D32="1,0,1,1,2,1,3,1,7,1,8,1,10,1,11,1,12,1,15,1,16,1,19"
D33="1,0,1,1,2,1,3,1,6,1,7,1,10,1,12,1,13,1,15,1,16,1,19"
D34="1,0,1,1,2,1,3,1,7,1,10,1,16,1,19"
D35="1,0,1,1,2,1,3,1,6,1,7,1,8,1,11,1,13,1,15"
D36="1,0,1,1,2,1,3,1,7,1,15"
D37="1,0,1,1,2,1,3,1,7,1,12,1,15,1,19"
D38="1,0,1,1,2,1,3,1,6,1,7,1,12,1,13,1,15,1,19"
D39="1,0,1,1,2,1,3,1,7,1,5,1,10,1,6,1,13,1,8,1,11,1,9,1,14,1,12,1,15,1,19"
D40="1,0,1,1,2,1,3,1,7,1,5,1,10,1,6,1,13,1,19"
D41="1,2,1,4,1,7,1,5,1,10,1,11,1,14,1,19"
D42="1,2,1,4,1,7,1,11,1,12,1,14,1,15,1,19"
D43="1,0,1,1,2,1,7,1,13,1,8,1,11,1,14,1,15,1,17"
D44="1,0,1,1,2,1,6,1,8,1,11,1,12,1,14"
D45="1,6,1,8,1,9,1,11,1,12,1,13,1,14,1,15,1,16,1,17,1,18,1,19"
D46="1,5,1,10,1,6,1,13,1,8,1,11,1,9,1,14,1,12,1,15,1,16,1,17,1,18,1,19"
D47="1,5,1,10,1,6,1,13,1,8,1,11,1,9,1,14,1,16,1,17,1,18,1,19"

# Normal form for 15 lines:
F_ALPHA_BETA_GAMMA_DELTA="beta*(gamma+1)*x0*x0*x2\n+.(alpha*delta-beta*gamma+alpha-beta-delta-1)*x0*x1*x2\n-1*(alpha*beta-alpha*delta+delta)*(gamma+1)*x0*x1*x3\n+.(-alpha*delta+alpha*gamma-beta-gamma-delta-gamma)*x0*x2*x2\n-(alpha*delta+beta-delta)*(gamma+1)*x0*x2*x3\n-(delta+1)*(alpha-1)*x1*x1*x2\n-(delta+1)*(alpha-1)*x1*x1*x3\n+.(alpha*delta-alpha*gamma+beta*gamma+beta-delta+gamma)*x1*x2*x2\n+.(alpha*beta*gamma+alpha-beta+alpha*delta)\n-alpha*gamma+beta*gamma-alpha-beta-delta+gamma)*x1*x2*x3\n+.alpha*beta*(gamma+1)*x1*x3*x3"

#general-normal-form-for-surfaces-with-27-lines:

F_abcd_eqn="-(a*b*c-d-a*b+d-a*c+d+a*d-b*c)*(b-d)*X0*X0*X2\n+(a+b+c+a*b+d-a*c+d+a*d-b*c)*(a+b-c-d)*X0*X1*X2\n+.a*c+d-a*b+c*c-a*a+a*b+d+b*c+c-b*c*d)*(b-d)*X0*X2*X3\n-(a+c)+(a*b+c-a*b+d-a*c+d+b*c+d+a*d-b*c)*X1*X1*X2\n-.a*c+d-a*b+c*c-a*a+a*b+d+b*c+c-b*c*d)*(b-d)*X1*X2*X3\n+.a*c+d-a*b+c*c-a*a+a*b+d+b*c+d+a*d-b*c)*X1*X2*X2\n+.((1+1)*a*a*b*c+d-a*a*b*d-d-(1+1)*a*a*c*d\n-(1+1)*a*b*c*c+a*b*c*c+(1+1)*a*b*c*c+a*b*c*d\n-b*b*c*c*+a*a*b*c+a*c+d+a*a*d+d+a*b*b*c+a*b*c*\n-(1+1+1)*a*b*c*d-a*c*c*d+a*c*d+d+b*b*c*c)*X1*X2*X3\n+.c*a*(a*d-b*c-a+b+c-.d)*(b-d)*X1*X3*X3"

#proposed-normal-form-for-smooth-cubic-surfaces-with-9-lines:

F_a_b_c_d_f_g="g*b*(c*f+d*f-d*g-c+f-g)*x0*x0*x2\n+.a*d*g*g-b*c*f*f-b*d*f+f+b*d*g*g.\n-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\n+(1+c+d)*f*(a*g-.b*f+b*g-a)*x0*x1*x3\n+.g*(c*f+d*f-d*g-c+f-g)*a*x0*x2*x2\n-.b*a*g*(c*f+d*f-d*g-c+f-g)*x0*x2*x3\n-(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\n-.c*f*(a*g-.b*f+b*g-a)*x1*x2*x2\n+.d-1)*c*f*(a*g-.b*f+b*g-a)*x1*x2*x3\n+.c*d*f*(a*g-.b*f+b*g-a)*x1*x3*x3"

KNECHT_13_1_AS_PAIRS="1,0,1,1,1,2,12,9"
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="1,1,0,0,0,0,0,0,8,0,8,0,0,0"

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

#[24,12,8]
0,1,2,3,4,5,6,7,8,9,10,11,132,913,1460,1750,1898,2518,2787,2874,3320,3357,3662,4004

CODE_RM_3_1 GENMA="
11111111"
01010101
00110011
00001111"

CODE_RM_4_1 GENMA="
1111111111111111"
0101010101010101
0011001100110011
0000111100001111
0000000011111111"

CODE_RS_8="
5610000"
0561000
0056100
0005610
0000561"

CODE_RS_11 RREF="
1,0,0,0,0,0,0,7,2",
0,1,0,0,0,0,0,8,3",
0,0,1,0,0,0,0,1,2",
0,0,0,1,0,0,0,8,8",
0,0,0,0,1,0,0,10,3",
0,0,0,0,0,1,0,0,9,1"
RS_8_reduced="\n0100011000000000000000\n0011100100000000000000\n1100110010000000000000\n0000100011000000000000\n0000011100100000000000\n0001100110010000000000\n0000001000110000000000\n0000000111001000000000\n0000001100110000000000\n0000000100011000000000\n0000000011100100000000\n0000000011100110000000\n0000000011001000000000\n0000000001100000000000\n0000000001100010000000\n0000000001100100000000\n0000000001100000000000\n0000000001100000000000
"

CODE_21_15_4="\n1100001000000000000000\n1101000100000000000000\n1011000010000000000000\n0110000010000000000000\n1100100000000000000000\n1010010000000000000000\n0110010000000000000000\n1001100000000000000000\n0101100000000000000000\n0011100000000000000000\n1111100000000000000000\n1100010000000000000000\n1010010000000000000000\n0110010000000000000000\n1001010000000000000000"
#ago=12
CODE_15.6.6_B="\n111111111100000-
111110000010000-
110011000010000-
101010100001000-
010110010000010-
011011001000001"

#ago=720:
CODE_15.6.6_C="\n111111111100000-
111110000010000-
110011000010000-
110101010000100-
101101001000010-
100010111000001"

#ago=96:
CODE_15.6.6_D="\n111111111100000-
111110000010000-
110011000010000-
111001010000100-
110101001000010-
111010011000001"

#ago=360
CODE_15.6.6_E="\n111111111100000-
111110000010000-
110011000010000-
110101010000100-
110110010000100-
01101011000001"

BCH_21.15.PROJ="-0, -1, -19, -37, -113, -420, -1651, -6577, -26284, -105115, -420442, -168175, -6727000, -26907991, -107631958, -27874647, -111498582, -43341143, -173364566, -156587350, -14"
476
477

478
479

480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496

497

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
*x7·+·x1*x6*x7·+·x2*x3*x4·+·x2*x3*x5·+·x2*x3*x6·+·x2*x3*x7·+·x2*x4*x5·+·x2*x4*x6·

440


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\"
Orb35=\"65554\"
<table>
<thead>
<tr>
<th>Orb</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orb83</td>
<td>65921</td>
</tr>
<tr>
<td>Orb84</td>
<td>65922</td>
</tr>
<tr>
<td>Orb85</td>
<td>66755</td>
</tr>
<tr>
<td>Orb86</td>
<td>66763</td>
</tr>
<tr>
<td>Orb87</td>
<td>66764</td>
</tr>
<tr>
<td>Orb88</td>
<td>67107</td>
</tr>
<tr>
<td>Orb89</td>
<td>67115</td>
</tr>
<tr>
<td>Orb90</td>
<td>67117</td>
</tr>
<tr>
<td>Orb91</td>
<td>67150</td>
</tr>
<tr>
<td>Orb92</td>
<td>67243</td>
</tr>
<tr>
<td>Orb93</td>
<td>73731</td>
</tr>
<tr>
<td>Orb94</td>
<td>73733</td>
</tr>
<tr>
<td>Orb95</td>
<td>73737</td>
</tr>
<tr>
<td>Orb96</td>
<td>73753</td>
</tr>
<tr>
<td>Orb97</td>
<td>73795</td>
</tr>
<tr>
<td>Orb98</td>
<td>73796</td>
</tr>
<tr>
<td>Orb99</td>
<td>73797</td>
</tr>
<tr>
<td>Orb100</td>
<td>73798</td>
</tr>
<tr>
<td>Orb101</td>
<td>73801</td>
</tr>
<tr>
<td>Orb102</td>
<td>73802</td>
</tr>
<tr>
<td>Orb103</td>
<td>73987</td>
</tr>
<tr>
<td>Orb104</td>
<td>73993</td>
</tr>
<tr>
<td>Orb105</td>
<td>74007</td>
</tr>
<tr>
<td>Orb106</td>
<td>74051</td>
</tr>
<tr>
<td>Orb107</td>
<td>74052</td>
</tr>
<tr>
<td>Orb108</td>
<td>74053</td>
</tr>
<tr>
<td>Orb109</td>
<td>74055</td>
</tr>
<tr>
<td>Orb110</td>
<td>74057</td>
</tr>
<tr>
<td>Orb111</td>
<td>74099</td>
</tr>
<tr>
<td>Orb112</td>
<td>74105</td>
</tr>
<tr>
<td>Orb113</td>
<td>74106</td>
</tr>
<tr>
<td>Orb114</td>
<td>74243</td>
</tr>
<tr>
<td>Orb115</td>
<td>74247</td>
</tr>
<tr>
<td>Orb116</td>
<td>74248</td>
</tr>
<tr>
<td>Orb117</td>
<td>74263</td>
</tr>
<tr>
<td>Orb118</td>
<td>74264</td>
</tr>
<tr>
<td>Orb119</td>
<td>74275</td>
</tr>
<tr>
<td>Orb120</td>
<td>74276</td>
</tr>
<tr>
<td>Orb121</td>
<td>74279</td>
</tr>
<tr>
<td>Orb122</td>
<td>74280</td>
</tr>
<tr>
<td>Orb123</td>
<td>74295</td>
</tr>
<tr>
<td>Orb124</td>
<td>74296</td>
</tr>
<tr>
<td>Orb125</td>
<td>74499</td>
</tr>
<tr>
<td>Orb126</td>
<td>74500</td>
</tr>
<tr>
<td>Orb127</td>
<td>74531</td>
</tr>
<tr>
<td>Orb128</td>
<td>74532</td>
</tr>
<tr>
<td>Orb129</td>
<td>76051</td>
</tr>
</tbody>
</table>
# elements of order 2:
# conjugacy class reps:
# elt order, class size, centralizer order:

#2A: 2.48960.40320.Baer.involution

#2B: 2.5355.368640.one block of 10,11

#2C: 2.64260.30720.two blocks of 10,11.(problem group)

#> pdflatex PGGL_4_4_classes_out.tex
#> open PGGL_4_4_classes_out.pdf

# elements of order 2:
# conjugacy class reps:
# elt order, class size, centralizer order:

#2A: 2.48960.40320.Baer.involution

#2B: 2.5355.368640.one block of 10,11

#2C: 2.64260.30720.two blocks of 10,11.(problem group)

CLASS_2A=-centralizer_of_element:"1,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1,1".--label."2A".

# Baer.involution
CLASS 2B=centralizer_of_element:"1,0,0,0,-1,1,0,0,0,0,1,0,-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"--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_S

> -subgroup_by_generators:"2A"::1:"1,0,0,0,-0,1,0,0,-0,0,1,0,-0,0,1,1"
PGGL 4 4_SUBGROUP 2A_NORMALIZER=PGGL 4 4_S

> -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,
1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,1,1,
1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,0,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,0,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,0,1,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,1,1,1,

0,1,0,0,0,1,0,1,1,1,1,0,1,0,1,1,1,1

#the-problem-group, two-blocks_of 10,11:

PGGL 4 4_SUBGROUP 2C=PGGL 4 4_S

> -subgroup_by_generators:"2C"::2:1:
PGGL 4 4_SUBGROUP 2C_NORMALIZER=PGGL 4 4_S

> -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,
1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1,1,
1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,0,
1,0,0,0,0,1,0,0,0,0,1,0,1,0,0,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,0,0,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,0,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,0,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,1,0,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,1,1,0,
1,0,0,0,0,1,0,0,0,0,1,1,1,1,1,1,1,1,
1,0,3,0,1,1,1,3,0,0,2,0,0,0,0,2,1,"
PGGL_4_4_SUBGROUP_5A=-PGGL_4_4\ 
格力(GG)

PGGL_4_4_SUBGROUP_5A_NORMALIZER=-PGGL_4_4\ 
格力(GG)

PGGL_4_4_SUBGROUP_5B=-PGGL_4_4\ 
格力(GG)

PGGL_4_4_SUBGROUP_2Cx2_0=-PGGL_4_4\ 
格力(GG)

PGGL_4_5_SUBGROUP_3B=-PGGL_4_5\ 
格力(GG)
# subgroup_by_generators."3B".3:1

"1,0,0,0,-01,0,0,0,0,0,2,1,0,0,3,2"

# PGL.4.5_SUBGROUP_3B_NORMALIZER=-PGL.4.5

# subgroup_by_generators."normalizer_3B"."5760".8

"1,0,0,0,0,4,0,0,0,0,1,0,0,0,0,1,"

"1,0,0,0,0,4,0,0,0,0,4,0,0,0,0,4,"

"1,0,0,0,0,4,0,0,0,0,3,0,0,0,0,3,"

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

"0,1,0,0,1,0,0,0,0,0,4,0,0,0,0,4,"

# elementary_abelian_subgroups_of_order_4_with_3_elements_of_class_2C:

# nice_generators, from Michael Epstein:

PGL.4.5_SUBGROUP_3B_ME=-PGL.4.5

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

PGL.4.5_SUBGROUP_31_ME=-PGL.4.5

"1,0,0,0,0,1,0,0,0,1,0,1,1,3,1,3,"
# subgroup of order 31 for the construction of regular packings in PG_3_5:
PGL_4_5_SUBGROUP_31=-PGL_4_5\`
PGL_4_5_SUBGROUP_31_NORMALIZER=-PGL_4_6\`

# Exterior square roots:

# elt of order 3:
# the exterior square root of f is X=
# elt of order 31:
# the exterior square root of g is Z=

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,\n0,1,1,0,1,1,1,\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,\n0,0,0,1,1,1,1"

HAMMING_CODE_ROWS_IN_BINARY_RANKS="67,37,22,15"

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"

CODE_GV_N15_K6="\n111111111100000\n11110000010000\n11001100001000\n10101010000100\n1010110000010\n10110100100001"

CODE_GV_N15_K6_CHECK="\n100000000111111\n010000000111100\n001000000111011\n000100000110101\n000010000110010\n000001000101101\n000000100101010\n000000010100110\n000000001100001"
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: "8,16,32,24,40,48,56,1,2,4,3,5,6,7,9,18,36,27,45,54,63,33,42,20,11,53,62,31"

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="8,9,12,13,24,25,28,29,10,11,14,15,26,27,30,31,40,41,44,45,46,47,58,59,62,63"

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="32,34,48,50,33,35,49,51,36,38,52,54,37,39,53,55,40,42,44,46,58,59,62,63"

AG_2_3_BLOCKS="0,13,22,27,35,41,47,53,55,59,71,76"


TEST_SYSTEM="\n0,1,0,1,0,0,\n0,0,1,0,1,0,\"
abel."group_cyclic"

DELANDTSHEER_DOYEN_PROBLEM_3_7

MASK1=-mask_label."mask1"-depth:5

PENTTILA_WILLIAMS_PRINCE_REG

PACKING_0="444,43313,154402,46682,108254,75363,27729
,32139,5244,60442,142811,111115,94209,129953,82168,1368
38,19253,23017,145985,134996,54705,36267,55066,117542,96699,69154,72460"

1055

PENTTILA_WILLIAMS_PRINCE_REG

PACKING_1="616,42728,152655,48576,105431,79607,28634
,32817,9799,62356,141176,110085,92557,122136,86312,13975,101942,126869,81478,1393
52,18028,24325,147284,130370,52074,36843,55602,118454,95973,69642,74036"

PENTTILA_WILLIAMS_PRINCE_REG

PACKING_0_DUAL="3938,66740,56555,93538,107785,6
4917,47567,54483,141012,138602,18308,6880,131351,88788,125484,102075,2
1234,99392,119149,80640,124839,148843,71862,11468,35950,27050,75338,11
3337,40002,154102,30567"

PG_3_5

PACKING_0_WITH_AGO3="0,5201,60427,86602,11453,121452,46663,19716,32921,108
680,23456,91963,68386,26921,74601,57067,36188,42312,78780,53117,118488,114700,839
60,99669,104791,126662,130960,145179,137230,150626,140216"

PG_3_5

PACKING_0_WITH_AGO3_FIXP444="444,5001,12957,18194,23485,26817,34667,38299,
41249,47472,50450,56601,62638,68986,71833,75369,80805,87025,92577,95676,104509,10
9718,114948,116333,124391,127498,133240,137711,144777,148059,150175"

#·consider-the-binary-code-with-generator-matrix:

#·1-0-1
#·0-1-1
CODE_N3_K2_Q2_GENMA="1,0,1,·0,1,1"

CODE_N6_K3_Q2_GENMA="\n111100\n110010\n101001"

TRIANGLE_GRAPH="0,1,1\n0,1\n1,1,0\n"

#·q=17:
#-3-3p.e. mod. 17.
# so we pick f = 3.
# then 2f^2 = 18 = 1
# 4f = -12

# x^4 - y^4 - z^4 + 2f z y^2 z + 4f x z y = 0

# (1, -1, 1, 0, 0, 0, 0, 0, 0, 0, 2f, 0, 2f, 0, 0)

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"

FILE_Q17 = "orbit, curve, pts.on.curve, bitangents, go",
"$(EDGE_CURVE_Q17_EQUATION)" ,"$(EDGE_CURVE_Q17_AS_POINTS)" ,"\", -1",
"nEND"


MONKEY_SADDLE_CUBIC = "1,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0"

ECKARDT_CUBIC_DEFORM1_LEX = "0,10,0,0,10,25,2,0,-20,-8,-20,-10,-24,-10 ,0,12,0,-8,-8,16".

ECKARDT_CUBIC_DEFORM2_LEX = "0,-5,0,-5,-5,10,-1,0,10,4,10,5,3,-5,1,- 6,0,-5,-4,1"

KUMMER_QUARTIC_LEX_35 = "-2,0,0,0,2,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,2,0,-2"

BEAUVILLE_QUINTIC_LEX_56 = "-4,928,400,315,-396,-852,\ 
-412,953,-1050,-354,284,-504,-62,-707,-1390,-1010,\ 
124,0,167,-1644,-1024,-72,-196,192,-373,322,78,150,\ 
125,966,1540,348,-475,-492,1063,1550,390,-96,3,-337,\ 
126,-426,-66,425,-673,-156,-216,-223,-60,1543,1998,618,\ 
127,263,-250,-919,557,1800,741"

ENDRASS_OCTIC_LEX_165 = "-93.2548,0,0,0,0,0,-309.019,0,0,527.529,0,395.647,\ 
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,\ 
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-309.019,\ 

Section 2.2: Orbiter Session

Example set:

```
$(ORBITER_PATH)orbiter.out
-define -S-set -here:2,3,5,7,11,13
-print_symbols
```
object_F_2:
$(ORBITER_PATH)orbiter.out\-v.3\-define F\-finite_field\-q 2\-end

object_PG_3_2:
$(ORBITER_PATH)orbiter.out\-v.3\-define F\-finite_field\-q 2\-end
-define P\-projective_space:3\-F\-end

vector_ex:
$(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

create BLT 5 1:
$(ORBITER_PATH)orbiter.out\-v.2\-define F\-finite_field\-q 5\-end
-define 0\-orthogonal_space:0 5\-F\-end
-with 0\-do\-orthogonal_space_activity
-create BLT_set\-catalogue:1\-end

create 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 S4_0\-q 4\-catalogue:0\-end
-with S4_0\-do
# Section 2.6: Set-Builder

**SECTION_SET_BUILDER:**

**set_of_primes:**

```bash
$(ORBITER_PATH)orbiter.out -v 2
```

```bash
$ define S -set -here "2,3,5,7,11,13" -end
```

```bash
-print_symbols
```

**set_interval:**

```bash
$(ORBITER_PATH)orbiter.out -v 2 -define S -set -loop 0 64 1 -end
```

```bash
-print_symbols
```

**set_builder_examples:**

```bash
$(ORBITER_PATH)orbiter.out -v 2 -long_code 64 7 -end
```

```bash
-set_builder -loop 0 64 1 -end
```

```bash
-set_builder -loop 0 32 1 -affine_function 2 1 -end
```

```bash
-set_builder -loop 0 16 1 -end
```

```bash
-affine_function 4 2 -clone_with_affine_function 4 3 -end
```

```bash
-set_builder -set_builder -set_builder -end
```

```bash
-loop 0 4 1 -affine_function 1 4 -end
```

```bash
-clone_with_affine_function 1 12 -end
```

```bash
-clone_with_affine_function 1 16 -end
```

```bash
-end -clone_with_affine_function 1 32 -end
```

```bash
-set_builder -set_builder -loop 0 8 1 -affine_function 1 8 -end
```

```bash
-clone_with_affine_function 1 24 -end
```

```bash
-clone_with_affine_function 1 32 -end
```

```bash
-set_builder -loop 0 16 1 -affine_function 1 16 -end
```

```bash
-clone_with_affine_function 1 48 -end
```

```bash
-set_builder -loop 0 32 1 -affine_function 1 32 -end
```

# Section 2.7: Vector-Builder

**SECTION_VECTOR_BUILDER:**
vector_example1:
-define F-finite_field-q5-end\n-define v-vector-field F-dense"0,1,2,3,4"-end\n-print_symbols

vector_example2:
-define F-finite_field-q5-end\n-define v-vector-field F-format-2-dense"0,1,2,3,4,0"-end\n-print_symbols

vector_example_sparse:
-define F-finite_field-q5-end\n-define v-vector-field F-format-4-sparse:20"1,0,1,19"-end\n-print_symbols

vector_example_repeat:
-define F-finite_field-q5-end\n-define v-vector-repeat"0,1,2,3"11-end\n-print_symbols

vector_example_all_one_11:
-define F-finite_field-q2-end\n-define v-vector-repeat11-end\n-print_symbols

matrix_example1:
-define F-finite_field-q2-end\n-define v-vector-field F-format-4\n-dense $(HAMMING_CODE_GENERATOR)-end\n-print_symbols

matrix_example_co_1:
-define F-finite_field-q2-end\n-define v-vector-field F-format-22\n-compact $(CONWAY_GEN1)-end
### Section 2.8: Input Streams

**SECTION_INPUT_STREAMS:**

---

### Chapter 3: Basic Algebra

---

### Section 3.1: Basic Number Theory

**SECTION_BASIC_NUMBER_THEORY:**

---

```plaintext
$\text{(ORBITER_PATH)orbiter.out -v -order_of_q_mod_n 2 3 151}$
```

---

**PR7:**

```plaintext
\text{(ORBITER_PATH)orbiter.out -v -smallest.primitive.root 7}
```

---

**PR11:**

```plaintext
\text{(ORBITER_PATH)orbiter.out -v -smallest.primitive.root 11}
```

---

**PR13:**

```plaintext
\text{(ORBITER_PATH)orbiter.out -v -smallest.primitive.root 13}
```

---

**PR17:**

```plaintext
\text{(ORBITER_PATH)orbiter.out -v -smallest.primitive.root 17}
```

---

**PR19:**

```plaintext
\text{(ORBITER_PATH)orbiter.out -v -smallest.primitive.root 19}
```

---

**PR23:**

```plaintext
460
```
PR29: $(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot1\cdot\text{-smallest\_primitive\_root}\cdot23$

PR31: $(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot1\cdot\text{-smallest\_primitive\_root}\cdot29$

PR37: $(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot1\cdot\text{-smallest\_primitive\_root}\cdot31$

PR100: $(\text{ORBITER PATH})\text{orbiter.out}\cdot-v\cdot1\cdot\text{-smallest\_primitive\_root\_interval}\cdot2\cdot100$

Eulerfunction150:

PR1000:

PE_number1000:

Eulerfunction10000:

power_function.2.mod.11:
$\text{randomized algo:}$

$\text{PR2:}$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-primitive_root}\cdot 915839$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-primitive_root}\cdot \text{modulo}\cdot 915839\cdot \text{is}\cdot 43085$

$\text{PM2:}$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-power_mod}\cdot 43085\cdot 49842\cdot 915839$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-power_mod}\cdot 43085\cdot 49842\cdot 915839\cdot \text{is}\cdot 487320$

$\text{DL2:}$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-discrete_log}\cdot 487320\cdot 43085\cdot 915839$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-discrete_log}\cdot 487320\cdot 43085\cdot 915839\cdot \text{is}\cdot 49842$

$\text{time: 0:22}$

$\text{IM:}$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-inverse_mod}\cdot 723\cdot 4060$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-inverse_mod}\cdot 3\cdot 19$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-inverse_mod}\cdot 1865025205\cdot 2147483647$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-extended_gcd}\cdot 1865025205\cdot 2147483647$

$\text{PM3a:}$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 5\cdot \text{-power_mod}\cdot 16807\cdot 1073741823\cdot 2147483647$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 2\cdot \text{-square_root_mod}\cdot 33\cdot 41$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 2\cdot \text{-square_root_mod}\cdot 5\cdot 11$

$\text{⊿ ⊿ } (\text{ORBITER PATH})\text{orbiter.out -v}\cdot 2\cdot \text{-square_root_mod}\cdot 5\cdot 19$
draw_mod_8:
  $(ORBITER_PATH) orbiter.out -v 2 -draw_options -embedded -end \-
  -draw_mod_n -n 8 -file_mod_8 -end
  pdflatex-mod_8.draw.tex
  open-mod_8_draw.pdf

draw_mod_13:
  $(ORBITER_PATH) orbiter.out -v 2 \-
  -draw_options -embedded -end \-
  -draw_mod_n -n 13 -file_mod_13 -power_cycle 2 -end
  pdflatex-mod_13.draw.tex
  open-mod_13_draw.pdf

draw_mod_3:
  $(ORBITER_PATH) orbiter.out -v 2 \-
  -draw_options -embedded -nodes_empty -end \-
  -draw_mod_n -n 3 -file_mod_3 -end.
  pdflatex-mod_3.draw.tex
  open-mod_3_draw.pdf

draw_mod_3_c:
  $(ORBITER_PATH) orbiter.out -v 2 \-
  -draw_options -embedded -nodes_empty -end \-
  -draw_mod_n -n 3 -file_mod_3_c \-
  -cyclotomic_sets 2 "1" -end
  pdflatex-mod_3_c.draw.tex
  open-mod_3_c_draw.pdf

draw_mod_4:
  $(ORBITER_PATH) orbiter.out -v 2 \-
  -draw_options -embedded -nodes_empty -end \-
  -draw_mod_n -n 4 -file_mod_4 -end.
  pdflatex-mod_4.draw.tex
  open-mod_4_draw.pdf

draw_mod_6:
  $(ORBITER_PATH) orbiter.out -v 2 \-
  -draw_options -embedded -nodes_empty -end \-
  -draw_mod_n -n 6 -file_mod_6 -end.
  pdflatex-mod_6.draw.tex
  open-mod_6_draw.pdf

draw_mod_7:
draw_mod_15:
  ▷ ▷ $(\text{ORBITER\_PATH})\text{orbiter.out-v.2}$\ \$
  ▷ ▷ \text{-draw\_options:}\text{-embedded-}\text{-nodes\_empty-}\text{-end}\$
  ▷ ▷ ▷ \text{-draw\_mod\_n-n.7-}\text{-file-mod\_7-}\text{-end}\$
  ▷ ▷ ▷ \text{pdflatex}\text{-mod}_7\text{-draw.tex}$
  ▷ ▷ ▷ \text{open-mod}_7\text{-draw.pdf}$
  ▷ ▷ ▷ \text{pdflatex}\text{-mod}_7\text{-draw.tex}$
  ▷ ▷ ▷ \text{open}\text{-mod}_7\text{-draw.pdf}$

draw_mod_127:
  ▷ ▷ $(\text{ORBITER\_PATH})\text{orbiter.out-v.2}$\ \$
  ▷ ▷ \text{-draw\_options:}\text{-scale}:0.8\text{-embedded-}\text{-end}\$
  ▷ ▷ ▷ \text{-draw\_mod\_n-n.15-}\text{-file-mod\_15-}\text{-end}\$
  ▷ ▷ ▷ \text{pdflatex}\text{-mod}_15\text{-draw.tex}$
  ▷ ▷ ▷ \text{open-mod}_15\text{-draw.pdf}$
  ▷ ▷ \text{pdflatex}\text{-mod}_15\text{-draw.tex}$
  ▷ ▷ \text{open}\text{-mod}_15\text{-draw.pdf}$

sqrt_mod_20\_31:
  ▷ ▷ $(\text{ORBITER\_PATH})\text{orbiter.out-v.2-}\text{-square\_root\_mod-20\_31}$
  ▷ ▷ ▷ \text{pdflatex-GF}_2\text{.tex}$
  ▷ ▷ ▷ \text{open-GF}_2\text{.pdf}$

F_2:
  ▷ $(\text{ORBITER\_PATH})\text{orbiter.out-v.3-}\text{-list\_arguments}$\$
  ▷ ▷ \text{-define}\text{-finite\_field-q.2-}\text{-end}$
  ▷ ▷ ▷ \text{-with}\text{-do-}\text{-finite\_field\_activity-}\text{-cheat\_sheet\_GF}$\text{-end}$
  ▷ ▷ ▷ \text{pdflatex-GF}_2\text{.tex}$
  ▷ ▷ ▷ \text{open-GF}_2\text{.pdf}$

F_3:
1553  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3$
1554  \text{define F--finite\_field--q.3--end}\$
1555  \text{with F--do--finite\_field\_activity--cheat\_sheet\_GF--end}
1556  \#pdflatex-GF_3.tex
1557  \#open-GF_3.pdf
1558
1559  F_5:
1560  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3$
1561  \text{define F--finite\_field--q.5--end}\$
1562  \text{with F--do--finite\_field\_activity--cheat\_sheet\_GF--end}
1563  \text{pdflatex-GF_5.tex}
1564  \text{open-GF_5.pdf}
1565
1566  F_7:
1567  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3$
1568  \text{define F--finite\_field--q.7--end}\$
1569  \text{with F--do--finite\_field\_activity--cheat\_sheet\_GF--end}
1570  \text{pdflatex-GF_7.tex}
1571  \text{open-GF_7.pdf}
1572
1573  F_13:
1574  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3$
1575  \text{define F--finite\_field--q.13--end}\$
1576  \text{with F--do--finite\_field\_activity--cheat\_sheet\_GF--end}
1577  \text{pdflatex-GF_13.tex}
1578  \text{open-GF_13.pdf}
1579
1580
1581
1582
1583  F_17:
1584  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3$
1585  \text{define F--finite\_field--q.17--end}\$
1586  \text{with F--do--finite\_field\_activity--cheat\_sheet\_GF--end}
1587  \text{pdflatex-GF_17.tex}
1588  \text{open-GF_17.pdf}
1589
1590
1591
1592
1593
1594
1595
1596  F_19:
1597  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3$
1598  \text{define F--finite\_field--q.19--end}\$
1599  \text{with F--do--finite\_field\_activity--cheat\_sheet\_GF--end}
\begin{verbatim}
1600   \$ pdflatex GF_19.tex
1601   \$ open GF_19.pdf
1602
1603 F_31:  
1604   \$(ORBITER\_PATH)orbiter.out-v.3\$
1605   \$\text{define F\text{-finite_field\text{-q\text{-31}}\text{-end}}\$
1606   \$\text{with F\text{-do\text{-finite_field\text{-activity\text{-cheat_sheet\text{-GF\text{-end}}}}}}\$
1607
1608 F_127: 
1609   \$(ORBITER\_PATH)orbiter.out-v.3\$
1610   \$\text{define F\text{-finite_field\text{-q\text{-127}}\text{-end}}\$
1611   \$\text{with F\text{-do\text{-finite_field\text{-activity\text{-cheat_sheet\text{-GF\text{-end}}}}}}\$
1612
1613 F_11\_product\_of\_all\_nonzero\_elements:  
1614   \$(ORBITER\_PATH)orbiter.out-v.3\$
1615   \$\text{define F\text{-finite_field\text{-q\text{-11}}\text{-end}}\$
1616   \$\text{define S\text{-vector\text{-field\text{-F\text{-loop\text{-11\_1}}\text{-end}}}}\$
1617   \$\text{with F\text{-do\text{-finite_field\text{-activity\text{-product\_of\_S\text{-end}}}}}}\$
1618
1619
1620 F_7\_vandermonde:  
1621   \$(ORBITER\_PATH)orbiter.out-v.3\$
1622   \$\text{define F\text{-finite_field\text{-q\text{-7}}\text{-end}}\$
1623   \$\text{with F\text{-do\text{-finite_field\text{-activity\text{-Vandermonde\_matrix}}}}\$
1624   \$\text{-end}\$
1625
1626
1627
1628
1629
1630
1631
1632 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1633 # Section 3.3: Polynomials over Finite Fields
1634
1635 SECTION\_POLYNOMIALS:  
1636
1637
1638
1639
1640 % check which polynomials are irreducible and which are primitive:  
1641
1642 sift\_polynomials\_deg3\_q2:  
1643   \$(ORBITER\_PATH)orbiter.out-v.2\$
1644   \$\text{define F\text{-finite_field\text{-q\text{-2}}\text{-end}}\$
1645   \$\text{with F\text{-do}}\$
1646   \$\text{-finite_field\text{-activity\text{-sift\_polynomials\_8\_16\text{-end}}}}\$
\end{verbatim}
sift_polynomials_deg4.q2:

poly_division:

poly_division2:

poly_gcd:

poly_mult_mod1:

poly_mult_mod2:
poly_mult_mod_F4:

poly_mult_mod: "3,1,2"."5,3,4"."6,0,0,1".

poly_mult_mod: "3,1,2"."5,3,4"."6,0,0,1".

poly_mult_mod: "1,1"."1,1"."1,1,1".

poly_mult_mod: "0,1"."0,1"."1,1,1".

poly_mult_mod: "0,1"."0,1"."1,1,1".

mult_polynomials_2_5_7:

mult_polynomials_2_5_7:

mult_polynomials_2_5_7:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:

polynomial_division_ranked_2_27_13:
\input{orbiter\_out}
$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_8\cdot 15\cdot end$
\input{polynomial\_mult\_8\_15.tex}$open\_polynomial\_mult\_8\_15.pdf$
\input{orbiter\_out}
$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_7\cdot 7\cdot end$
\input{polynomial\_mult\_7\_7.tex}$open\_polynomial\_mult\_7\_7.pdf$
\input{orbiter\_out}
$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_4\cdot 6\cdot end$
\input{polynomial\_mult\_4\_6.tex}$open\_polynomial\_mult\_4\_6.pdf$
\input{orbiter\_out}
$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_2\cdot 8\cdot 15\cdot end$
\input{polynomial\_mult\_2\_8\_15.tex}$open\_polynomial\_mult\_2\_8\_15.pdf$
\input{orbiter\_out}$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_4\cdot 6\cdot end$
\input{polynomial\_mult\_4\_6.tex}$open\_polynomial\_mult\_4\_6.pdf$
\input{orbiter\_out}$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_2\cdot 7\cdot 7\cdot end$
\input{polynomial\_mult\_2\_7\_7.tex}$open\_polynomial\_mult\_2\_7\_7.pdf$
\input{orbiter\_out}$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_2\cdot 4\cdot 6\cdot end$
\input{polynomial\_mult\_2\_4\_6.tex}$open\_polynomial\_mult\_2\_4\_6.pdf$
\input{orbiter\_out}$\cdot v \cdot 2 \cdot$
\input{finite\_field-\_q\_2\_\_end}$\cdot F \cdot finite\_field \cdot q \cdot 2 \cdot end$
\input{finite\_field-\_do}$finite\_field\_activity \cdot mult\_polynomials\_2\cdot 4\cdot 6\cdot end$
\input{polynomial\_mult\_2\_4\_6.tex}$open\_polynomial\_mult\_2\_4\_6.pdf$
1788  \end{verbatim}
1789  \pdflatex polynomial\_division\_24\_13.tex
1790  \open\-polynomial\_division\_24\_13.pdf
1791
1792
1793 mult\_polynomials\_1024\_999\_997:
1795  $(\text{ORBITER\_PATH})$orbiter.out\(-v\_2:\$
1796  \\end{verbatim}
1797  \end{verbatim}
1798  \end{verbatim}
1799  \begin{verbatim}
1800  \end{verbatim}
1801  \pdflatex polynomial\_mult\_999\_997.tex
1802  \open\-polynomial\_mult\_999\_997.pdf
1803
1804 polynomial\_division\_ranked\_2\_349147\_1033:
1806  $(\text{ORBITER\_PATH})$orbiter.out\(-v\_2:\$
1808  \\end{verbatim}
1809  \end{verbatim}
1810  \end{verbatim}
1811  \begin{verbatim}
1812  \end{verbatim}
1813  \pdflatex polynomial\_division\_349147\_1033.tex
1814  \open\-polynomial\_division\_349147\_1033.pdf
1815
1816 mult\_polynomials\_1024\_999\_997\_check:
1818  $(\text{ORBITER\_PATH})$orbiter.out\(-v\_3:\$
1820  \\end{verbatim}
1821  \end{verbatim}
1822  \end{verbatim}
1823  \begin{verbatim}
1824  \end{verbatim}
1825  \# evaluates to 61
1827 mult\_polynomials\_17\_12:
1829  $(\text{ORBITER\_PATH})$orbiter.out\(-v\_2:\$
1831  \\end{verbatim}
1832  \end{verbatim}
1833  \end{verbatim}
1834  \pdflatex polynomial\_mult\_17\_12.tex
1835  \open\-polynomial\_mult\_17\_12.pdf
470
1835 ▷
1836 # gives 204
1837
1838 polynomial_division_ranked_2_204_37:
1839 ▷ $(ORBITER_PATH)orbiter.out -v.2\n1840 ▷ ▷ -define F=finite_field-q.2-end\n1841 ▷ ▷ -with F-do\n1842 ▷ ▷ -finite_field_activity\n1843 ▷ ▷ ▷ -polynomial_division_ranked_204_37-\n1844 ▷ ▷ -end
1845 ▷ pdflatex-polynomial_division_204_37.tex
1846 ▷ open-polynomial_division_204_37.pdf
1847 ▷
1848 # answer is 18
1849 ▷
1850 ▷
1851 1852
1853 1854 1855
1856 # Section 3.4: Extension Fields
1857
1858 SECTION_EXTENSION_FIELDS:
1859 ▷
1860 ▷
1861 1862
1863 1864 F_4:
1865 ▷ $(ORBITER_PATH)orbiter.out -v.3\n1866 ▷ ▷ -define F=finite_field-q.4-end\n1867 ▷ ▷ -with F-do-finite_field_activity-cheat_sheet_GF-end
1868 1869
1870 F_4_tables:
1871 ▷ $(ORBITER_PATH)orbiter.out -v.3\n1872 ▷ ▷ -define F=finite_field-q.4-end\n1873 ▷ ▷ -with F-do-finite_field_activity-cheat_sheet_GF-end\n1874 ▷ ▷ -draw_matrix-input_csv_file GF_q4_multiplication_table.csv\n1875 ▷ ▷ ▷ -box.width 40-bit.depth 24-partition 3.3.3-end
1876 ▷ ▷ -draw_matrix-input_csv_file GF q4_multiplication_table.csv\n1877 ▷ ▷ ▷ -box_width 40-bit_depth 24-partition 3.3.3-end
1878 ▷ # pdflatex-GF_4.tex
1879 ▷ # open-GF_4.pdf
1880 1881 trace_4:
\begin{verbatim}
1882  \$\{\texttt{ORBITER\_PATH}\texttt{or}b\texttt{iter.out}}\texttt{-v.3}::\n1883  \>
1884  \>
1885  \>
1886  \>
1887  \>
1888  \>
1889  \>
1890  \>
1891  \>
1892  \>
1893  \texttt{trace\_4\_WH\_transform}::\n1894  \$\{\texttt{ORBITER\_PATH}\texttt{or}b\texttt{iter.out}}\texttt{-v.3}::\n1895  \>
1896  \>
1897  \>
1898  \>
1899  \>
1900  \>
1901  \texttt{F.8}::\n1902  \$\{\texttt{ORBITER\_PATH}\texttt{or}b\texttt{iter.out}}\texttt{-v.3}::\n1903  \>
1904  \>
1905  \>
1906  \>
1907  \>
1908  \>
1909  \>
1910  \>
1911  \texttt{F8\_arithmetic}::\n1912  \$\{\texttt{ORBITER\_PATH}\texttt{or}b\texttt{iter.out}}\texttt{-v.3}::\n1913  \>
1914  \>
1915  \>
1916  \>
1917  \>
1918  \>
1919  \>
1920  \>
1921  \texttt{F.16.7\_power}::\n1922  \$\{\texttt{ORBITER\_PATH}\texttt{or}b\texttt{iter.out}}\texttt{-v.3}::\n1923  \>
1924  \>
1925  \>
1926  \>
1927
\end{verbatim}
the-answer-is-11.

F8_near_bent_5:

\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\}}\)
\(\text{\texttt{-define F:\text{-finite\_field}\:-q\text{-8}-end\}}\)
\(\text{\texttt{-with F:\text{-do\_finite\_field\_activity\}}\}
\(\text{\texttt{-identity\_function\:F8.csv\}}\)
\(\text{\texttt{-end\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\}}\)
\(\text{\texttt{-define F:\text{-finite\_field\:-q\text{-8}-end\}}\)
\(\text{\texttt{-with F:\text{-do\_finite\_field\_activity\}}\}
\(\text{\texttt{-apply\_power\_function\:F8.csv\_5\}}\)
\(\text{\texttt{-end\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\}}\)
\(\text{\texttt{-define F:\text{-finite\_field\:-q\text{-2}-end\}}\)
\(\text{\texttt{-with F:\text{-do\_finite\_field\_activity\}}\}
\(\text{\texttt{-Walsh\_Hadamard\_transform\}}\)
\(\text{\texttt{-F8\_power\_5\_trace\_csv\_4\}}\)
\(\text{\texttt{-end\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\}}\)
\(\text{\texttt{-input\_csv\_file\:F8\_power\_5\_trace\_8x1.csv\}}\)
\(\text{\texttt{-partition\:4-8-1\}}\)
\(\text{\texttt{-end\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.2\}}\)
\(\text{\texttt{-box\_width\:40\_\text{-bit\_depth\:24\}}\)
\(\text{\texttt{-partition\:4-8-8\_\text{-end\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\}}\)
\(\text{\texttt{-reformat\_F8\_power\_5\_trace\_transformed\_csv\}}\)
\(\text{\texttt{-F8\_power\_5\_trace\_transformed\_8x1.csv\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.2\}}\)
\(\text{\texttt{-box\_width\:40\_\text{-bit\_depth\:24\_\text{-partition\:4-8-1\_\text{-end\}}\}
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\}}\)
\(\text{\texttt{-bent\_4\}}\)
\(\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-v.3\_\text{-bent\_4\}}\)

473
F_9:

1989 \#X0*X3*X4^2+X1*X2*X4^2

1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018 $\triangleright$ $\triangleright$ -with F-do-finite_field_activity-trace-end
2019 $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.3\`
2020 $\triangleright$ $\triangleright$ -reformat F_q9_trace.csv-F_q9_trace_3x3.csv-3
2021 $\triangleright$ $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.2-draw_matrix\`
2022 $\triangleright$ $\triangleright$ -input_csv_file-F_q9_trace_3x3.csv-
2023 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2024 $\triangleright$
2025
2026
2027
2028
2029
2030 $F_{16}$:
2031 $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.3\`
2032 $\triangleright$ $\triangleright$ -define F-finite_field-q.16-end\`
2033 $\triangleright$ $\triangleright$ -with F-do-finite_field_activity-cheat_sheet_GF-end
2034 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2035 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2036
2037 $F_{16}$ tables:
2038 $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.3\`
2039 $\triangleright$ $\triangleright$ -define F-finite_field-q.16-end\`
2040 $\triangleright$ $\triangleright$ -with F-do-finite_field_activity-cheat_sheet_GF-end
2041 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2042 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2043 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2044 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2045 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2046 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2047 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2048 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2049 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2050 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2051 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2052 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2053 $\triangleright$ $\triangleright$ -reformat F_{16_trace}csv-F_{16_trace_3x3}.csv-4
2054 $\triangleright$ $\triangleright$ -box_width-40-bit_depth-24-partition-4-3-3-end
2055
2056
2057 trace_16:
2058 $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.3\`
2059 $\triangleright$ $\triangleright$ -define F-finite_field-q.16-end\`
2060 $\triangleright$ $\triangleright$ -with F-do-finite_field_activity-trace-end
2061 $\triangleright$ $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.3\`
2062 $\triangleright$ $\triangleright$ -reformat F_q16_trace.csv-F_q16_trace_4x4.csv-4
2063 $\triangleright$ $\triangleright$ $(\text{ORBITER PATH})$ orbiter.out-v.2-draw_matrix\`
2064 $\triangleright$ $\triangleright$ -input_csv_file-F_q16_trace_4x4.csv-
F_16.bent_wrong:

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-3}\backslash$

$\text{-define\,F\,-finite\,field\,-q\,16\,-end}\backslash$

$\text{-with\,F\,-do\,-finite\,field\,activity}\backslash$

$\text{-identity\,function\,F16\,csv\,-end}\\$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-3}\backslash$

$\text{-define\,F\,-finite\,field\,-q\,16\,-end}\backslash$

$\text{-with\,F\,-do\,-finite\,field\,activity}\backslash$

$\text{-apply\,trace\,function\,F16\,power\,9\,csv\,-end}\backslash$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-3}\backslash$

$\text{-define\,F\,-finite\,field\,-q\,2\,-end}\backslash$

$\text{-with\,F\,-do\,-finite\,field\,activity}\backslash$

$\text{-Walsh\,Hadamard\,transform\,F16\,power\,9\,trace\,csv\,-4\,-end}\\$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-3}\backslash$

$\text{-reformat\,F16\,power\,9\,trace\,csv\,-F16\,power\,9\,trace\,16x1\,csv\,-1}\\$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-2\,-draw\,matrix}\backslash$

$\text{-input\,csv\,file\,F16\,power\,9\,trace\,16x1\,csv}\backslash$

$\text{-box\,width\,40\,-bit\,depth\,24\,-partition\,4\,16\,1\,-end}\\$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-2\,-draw\,matrix}\backslash$

$\text{-input\,csv\,file\,Walsh\,01\,4\,csv}\backslash$

$\text{-box\,width\,40\,-bit\,depth\,24\,-partition\,4\,16\,1\,-end}\\$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-3}\backslash$

$\text{-reformat\,F16\,power\,9\,trace\,transformed\,csv\,-}\\$

$\text{F16\,power\,9\,trace\,transformed\,16x1\,csv\,-1}\\$

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-2\,-draw\,matrix}\backslash$

$\text{-input\,csv\,file\,F16\,power\,9\,trace\,transformed\,16x1\,csv}\backslash$

$\text{-box\,width\,40\,-bit\,depth\,24\,-partition\,4\,16\,1\,-end}\\$

F_16\,over\,F_4\,field\,reduction:

$\textsc{ORBITER\_PATH}\\text{orbiter.out\,-v\,-2}\backslash$

$\text{-define\,F\,-finite\,field\,-q\,16\,-end}\backslash$

$\text{-loop\,L\,0\,16\,1}\backslash$

$\text{-with\,F\,-do}\backslash$

$\text{-finite\,field\,activity}\backslash$

$\text{-field\,reduction\,"\text{F16\,over\,F4\,}L\,4\,1\,1\,\%L\,\text{\,-end}\,}\\$

$\text{-end_loop}\backslash$

$\text{-loop\,L\,0\,16\,1}\backslash$

$\text{-draw\,matrix\,-input\,csv\,file\,F16\,over\,F4\,\%L\,csv}\backslash$

$\text{-box\,width\,40\,-bit\,depth\,24\,-partition\,4\,2\,2\,-end}\\$

476
2112  \>  \>  \>  \-end_loop
2113
2114
2115
2116
2117  \# the polynomial 31 is not primitive:
2118
2119  F_{16}.poly31:
2120  \>  $$\$(\text{ORBITER\_PATH})\text{or}
2121  \>  \>  \>  \text{bit}_\text{\textout}\text{-v.3}\text{\textbackslash
2122  \>  \>  \>  \text{-define}\text{-finite_field}\text{-q.16}\text{-override}\text{polynomial}\text{-31}\text{-end}\text{\textbackslash
2123  \>  \>  \>  \text{-with}\text{-do}\text{-finite_field_activity}\text{\textbackslash
2124  \>  \>  \>  \>  \text{-cheat}\text{sheet}\text{GF}\text{-end}
2125
2126
2127
2128
2129
2130
2131
2132
2133  F_{25}:
2134  \>  $$\$(\text{ORBITER\_PATH})\text{or}
2135  \>  \>  \>  \text{bit}_\text{\textout}\text{-v.2}\text{-draw_matrix}\text{\textbackslash
2136  \>  \>  \>  \text{-box_width}40\text{-bit_depth}24\text{-partition}3\text{\textbackslash
2137  \>  \>  \>  \text{-input_csv_file}\text{GF}_\text{q25}\text{addition_table.csv}\text{\textbackslash
2138  \>  \>  \>  \text{-input_csv_file}\text{GF}_\text{q25}\text{multiplication_table.csv}\text{\textbackslash
2139  \>  \>  \>  \text{-box_width}40\text{-bit_depth}24\text{-partition}3\text{\textbackslash
2140  \>  \>  \>  \text{-input_csv_file}\text{GF}_\text{q25}\text{addition_table_reordered.csv}\text{\textbackslash
2141  \>  \>  \>  \text{-input_csv_file}\text{GF}_\text{q25}\text{multiplication_table_reordered.csv}\text{\textbackslash
2142  \>  \>  \>  \text{-box_width}40\text{-bit_depth}24\text{-partition}3\text{\textbackslash
2143  \>  \>  \>  \text{-draw_matrix}\text{-end}
2144
2145
2146
2147
2148
2149
2150
2151  \text{trace}_\text{25}:
2152  \>  $$\$(\text{ORBITER\_PATH})\text{or}
2153  \>  \>  \>  \text{bit}_\text{\textout}\text{-v.3}\text{\textbackslash
2154  \>  \>  \>  \text{-define}\text{-finite_field}\text{-q.25}\text{-end}\text{\textbackslash
2155  \>  \>  \>  \text{-with}\text{-do}\text{-finite_field_activity}\text{-trace}\text{-end}
2156  \>  \>  \>  \>  $$\$(\text{ORBITER\_PATH})\text{or}
2157  \>  \>  \>  \>  \text{bit}_\text{\textout}\text{-v.2}\text{-draw_matrix}\text{\textbackslash
2158  \>  \>  \>  \>  \text{-input_csv_file}\text{GF}_\text{q25}\text{addition_table.csv}\text{\textbackslash

477
```
2159 ▶ ▶ -box_width-40.-bit_depth-24.-partition-3:25:25.-end
2160
2161
2162
2163 F_32:
2164 ▶ $(ORBITER_PATH)orbiter.out-v.3\n2165 ▶ ▶ -define-F.-finite_field-q.32.-end\n2166 ▶ ▶ -with-F.-do.-finite_field_activity-\n2167 ▶ ▶ -cheat_sheet_GF.-end
2168 ▶ pdflatex-GF_32.tex
2169 ▶ open-GF_32.pdf
2170
2171
2172
2173 F_49:
2174 ▶ $(ORBITER_PATH)orbiter.out-v.3\n2175 ▶ ▶ -define-F.-finite_field-q.49.-end\n2176 ▶ ▶ -with-F.-do.-finite_field_activity-\cheat_sheet_GF.-end
2177 ▶ $(ORBITER_PATH)orbiter.out-v.2.-draw_matrix-\n2178 ▶ ▶ -input_csv_file-GF_q49_addition_table.csv-\n2179 ▶ ▶ -box_width-40.-bit_depth-24.-partition-3:49:49.-end
2180 ▶ $(ORBITER_PATH)orbiter.out-v.2.-draw_matrix-\n2181 ▶ ▶ -input_csv_file-GF_q49_multiplication_table.csv-\n2182 ▶ ▶ -box_width-40.-bit_depth-24.-partition-3:48:48.-end
2183 ▶ $(ORBITER_PATH)orbiter.out-v.2.-draw_matrix-\n2184 ▶ ▶ -input_csv_file-GF_q49_addition_table_reordered.csv-\n2185 ▶ ▶ -box_width-40.-bit_depth-24.-partition-3:49:49.-end
2186 ▶ $(ORBITER_PATH)orbiter.out-v.2.-draw_matrix-\n2187 ▶ ▶ -input_csv_file-GF_q49_multiplication_table_reordered.csv-\n2188 ▶ ▶ -box_width-40.-bit_depth-24.-partition-3:48:48.-end
2189 ▶ #pdflatex-GF_49.tex
2190 ▶ #open-GF_49.pdf
2191
2192 trace_49:
2193 ▶ $(ORBITER_PATH)orbiter.out-v.3\n2194 ▶ ▶ -define-F.-finite_field-q.49.-end\n2195 ▶ ▶ -with-F.-do.-finite_field_activity-\trace_49-\end
2196 ▶ $(ORBITER_PATH)orbiter.out-v.3-\n2197 ▶ ▶ -reformat-F_q49_trace.csv-F_q49_trace_7x7.csv-7
2198 ▶ $(ORBITER_PATH)orbiter.out-v.2.-draw_matrix-\n2199 ▶ ▶ -input_csv_file-F_q49_trace_7x7.csv-\n2200 ▶ ▶ -box_width-40.-bit_depth-24.-partition-4:7:7.-end
2201
2202
2203
2204 F_64:
2205 ▶ $(ORBITER_PATH)orbiter.out-v.3-\n```
\begin{verbatim}
trace_64:
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -3\$
\end{verbatim}
\begin{verbatim}
2253 \texttt{\$(ORBITER\_PATH)orbiter.out-v.3\textbackslash{}
2254 \texttt{ \texttt{-define F\textunderscore finite\_field\textunderscore q\textunderscore 81\textunderscore end\textbackslash{}
2255 \texttt{ \texttt{-with F\textunderscore do\textunderscore finite\_field\_activity\textunderscore trace\textunderscore end\textbackslash{}
2256 \texttt{ $(ORBITER\_PATH)orbiter.out-v.3\textbackslash{}
2257 \texttt{ \texttt{-reformat F\textunderscore q81\_trace\textunderscore csv\textunderscore F\textunderscore q81\_trace\textunderscore 9x9\textunderscore csv\textminus{}9\textbackslash{}
2258 \texttt{ $(ORBITER\_PATH)orbiter.out-v.2\textunderscore draw\_matrix\textbackslash{}
2259 \texttt{ \texttt{-input\_csv\_file\textunderscore F\textunderscore q81\_trace\textunderscore 9x9\textunderscore csv\textbackslash{}
2260 \texttt{ \texttt{-box\_width\textunderscore 40\textunderscore bit\_depth\textunderscore 24\textunderscore partition\textunderscore 4\textunderscore 9\textunderscore 9\textunderscore end\textbackslash{}
2261 \texttt{ \texttt{\textbackslash{}
2262 \texttt{ F\textunderscore 125:\n2263 \texttt{ $(ORBITER\_PATH)orbiter.out-v.3\textbackslash{}
2264 \texttt{ \texttt{-define F\textunderscore finite\_field\textunderscore q\textunderscore 125\textunderscore end\textbackslash{}
2265 \texttt{ \texttt{-with F\textunderscore do\textunderscore finite\_field\_activity\textunderscore cheat\_sheet\_GF\textunderscore end\textbackslash{}
2266 \texttt{ \texttt{pdf\_latex\textunderscore GF\textunderscore 125\textunderscore tex\textbackslash{}
2267 \texttt{ open\_GF\textunderscore 125.pdf\textbackslash{}
2268 \texttt{ \texttt{\textbackslash{}
2269 \texttt{ \texttt{\textbackslash{}
2270 \texttt{ \texttt{\textbackslash{}
2271 \texttt{ F\textunderscore 256:\n2272 \texttt{ $(ORBITER\_PATH)orbiter.out-v.3\textbackslash{}
2273 \texttt{ \texttt{-define F\textunderscore finite\_field\textunderscore q\textunderscore 256\textunderscore end\textbackslash{}
2274 \texttt{ \texttt{-with F\textunderscore do\textunderscore finite\_field\_activity\textunderscore cheat\_sheet\_GF\textunderscore end\textbackslash{}
2275 \texttt{ $(ORBITER\_PATH)orbiter.out-v.2\textunderscore draw\_matrix\textbackslash{}
2276 \texttt{ \texttt{-input\_csv\_file\textunderscore GF\textunderscore q256\_addition\_table\textunderscore csv\textbackslash{}
2277 \texttt{ \texttt{-box\_width\textunderscore 40\textunderscore bit\_depth\textunderscore 24\textunderscore partition\textunderscore 3\textunderscore 256\textunderscore 256\textunderscore end\textbackslash{}
2278 \texttt{ $(ORBITER\_PATH)orbiter.out-v.2\textunderscore draw\_matrix\textbackslash{}
2279 \texttt{ \texttt{-input\_csv\_file\textunderscore GF\textunderscore q256\_multiplication\_table\textunderscore csv\textbackslash{}
2280 \texttt{ \texttt{-box\_width\textunderscore 40\textunderscore bit\_depth\textunderscore 24\textunderscore partition\textunderscore 3\textunderscore 255\textunderscore 255\textunderscore end\textbackslash{}
2281 \texttt{ $(ORBITER\_PATH)orbiter.out-v.2\textunderscore draw\_matrix\textbackslash{}
2282 \texttt{ \texttt{-input\_csv\_file\textunderscore GF\textunderscore q256\_addition\_table\textunderscore reordered\textunderscore csv\textbackslash{}
2283 \texttt{ \texttt{-box\_width\textunderscore 40\textunderscore bit\_depth\textunderscore 24\textunderscore partition\textunderscore 3\textunderscore 256\textunderscore 256\textunderscore end\textbackslash{}
2284 \texttt{ $(ORBITER\_PATH)orbiter.out-v.2\textunderscore draw\_matrix\textbackslash{}
2285 \texttt{ \texttt{-input\_csv\_file\textunderscore GF\textunderscore q256\_multiplication\_table\textunderscore reordered\textunderscore csv\textbackslash{}
2286 \texttt{ \texttt{-box\_width\textunderscore 40\textunderscore bit\_depth\textunderscore 24\textunderscore partition\textunderscore 3\textunderscore 255\textunderscore 255\textunderscore end\textbackslash{}
2287 \texttt{ \texttt{pdf\_latex\textunderscore GF\textunderscore 256.pdf\textbackslash{}
2288 \texttt{ open\_GF\textunderscore 256.pdf\textbackslash{}
2289 \texttt{ \texttt{\textbackslash{}
2290 \texttt{ \texttt{\textbackslash{}
2291 \texttt{ trace\textunderscore 256:\n2292 \texttt{ $(ORBITER\_PATH)orbiter.out-v.3\textbackslash{}
2293 \texttt{ \texttt{-define F\textunderscore finite\_field\textunderscore q\textunderscore 256\textunderscore end\textbackslash{}
2294 \texttt{ \texttt{-with F\textunderscore do\textunderscore finite\_field\_activity\textunderscore trace\textunderscore end\textbackslash{}
2295 \texttt{ $(ORBITER\_PATH)orbiter.out-v.3\textbackslash{}
2296 \texttt{ \texttt{-reformat F\textunderscore q256\_trace\textunderscore csv\textunderscore F\textunderscore q256\_trace\textunderscore 16x16\textunderscore csv\textunderscore 16\textbackslash{}
2297 \texttt{ $(ORBITER\_PATH)orbiter.out-v.2\textunderscore draw\_matrix\textbackslash{}
2298 \texttt{ \texttt{-input\_csv\_file\textunderscore F\textunderscore q256\_trace\textunderscore 16x16\textunderscore csv\textbackslash{}
2299 \texttt{ \texttt{\textbackslash{}
\end{verbatim}


```
2300 ▶ ▶ -box_width 40 -bit_depth 24  ▼
2301 ▶ ▶ -partition 4 "1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"  ▼
2302 ▶ ▶ "1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1"-end
2303
2304
2305
2306 F_289:
2307 ▶ ▶ $(ORBITER_PATH) orbiter.out -v 3  ▼
2308 ▶ ▶ -define F -finite_field -q 289 -end  ▼
2309 ▶ ▶ -with F -do -finite_field_activity -cheat_sheet_GF -end
2310 ▶ ▶ pdflatex GF_289.tex
2311 ▶ ▶ open GF_289.pdf
2312
2313
2314 F_512:
2315 ▶ ▶ $(ORBITER_PATH) orbiter.out -v 3  ▼
2316 ▶ ▶ -define F -finite_field -q 512 -end
2317
2318
2319 # User_time 2 100 seconds
2320
2321
2322 F_1024:
2323 ▶ ▶ $(ORBITER_PATH) orbiter.out -v 3  ▼
2324 ▶ ▶ -define F -finite_field -q 1024 -end
2325
2326 # User_time 10 100 seconds
2327
2328 F_2048:
2329 ▶ ▶ $(ORBITER_PATH) orbiter.out -v 3  ▼
2330 ▶ ▶ -define F -finite_field -q 2048 -end
2331
2332 F_4096:
2333 ▶ ▶ $(ORBITER_PATH) orbiter.out -v 3  ▼
2334 ▶ ▶ -define F -finite_field -q 4096 -end
2335
2336
2337 # User_time 0:2
2338
2339 F_8192:
2340 ▶ ▶ $(ORBITER_PATH) orbiter.out -v 3  ▼
2341 ▶ ▶ -define F -finite_field -q 8192 -end
2342
2343
2344 # User_time 0:8
2345
2346 F_16384:
```

481
# Section 3.5: Linear Algebra over Finite Fields
SECTION_LINEAR_ALGEBRA:

RREF:

```
RREF.ORB: $(ORBITER_PATH)orbiter.out-v.2:\
  -define-F-finite_field-q.2-\end\\
```

```
RREF.V7:
RREF.ORB: $(ORBITER_PATH)orbiter.out-v.2:\
  -define-F-finite_field-q.7-\end\\
```

```
nullspace:
nullspace.ORB: $(ORBITER_PATH)orbiter.out-v.2:\
  -define-F2-finite_field-q.2-\end\\
```

```
eigenstuff:
eigenstuff.ORB: $(ORBITER_PATH)orbiter.out-v.6:\
  -define-F-finite_field-q.5-\end\\
```

classes_GL_3_2:
  $(ORBITER_PATH)orbiner.out -v.7:
  -define F=finite_field -q2 -end
  -all_rational_normal_forms F=3
  #pdfflatex-Class_reps_GL3_2.tex
  #open-Class_reps_GL3_2.pdf
  #252:classes

classes_GL_4_2:
  $(ORBITER_PATH)orbiner.out -v.7:
  -define F=finite_field -q2 -end
  -all_rational_normal_forms F=4
  pdflatex-Class_reps_GL4_2.tex
  open-Class_reps_GL4_2.pdf

RREF_demo_4_4_q5:
  $(ORBITER_PATH)orbiner.out -v.2:
  -define F=finite_field -q5 -end
  -with F=do
  -finite_field_activity -RREF demo 4 4 -end
  pdflatex RREF_example_q5_4_4.tex
  #open RREF_example_q5_4_4.pdf
gs -sDEVICE=png16 -dFIXEDMEDIA:
  -dDEVICEWIDTHPOINTS=500 -dDEVICEHEIGHTPOINTS=450:
  -r240 -oRREF_example_q5_4_4_page%02d.png:
  RREF_example_q5_4_4.pdf

RREF demo 4 4_q7:
  $(ORBITER_PATH)orbiner.out -v.2:
  -define F=finite_field -q7 -end
  -with F=do
  -finite_field_activity -RREF random_matrix 4 6 -end
  pdflatex RREF_example_q7_4_6.tex
gs -sDEVICE=png16 -dFIXEDMEDIA:
  -dDEVICEWIDTHPOINTS=500 -dDEVICEHEIGHTPOINTS=450:
  -r240 -oRREF_example_q7_4_6_page%02d.png:
  RREF_example_q7_4_6.pdf
Section 3.6: Advanced Topics in finite fields

SECTION_ADVANCED_TOPICS_INFINITE_FIELDS:

normal_basis_2.3:

RREF demo 4.8.q8:

\[(ORBITER\_PATH) orbiter.out -v 2 \backslash \]
\[-define F -finite_field -q 8 -end \backslash \]
\[-with F -do \backslash \]
\[-finite_field_activity -RREF_random_matrix 4 8 -end \]
\[pdflatex RREF_example_q8_4_8.tex \]
\[#open RREF_example_q8_4_8.pdf \]
\[gs -sDEVICE=png16 -dFIXEDMEDIA \backslash \]
\[-dDEVICEWIDTHPOINTS=500 -dDEVICEHEIGHTPOINTS=450 \backslash \]
\[-r240 -o RREF_example_q8_4_8_page%02d.png \]
\[RREF_example_q8_4_8.pdf \]

RREF_RS_6_4.7:

\[(ORBITER\_PATH) orbiter.out -v 2 \backslash \]
\[-define F -finite_field -q 7 -end \backslash \]
\[-define v -vector -field F -format 4 \backslash \]
\[ -dense $(CODE_RS_6_4_7) \backslash \]
\[-end \backslash \]
\[-with F -do \backslash \]
\[-finite_field_activity -RREF v -end \]
\[pdflatex RREF_example_q7_4_6.tex \]
\[gs -sDEVICE=png16 -dFIXEDMEDIA \backslash \]
\[-dDEVICEWIDTHPOINTS=500 -dDEVICEHEIGHTPOINTS=450 \backslash \]
\[-r240 -o RREF_example_q7_4_6_page%02d.png \]
\[RREF_example_q7_4_6.pdf \]
normal_basis_2,6:

\begin{verbatim}
2535 } } -normal_basis_3-end
2536
2537
2538 "normal_basis_2,6:
2539 } } $(ORBITER_PATH)orbiter.out-v.2\}
2540 } } -define-F:-finite_field-q:2-end\}
2541 } } -with-F:-do:-finite_field_activity\}
2542 } } -normal_basis_6-end
2543
2544
2545
2546 F8_over_F2_field_reduction:
2547 } } $(ORBITER_PATH)orbiter.out-v.2\}
2548 } } -define-F:-finite_field-q:8-end\}
2549 } } -loop-L:0:8:1\}
2550 } } -with-F:-do\}
2551 } } -finite_field_activity\}
2552 } } -field_reduction:"F8_red_%L":2:1:1"%L":\}
2553 } } -end\}
2554 } } -end_loop
2555 } } $(ORBITER_PATH)orbiter.out-v.2-loop-L:0:8:1\}
2556 } } -draw_matrix-input_csv_file:F8_red_%L.csv\}
2557 } } -box_width:40-bit_depth:24-partition:4:3:3:-end\}
2558 } } -end_loop
2559 } } "pdflatex_field_reduction_Q8_q2.5_7.tex"
2560
2561
2562
2563
2564 F_64_over_F8_field_reduction:
2565 } } $(ORBITER_PATH)orbiter.out-v.2\}
2566 } } -define-F:-finite_field-q:64-end\}
2567 } } -define-els:-vector-field-F:-loop-0:64:1-end\}
2568 } } -with-F:-do\}
2569 } } -finite_field_activity-field_reduction:"F64_over_F8":8:8:8\}
2570 } } -els-end
2571 } } $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\}
2572 } } -input_csv_file:F64_over_F8.csv\}
2573 } } -box_width:40-bit_depth:24\}
2574 } } -partition:4:2,2,2,2,2,2:2,2,2,2,2,2:-end
2575 } } open:F64_over_F8_draw.bmp
2576 } } "pdflatex_field_reduction_Q64_q8.8.8.tex"
2577 } } "open_field_reduction_Q64_q8.8.8.pdf"
2578
2579
2580 F_64_over_F4_field_reduction:
2581 } } $(ORBITER_PATH)orbiter.out-v.2\}
\end{verbatim}
define F -finite_field-q 64 -end \ndefine elts -vector-field F-loop 0-64 1 -end \nwith F -do \nfinite_field_activity \n-define field_reduction"F64_over_F4" 4 8 8 elts -end \n$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix \n-input_csv_file F64_over_F4.csv \n-box_width 40 -bit_depth 24 \n-partition 4 "3,3,3,3,3,3,3,3" "3,3,3,3,3,3,3,3" -end \nopen F64_over_F4_draw.bmp \n#pdflatex field_reduction_Q64_q4_8_8.tex \n#open field_reduction_Q64_q4_8_8.pdf

F64_over_F2_field_reduction: \n$ (ORBITER_PATH) orbiter.out -v 2 \ndefine F -finite_field-q 64 -end \ndefine elts -vector-field F-loop 0-64 1 -end \nwith F -do \nfinite_field_activity \n-define field_reduction"F64_over_F2" 2 8 8 elts -end \n$ (ORBITER_PATH) orbiter.out -v 2 -draw_matrix \n-input_csv_file F64_over_F2.csv \n-box_width 40 -bit_depth 24 \n-partition 4 "6,6,6,6,6,6,6,6" "6,6,6,6,6,6,6,6" -end \nopen F64_over_F2_draw.bmp \n#pdflatex field_reduction_Q64_q2_8_8.tex \n#open field_reduction_Q64_q2_8_8.pdf

F_8_Nth_roots_21: \n$ (ORBITER_PATH) orbiter.out -v 3 \ndefine F -finite_field-q 8 -override_polynomial 11 -end \nwith F -do -finite_field_activity -nth_roots_21 -end \npdflatex Nth_roots_q8_n21.tex \nopen Nth_roots_q8_n21.pdf

F_8_vandermonde: \n$ (ORBITER_PATH) orbiter.out -v 3 \ndefine F -finite_field-q 8 -end \nwith F -do -finite_field_activity \n
487
F_1024_vandermonde:
$\text{(ORBITER PATH)\, orbiter.out-v.3}$
-define F -finite_field -q 1024 -end
-define F -finite_field_activity
-define F -finite_field_activity
with F
Berlekamp_matrix_2_3:
$\text{(ORBITER PATH)\, orbiter.out-v.2}$
-define F -finite_field -q 2 -end
-define v -vector -field F -dense "1,1,0,1" -end
-define v -vector -field F -dense "1,1,0,1" -end
-define v -vector -field F -dense "1,1,0,1" -end
with F
Berlekamp_matrix_v -end
Berlekamp_matrix_v -end
#the polynomial $X^3+X+1$ is irreducible over GF(2) because the rank of the Berlekamp matrix is 2.
Berlekamp_matrix_2_4:
$\text{(ORBITER PATH)\, orbiter.out-v.2}$
-define F -finite_field -q 2 -end
-define v -vector -field F -dense "1,1,0,1" -end
-define v -vector -field F -dense "1,1,0,1" -end
with F
Berlekamp_matrix_v -end
Berlekamp_matrix_v -end
#the polynomial $X^4+X+1$ is irreducible over GF(2) because the rank of the Berlekamp matrix is 3.
Berlekamp_matrix_4_3a:
Berklekmatrix_43b:

find_roots_a:

find_roots_b:

find_roots_c:

find_roots_d:
find_roots_e:
- $(\text{ORBITER\ PATH})\text{orbiter.out}\cdot-v.2$
- \text{define}\ F\cdot\text{finite_field}\cdot-q.19\cdot\text{end}$\$
- \text{define}\ v\cdot\text{vector}\cdot\text{field}\cdot F\cdot\text{dense}^1\cdot16\cdot3\cdot\text{end}\$
- \text{with}\ F\cdot\text{do}$\$
- \text{finite_field_activity}$\$
- \text{polynomial}\_\text{find_roots}\cdot v\cdot\text{end}$

roots_over_F2:
- $(\text{ORBITER\ PATH})\text{orbiter.out}\cdot-v.2$
- \text{define}\ F\cdot\text{finite_field}\cdot-q.2\cdot\text{end}$\$
- \text{define}\ v\cdot\text{vector}\cdot\text{field}\cdot F\cdot\text{dense}^0\cdot1\cdot0\cdot1\cdot1\cdot1\cdot\text{end}\$
- \text{with}\ F\cdot\text{do}$\$
- \text{finite_field_activity}$\$
- \text{polynomial}\_\text{find_roots}\cdot v\cdot\text{end}$

roots_over_F8:
- $(\text{ORBITER\ PATH})\text{orbiter.out}\cdot-v.2$
- \text{define}\ F\cdot\text{finite_field}\cdot-q.8\cdot\text{end}$\$
- \text{define}\ v\cdot\text{vector}\cdot\text{field}\cdot F\cdot\text{dense}^0\cdot1\cdot0\cdot1\cdot1\cdot1\cdot\text{end}\$
- \text{with}\ F\cdot\text{do}$\$
- \text{finite_field_activity}$\$
- \text{polynomial}\_\text{find_roots}\cdot v\cdot\text{end}$

# degree and then order of the field of coefficients:

irred_3.2:
- $(\text{ORBITER\ PATH})\text{orbiter.out}\cdot-v.3$
- \text{define}\ F\cdot\text{finite_field}\cdot-q.2\cdot\text{end}$\$
- \text{with}\ F\cdot\text{do}$\$
- \text{finite_field_activity}$\$
- \text{make_table_of_irreducible_polynomials}\cdot 3\cdot\text{end}$
- pdflatex\text{Irred}\_q2\_d3.tex
- open\_Irred\_q2\_d3.pdf

irred_4.2:
- $(\text{ORBITER\ PATH})\text{orbiter.out}\cdot-v.3$
- \text{define}\ F\cdot\text{finite_field}\cdot-q.2\cdot\text{end}$\$
- \text{with}\ F\cdot\text{do}$\$
- \text{finite_field_activity}$\$

490
-make_table_of_irreducible_polynomials 4 - end
pdflatex Irred_q2_d4.tex
open Irred_q2_d4.pdf

# 3 polys

irred_5_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 5 - end
pdflatex Irred_q2_d5.tex
open Irred_q2_d5.pdf

# 6 polys

irred_6_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 6 - end
pdflatex Irred_q2_d6.tex
open Irred_q2_d6.pdf

# 9 polys

irred_7_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 7 - end
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 8 - end
pdflatex Irred_q2_d8.tex
open Irred_q2_d8.pdf
irred_9_2:
$\text{(ORBITER\_PATH)}\text{orbiter.out}\text{-v}\cdot3$
-define F\text{-finite field}\text{-q}\cdot2\text{-end}
-with F\text{-do}
-finite field activity
-make table of irreducible polynomials 9\text{-end}

irred_10_2:
$\text{(ORBITER\_PATH)}\text{orbiter.out}\text{-v}\cdot3$
-define F\text{-finite field}\text{-q}\cdot2\text{-end}
-with F\text{-do}
-finite field activity
-make table of irreducible polynomials 10\text{-end}

irred_2_4:
$\text{(ORBITER\_PATH)}\text{orbiter.out}\text{-v}\cdot3$
-define F\text{-finite field}\text{-q}\cdot4\text{-end}
-with F\text{-do}
-finite field activity
-make table of irreducible polynomials 2\text{-end}

irred_3_4:
$\text{(ORBITER\_PATH)}\text{orbiter.out}\text{-v}\cdot6$
-define F\text{-finite field}\text{-q}\cdot4\text{-end}
-with F\text{-do}
-finite field activity
-make table of irreducible polynomials 3\text{-end}

#-30-polys

#-56-polys

#-99-polys

#-6-polys

#-20-polys
```bash
search_primitive_poly_2:
  $(ORBITER_PATH)orbiter.out-v.3:\
  -search_for_primitive_polynomial_in_range-2^2\*10\#|grep://
```

```
# stuck in factoring 2^61-1 (which is prime)
```

```bash
search_primitive_poly_3:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-3^2\*60
```

```bash
search_primitive_poly_4:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-4^2\*30
```

```bash
search_primitive_poly_5:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-5^2\*30
```

```bash
search_primitive_poly_7:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-7^2\*20
```

```bash
search_primitive_poly_8:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-8^2\*20
```

```bash
search_primitive_poly_9:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-9^2\*15
```

```bash
search_primitive_poly_11:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-11^2\*15
```

```bash
search_primitive_poly_13:
  $(ORBITER_PATH)orbiter.out-v.6:\
  -search_for_primitive_polynomial_in_range-13^2\*15
```
2909  search_primitive_poly_degree_16:
2910  ▷ $(ORBITER_PATH)orbiter.out--v-6:\
2911  ▷ ▷ -search_for_primitive_polynomial_in_range:2:16:16
2912  ▷
2913  search_primitive_poly_32:
2914  ▷ $(ORBITER_PATH)orbiter.out--v-6:\
2915  ▷ ▷ -search_for_primitive_polynomial_in_range:32:32:2:10
2916  ▷
2917
2918
2919
2920
2921  NTT_k4_q17.cpp:
2922  ▷ $(ORBITER_PATH)orbiter.out--v-3:\
2923  ▷ ▷ -define:F--finite_field--q:17--end:\
2924  ▷ ▷ -with:F--do--finite_field_activity--NTT:4:17--end
2925
2926  F_17_NTT_compile:NTT_k4_q17.cpp
2927  ▷ $(MY_CPP)--NTT_k4_q17.cpp$(CPPFLAGS)\n2928  ▷ ▷ $(LIB)$LFLAGS)--oNTT_k4_q17.out
2929  ▷ ./NTT_k4_q17.out
2930
2931  #.ToDo:
2932
2933  FGDTTP:
2934  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_P_k4.csv:20:8
2935  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_F_k4.csv:20:8
2936  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_AAv_k4.csv:20:8
2937  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_G_k4.csv:20:8
2938  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_D_k4.csv:20:8
2939  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_T_k4.csv:20:8
2940  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−nttTv_k4.csv:20:8
2941  ▷
2942  no:
2943  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_P_k4.csv:20:8
2944  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_F_k3.csv:20:8
2945  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_Gr_k3.csv:20:8
2946  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_Dr_k3.csv:20:8
2947  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_Tr_k3.csv:20:8
2948  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−nttPr_k3.csv:20:8
2949  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_G.k3.csv:20:8
2950  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_D.k3.csv:20:8
2951  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_T.k3.csv:20:8
2952  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_P.k3.csv:20:8
2953  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_G.k2.csv:20:8
2954  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_D.k2.csv:20:8
2955  ▷ $(ORBITER_PATH)orbiter.out--v-2:−draw_matrix−ntt_T.k2.csv:20:8
Chapter 4 - Geometry

Section 4.1: Finite-Projective Spaces

SECTION_FINE_PROJECTIVE_SPACES:

PG_3_2_easy:

PG_1_16:

PG_2_4:
PG_2.13:
\$\text{\texttt{(ORBITER\_PATH)orbiter.out\\}}$
\$\text{\texttt{define\_F\_finite\_field\_F\_13\_end}}$
\$\text{\texttt{define\_P\_projective\_space\_2\_F\_end}}$
\$\text{\texttt{with\_P\_do\_projective\_space\_activity}}$
\$\text{\texttt{-cheat\_sheet}}$
\$\text{\texttt{-end}}$
\$\text{\texttt{pdflatex\_PG\_2.13.tex}}$
\$\text{\texttt{open\_PG\_2.13.pdf}}$

PG_2.64:
\$\text{\texttt{(ORBITER\_PATH)orbiter.out\\}}$
\$\text{\texttt{define\_F\_finite\_field\_F\_64\_end}}$
\$\text{\texttt{define\_P\_projective\_space\_2\_F\_end}}$
\$\text{\texttt{with\_P\_do\_projective\_space\_activity}}$
\$\text{\texttt{-cheat\_sheet}}$
\$\text{\texttt{-end}}$
\$\text{\texttt{pdflatex\_PG\_2.64.tex}}$
\$\text{\texttt{open\_PG\_2.64.pdf}}$

PG_3.2:
\$\text{\texttt{(ORBITER\_PATH)orbiter.out\_v\_0}}$
\$\text{\texttt{define\_F\_finite\_field\_F\_2\_end}}$
\$\text{\texttt{define\_P\_projective\_space\_3\_F\_end}}$
\$\text{\texttt{with\_P\_do\_projective\_space\_activity}}$
\$\text{\texttt{-cheat\_sheet}}$
\$\text{\texttt{-end}}$
\$\text{\texttt{pdflatex\_PG\_3.2.tex}}$
\$\text{\texttt{open\_PG\_3.2.pdf}}$

PG_3.4:
\$\text{\texttt{(ORBITER\_PATH)orbiter.out\_v\_10}}$

496
\begin{center}
\begin{verbatim}
3050   ▶ ▶  -define-F\-finite\_field\-q\-4\-end\
3051   ▶ ▶  -define-P\-projective\_space\-3\-F\-end\
3052   ▶ ▶  -with-P\-do\-projective\_space\_activity\
3053   ▶ ▶ ▶  -cheat\_sheet\
3054   ▶ ▶  -end
3055  ▶ pdflatex\-PG\_3\_4\.tex
3056  ▶ open\-PG\_3\_4.pdf
3057
3058  PG\_3\_5:
3059  ▶  $(\text{ORBITER\_PATH})\text{orbiter}\.out\
3060  ▶ ▶  -define-F\-finite\_field\-q\-5\-end\
3061  ▶ ▶  -define-P\-projective\_space\-3\-F\-end\
3062  ▶ ▶  -with-P\-do\-projective\_space\_activity\
3063  ▶ ▶ ▶  -cheat\_sheet\
3064  ▶ ▶  -end
3065  ▶ pdflatex\-PG\_3\_5\.tex
3066  ▶ open\-PG\_3\_5.pdf
3067
3068  PG\_3\_7:
3069  ▶  $(\text{ORBITER\_PATH})\text{orbiter}\.out\
3070  ▶ ▶  -define-F\-finite\_field\-q\-7\-end\
3071  ▶ ▶  -define-P\-projective\_space\-3\-F\-end\
3072  ▶ ▶  -with-P\-do\-projective\_space\_activity\
3073  ▶ ▶ ▶  -cheat\_sheet\
3074  ▶ ▶  -end
3075  ▶ pdflatex\-PG\_3\_7\.tex
3076  ▶ open\-PG\_3\_7.pdf
3077
3078
3079
3080
3081  PG\_3\_8:
3082  ▶  $(\text{ORBITER\_PATH})\text{orbiter}\.out\
3083  ▶ ▶  -define-F\-finite\_field\-q\-8\-end\
3084  ▶ ▶  -define-P\-projective\_space\-3\-F\-end\
3085  ▶ ▶  -with-P\-do\-projective\_space\_activity\
3086  ▶ ▶ ▶  -cheat\_sheet\
3087  ▶ ▶  -end
3088  ▶ pdflatex\-PG\_3\_8\.tex
3089  ▶ open\-PG\_3\_8.pdf
3090
3091
3092
3093  PG\_3\_16:
3094  ▶  $(\text{ORBITER\_PATH})\text{orbiter}\.out\
3095  ▶ ▶  -define-F\-finite\_field\-q\-16\-end\
3096  ▶ ▶  -define-P\-projective\_space\-3\-F\-end\
3097
497
\end{verbatim}
\end{center}
\begin{verbatim}
3097 \$ \$ -with-P-do-projective_space_activity_\\ 
3098 \$ \$ -cheat_sheet_\\ 
3099 \$ \$ -end \\
3100 pdflatex PG_3_16.tex \\
3101 open-PG_3_16.pdf \\
3102 \\
3103 \\
3104 \\
3105 \\
3106 PG_3_25: \\
3107 \$ (ORBITER_PATH) orbiter.out_\\ 
3108 \$ -define-F-finite_field-q-25-end_\\ 
3109 \$ -define-P-projective_space-3-F-end_\\ 
3110 \$ -with-P-do-projective_space_activity_\\ 
3111 \$ -cheat_sheet_\\ 
3112 \$ -end \\
3113 pdflatex PG_3_25.tex \\
3114 open-PG_3_25.pdf \\
3115 \\
3116 \\
3117 \\
3118 \\
3119 PG_4_3: \\
3120 \$ (ORBITER_PATH) orbiter.out_\\ 
3121 \$ -define-F-finite_field-q-3-end_\\ 
3122 \$ -define-P-projective_space-4-F-end_\\ 
3123 \$ -with-P-do-projective_space_activity_\\ 
3124 \$ -cheat_sheet_\\ 
3125 \$ -end \\
3126 pdflatex PG_4_3.tex \\
3127 open-PG_4_3.pdf \\
3128 \\
3129 \\
3130 PG_8_2: \\
3131 \$ (ORBITER_PATH) orbiter.out_\\ 
3132 \$ -define-F-finite_field-q-2-end_\\ 
3133 \$ -define-P-projective_space-8-F-end_\\ 
3134 \$ -with-P-do-projective_space_activity_\\ 
3135 \$ -cheat_sheet_\\ 
3136 \$ -end \\
3137 pdflatex PG_8_2.tex \\
3138 open-PG_8_2.pdf \\
3139 \\
3140 \\
3141 \\
3142 \\
3143 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
\end{verbatim}
# Section 4.2: Indexing Points

SECTION_INDEXING_POINTS:

PG_2.4_rank_point:

```plaintext
> $(ORBITER_PATH)orbiter.out -v 2: \
> -define F -finite_field -q 4 -end: \
> -with F -do -finite_field_activity: \
> -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:

```plaintext
> $(ORBITER_PATH)orbiter.out -v 2: \
> -define F -finite_field -q 11 -end: \
> -define P -projective_space 2 F -end: \
> -define EC -geometric_object P: \
> -elliptic_curve 1 3: \
> -end: \
> -with EC -do -combinatorial_object_activity - save: \
> -end
```

PG_2.2_incidence_matrix:

```plaintext
> $(ORBITER_PATH)orbiter.out -v 2: \
> -define F -finite_field -q 2 -end: \
> -define P -projective_space 2 F -end: \
> -with P -do -projective_space_activity: \
> -export_point_line_incidence_matrix: \
> -end \
> $(ORBITER_PATH)orbiter.out -v 2: \
> -define all_one -vector -repeat 1 7 -end: \
> -draw_matrix: \
> -input_csv_file PG n2 q2 incidence_matrix.csv: \
> -box_width 20 -bit_depth 8: \
> -partition 3: \
> -partition 3: \
> -all_one -all_one: \
> -end \
> open PG n2 q2 incidence_matrix draw.bmp
```

PG_2.4_incidence_matrix:

```plaintext
> $(ORBITER_PATH)orbiter.out -v 2: \
```
Section 4.3: Finite-Desarguesian Projective Planes

SECTION_FINITE_DESARGUESIAN_PROJECTIVE_PLANES:
PG_2.16:
\$\$(\text{ORNITR_PATH})\text{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{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{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{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{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{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{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{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{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{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{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{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{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{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{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{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{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}\$
3285 \> open-PG_2_4_singer_incma_cyclic_draw.bmp
3286
3287 PG_2_4_incma_singer_sub_3:
3289 \> $(\text{ORBITER\ PATH})\text{orbiter.out}\,-\text{-v\ 4}\,
3290 \> \> -\text{list\ arguments}\,
3291 \> \> \> -\text{define\ R\ -\ vector\ -\ repeat\ 3\ -\ 7\ -\ end}\,
3292 \> \> \> -\text{define\ C\ -\ vector\ -\ repeat\ 3\ -\ 7\ -\ end}\,
3293 \> \> \> -\text{draw\ matrix}\,
3294 \> \> \> -\text{input\ csv\ file\ PG_2_4_singer_incma_subgroup_index_3.csv}\,
3295 \> \> \> -\text{box\ width\ 40\ -\ bit\ depth\ 24}\,
3296 \> \> \> -\text{partition\ 3\ -\ R\ -\ C}\,
3297 \> \> \> -\text{end}
3298 \> \> open-PG_2_4_singer_incma_subgroup_index_3.draw.bmp
3299
3300 PG_2_4_incma_singer_sub_7:
3301 \> $(\text{ORBITER\ PATH})\text{orbiter.out}\,-\text{-v\ 2\ -\ draw\ matrix}\,
3302 \> \> -\text{input\ csv\ file\ PG_2_4_singer_incma_subgroup_index_7.csv}\,
3303 \> \> \> -\text{box\ width\ 20\ -\ bit\ depth\ 24}\,
3304 \> \> \> -\text{partition\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ 3\ -\ end}\,
3305 \> \> \> open-PG_2_4_singer_incma_subgroup_index_7.draw.bmp
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316 ####################################################################################
3317 # Section 4.4: The Grassmannian
3318
3320 SECTION_GRASSMANNIAN:
3321
3322 GR_3_2_2:
3323 \> $(\text{ORBITER\ PATH})\text{orbiter.out}\,
3324 \> \> \> -\text{define\ F\ -\ finite\ field\ -\ q\ 2\ -\ end}\,
3325 \> \> \> -\text{with\ F\ -\ do\ -\ finite\ field\ activity}\,
3326 \> \> \> \> -\text{cheat\ sheet\ Gr\ 3\ -\ 2}\ -\ end
3327 \> \> pdf\ latex\ Gr_3_2_2.tex.
3328 \> \> open-Gr_3_2_2.pdf
3329
3330
3331

502
rank_lines:

$\text{define v1-vector-format 3 dense:}
\begin{array}{c}
1,0,2,2,0,1,1,2,1,0,2,0,1,1,2,1,0,2,0,1,2,1
\end{array}$

$\text{-end}$

$\text{define v2-vector-format 3 dense:}
\begin{array}{c}
1,0,0,0,0,1,0,0,1,0,0,0,0,1,0,1,0,0,0,2,1
\end{array}$

$\text{-end}$

$\text{define F finite field q 3 -end}$

$\text{define P projective space 3 F -end}$

$\text{define EC geometric object P}$

$\text{-projective variety }"\text{EC_11".EC_11" EC_11 EQUATION}"

$\text{-monomial type PART}$

$\text{-end}$

$\text{with EC do -combinatorial_object_activity -save}$

$\text{-end}$

$\text{Hirschfeld_surface_q4.txt:}$

$\text{define F finite field q 4 -end}$

$\text{define P projective space 3 F -end}$
define H4 geometric object P;
-define projective variety "Hirschfeld_surface_q4";
"Hirschfeld\_surface\_q4";
-define H4 geometric object P;
-define H16 geometric object P;
"Hirschfeld\_surface\_q16";
("HIRSCHFELD\_SURFACE\_EQUATION)\;
-define monomial type PART\;
-end\;
-define with H4 do -combinatorial object activity -save\;
-end\;
# creates Hirschfeld_surface_q4.txt

Hirschfeld_surface_q16.txt:
$(ORBITER\_PATH) orbiter.out -v 2\;
-define F finite field q16 -end\;
-define P projective space 3 F -end\;
-define H16 geometric object P\;
"Hirschfeld\_surface\_q16";
("HIRSCHFELD\_SURFACE\_EQUATION)\;
-define monomial type PART\;
-end\;
-define with H16 do -combinatorial object activity -save\;
-end\;

# the coefficient vector is given as a list of pairs.
165 = \text{binomial}(11,3)

Endrass_F7.txt:
$(ORBITER\_PATH) orbiter.out -v 2\;
-define F finite field q7 -end\;
-define eqn vector field F sparse 165\;
-define Endrass_F7 geometric object P\;
"Endrass\_F7";
-define projective variety "Endrass\_F7"\;
"Endrass\_F7";
-define 8 eqn\;
-define monomial type PART\;
-end\;
-define with Endrass_F7 do\;
-define with combinatorial object activity -save\;
-end

504
we created a set of 33 points, called Endrass_F7.txt

octic_prepare:
\>$\{(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot4\}\\
\>$ -\text{define}\ A -\text{vector\_dense\_"1,1,1"\_end}\\
\>$ -\text{define}\ D -\text{diophant}\\
\>$ -\text{label\_octic\_monomials}\\
\>$ -\text{coefficient\_matrix\_A}\\
\>$ -\text{RHS\_"8,8,1"}\\
\>$ -\text{x\_min\_global\_0\_x\_max\_global\_8}\\
\>$ -\text{end}\\
\>$ -\text{with\_D\_do}\\
\>$ -\text{diophant\_activity\_solve\_mckay}\\
\>$ -\text{end}\\
\>$ -\text{sort\_r\_octic\_monomials\_sol\_octic\_monomials\_sorted.txt}\\

#Found 165 solutions with 210 backtrack steps
# 165 = \text{binomial}(11,3)

\\
\#Section 4.6: The Klein Quadric and Pluecker coordinates

SECTION_KLEIN_QUADRIC_AND_PLUECKER_COORDINATES:

\text{GR}_4_{2,2}: \\
\>$\{(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot2\}\\
\>$ -\text{define}\ F -\text{finite\_field\_q\_2\_end}\\
\>$ -\text{with}\ F -\text{do\_finite\_field\_activity}\\
\>$ -\text{cheat\_sheet\_Gr\_4\_2\_end}\\
\>$ \text{pdflatex\_Gr\_4\_2\_2.tex}\\
\>$ \text{open\_Gr\_4\_2\_2.pdf}\\

\#Section 4.7: Orthogonal spaces

SECTION_ORTHOGONAL_SPACES:
Op. 4.2:

\[ ORBITER\ PATH\] orbiter.out -v.2

\[ -define\ F::finite_field\ q=2\ -end\]
\[ -define\ 0::orthogonal\_space\ 1.4\ -without\_group\ -end\]
\[ -with\:0::do::orthogonal\_space\_activity\]
\[ -cheat\_sheet\_orthogonal\ -end\]

pdflatex 0.1.4.2_report.tex

open 0.1.4.2_report.pdf

Op. 6.2:

\[ ORBITER\ PATH\] orbiter.out -v.2

\[ -define\ F::finite_field\ q=2\ -end\]
\[ -define\ 0::orthogonal\_space\ 1.6\ -without\_group\ -end\]
\[ -with\:0::do::orthogonal\_space\_activity\]
\[ -cheat\_sheet\_orthogonal\ -end\]

pdflatex 0.1.6.2_report.tex

open 0.1.6.2_report.pdf
\begin{verbatim}
3518 ▶ ▶ -define::orthogonal_space::1-6:F::without_group::end\
3519 ▶ ▶ -with::do::orthogonal_space::activity\
3520 ▶ ▶ ▶ -export::point_line_incidence_matrix\
3521 ▶ ▶ -end
3522 ▶ $(ORBITER PATH)orbiter.out::v::2\
3523 ▶ ▶ -define::all_one_r::vector::repeat::135::end\
3524 ▶ ▶ -define::all_one_c::vector::repeat::1105::end\
3525 ▶ ▶ -draw_matrix\
3526 ▶ ▶ ▶ -input::csv_file::0p_6_2_incidence_matrix.csv\
3527 ▶ ▶ ▶ -box_width::20::bit_depth::8\
3528 ▶ ▶ ▶ -partition::2\
3529 ▶ ▶ ▶ ▶ all_one_r::all_one_c\
3530 ▶ ▶ -end
3531 ▶ open::0p_6_2_incidence_matrix::draw.bmp
3532
3533 0p_6_2::with_group:
3534 ▶ $(ORBITER PATH)orbiter.out::v::2\
3535 ▶ ▶ -define::F::finite_field::q::2::end\
3536 ▶ ▶ -define::0::orthogonal_space::1-6:F::end\
3537 ▶ ▶ -with::0::do::orthogonal_space::activity\
3538 ▶ ▶ ▶ -cheat_sheet::orthogonal::end
3539 ▶ pdflatex::0_1_6_2_report.tex\
3540 ▶ open::0_1_6_2_report.pdf
3541 ▶ #problem:
3542 ▶ #error::message:
3543 #stabilizer::chain::base::data::allocate::base::data::degree::is::too::large
3544
3545 0p_8_2:
3546 ▶ $(ORBITER PATH)orbiter.out::v::2\
3547 ▶ ▶ -define::F::finite_field::q::2::end\
3548 ▶ ▶ -define::0::orthogonal_space::1-8:F::end\
3549 ▶ ▶ -with::0::do::orthogonal_space::activity\
3550 ▶ ▶ ▶ -cheat_sheet::orthogonal::end
3551 ▶ pdflatex::0_1_8_2_report.tex\
3552 ▶ open::0_1_8_2_report.pdf
3553
3554 0p_6_64:
3555 ▶ $(ORBITER PATH)orbiter.out::v::2\
3556 ▶ ▶ -define::F::finite_field::q::64::end\
3557 ▶ ▶ -define::0::orthogonal_space::1-64:F::end\
3558 ▶ ▶ -with::0::do::orthogonal_space::activity\
3559 ▶ ▶ ▶ -cheat_sheet::orthogonal::end
3560 ▶ pdflatex::0_1_64_report.tex\
3561
3562 \end{verbatim}
#problem, because we are trying to create PGL(6,64):

Op_6_64_line_rank_problem:

```
-define F = finite_field q=64 -end
-define O = orthogonal_space 1 6 F -end
-with 0 do orthogonal_space_activity
-unrank_line_through_two_points 15447347 15225451
-end
```

#use option -without_group to skip the group. This will work:

Op_6_64_line_rank:

```
-define F = finite_field q=64 -end
-define O = orthogonal_space 1 6 F -without_group -end
-with 0 do orthogonal_space_activity
-unrank_line_through_two_points 15447347 15225451
-end
```

#this will create a basic report without the group:

Op_6_64_report:

```
-define F = finite_field q=64 -end
-define O = orthogonal_space 1 6 F -without_group -end
-with 0 do orthogonal_space_activity
-cheat_sheet_orthogonal
-end
  pdflatex 0_1_6_64_report.tex
  open 0_1_6_64_report.pdf
```

Op_6_8_2:

```
-define F = finite_field q=8 -end
-define O = orthogonal_space 1 6 F -without_group -end
-with 0 do orthogonal_space_activity
-cheat_sheet_orthogonal
-end
  pdflatex 0_1_6_8_report.tex
  open 0_1_6_8_report.pdf
```
4.8: Hermitian varieties

H_2_4:
$\text{(ORBITER_PATH)} orbiter.out -v 2$
-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

H_2_9:
$\text{(ORBITER_PATH)} orbiter.out -v 2$
-define F finite_field -q 9 end
-with F do finite_field activity
-cheat sheet hermitian 2 end
pdflatex H_2_9.tex
open H_2_9.pdf

#28 points: 6, 11, 9, 7, 14, 19, 17, 15, 80, 75, 78, 74, 35, 30, 33, 29, 62, 57, 60, 56, 26, 21, 24, 3, 37, 82, 64, 46

H_3_4:
$\text{(ORBITER_PATH)} orbiter.out -v 2$
-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

#H_3_4 = the Hirschfeld surface

#Section 4.9: Projective Space Advanced Topics

SECTION_PROJECTIVE_SPACE_ADVANCED_TOPICS:
fix_structure_2A:
  $(ORBITER\_PATH)\texttt{orbiter.out} -v.2\backslash
  -define F-\textit{finite\_field}\ -q.4\ -end\backslash
  -define P-\textit{projective\_space}\ 3\ F\ -end\backslash
  -with P\ -do\backslash
  -projective\_space\_activity\backslash
  -cheat\_sheet\_for\_decomposition\_by\_element PG\ 1\backslash
  "1,0,0,0,\cdot,1,0,0,\cdot,0,1,0,0,0,1,\cdot,1"\backslash
  fix\_structure\_2A\backslash
  -end\backslash
  pdflatex fix\_structure\_2A.tex\backslash
  open fix\_structure\_2A.pdf

fix_structure_2B:
  $(ORBITER\_PATH)\texttt{orbiter.out} -v.2\backslash
  -define F-\textit{finite\_field}\ -q.4\ -end\backslash
  -define P-\textit{projective\_space}\ 3\ F\ -end\backslash
  -with P\ -do\backslash
  -projective\_space\_activity\backslash
  -cheat\_sheet\_for\_decomposition\_by\_element PG\ 1\backslash
  "1,0,0,0,\cdot,1,1,0,0,\cdot,0,1,0,\cdot,0,0,0,1,\cdot,0"\backslash
  fix\_structure\_2B\backslash
  -end\backslash
  pdflatex fix\_structure\_2B.tex\backslash
  open fix\_structure\_2B.pdf

fix_structure_2C:
  $(ORBITER\_PATH)\texttt{orbiter.out} -v.2\backslash
  -define F-\textit{finite\_field}\ -q.4\ -end\backslash
  -define P-\textit{projective\_space}\ 3\ F\ -end\backslash
  -with P\ -do\backslash
  -projective\_space\_activity\backslash
  -cheat\_sheet\_for\_decomposition\_by\_element PG\ 1\backslash
  "1,0,0,0,\cdot,1,1,0,0,\cdot,0,1,0,\cdot,0,0,1,1,\cdot,0"\backslash
  fix\_structure\_2C\backslash
  -end\backslash
  pdflatex fix\_structure\_2C.tex\backslash
  open fix\_structure\_2C.pdf

trans:
\texttt{\$(ORBITER\_PATH)\,orbiter.out\,-v\,3:\}\texttt{\$
\texttt{-define\,F\,-finite\_field\,-q\,13\,-end}\}\texttt{\$
\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\}\texttt{-with\,P\,-do\}\texttt{-projective\_space\,activity\,}\texttt{-move\,two\_lines\,in\,hyperplane\,stabilizer\,text\,}\texttt{"1,0,0,0,0,0,1\,\",\,"1,1,0,2,0,0,1,0\,\"}\}\texttt{-define\,P\,-projective\_space\,3,\,F}\texttt{-end}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13ab\,report:\}\texttt{-define\,F\,-finite\_field\,-q\,16\,-end}\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\texttt{-with\,P\,-do\}\texttt{-projective\_space\,activity\,}\texttt{-move\,two\,lines\,in\,hyperplane\,stabilizer\,text\,}\texttt{"1,0,0,0,0,0,1\,\",\,"0,1,0,1,0,0,1,0\,\"}\texttt{-end}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13ab\,report:\}\texttt{\$
\texttt{-define\,F\,-finite\_field\,-q\,13\,-end}\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\texttt{-define\,f3\,-formula\,\"del\,Pezzo\,F13a\,\",\,\"x,y,z\,\"\}\texttt{-define\,f4\,-formula\,\"del\,Pezzo\,F13b\,\",\,\"x,y,z\,\"\}\texttt{-define\,del\,Pezzo13\,-collection\,\"f3,f4\,\"\}\texttt{\$\,del\,Pezzo13\,report:\}\texttt{-analyze\,del\,Pezzo\,surface\,del\,Pezzo13\,\"\,}\texttt{\$\,del\,Pezzo13\,report\,pdf}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13a\,points\,txt:\}\texttt{-define\,F\,-finite\_field\,-q\,13\,-end}\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\texttt{-define\,f1\,-formula\,\"del\,Pezzo\,F9\,\",\,\"x,y,z\,\"\}\texttt{-define\,f2\,-formula\,\"del\,Pezzo\,F11\,\",\,\"x,y,z\,\"\}\texttt{-define\,f3\,-formula\,\"del\,Pezzo\,F13a\,\",\,\"x,y,z\,\"\}\texttt{-define\,f4\,-formula\,\"del\,Pezzo\,F13b\,\",\,\"x,y,z\,\"\}\texttt{-with\,P\,-do\}\texttt{-projective\_space\,activity\,}\texttt{-move\,two\,lines\,in\,hyperplane\,stabilizer\,text\,}\texttt{"1,0,0,0,0,0,1\,\",\,"0,1,0,1,0,0,1,0\,\"}\texttt{-end}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13a\,points\,txt:\}\texttt{-define\,F\,-finite\_field\,-q\,16\,-end}\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\texttt{-define\,f1\,-formula\,\"del\,Pezzo\,F9\,\",\,\"x,y,z\,\"\}\texttt{-define\,f2\,-formula\,\"del\,Pezzo\,F11\,\",\,\"x,y,z\,\"\}\texttt{-define\,f3\,-formula\,\"del\,Pezzo\,F13a\,\",\,\"x,y,z\,\"\}\texttt{-define\,f4\,-formula\,\"del\,Pezzo\,F13b\,\",\,\"x,y,z\,\"\}\texttt{-with\,P\,-do\}\texttt{-projective\_space\,activity\,}\texttt{-move\,two\,lines\,in\,hyperplane\,stabilizer\,text\,}\texttt{"1,0,0,0,0,0,1\,\",\,"0,1,0,1,0,0,1,0\,\"}\texttt{-end}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13a\,points\,txt:\}\texttt{-define\,F\,-finite\_field\,-q\,13\,-end}\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\texttt{-define\,f1\,-formula\,\"del\,Pezzo\,F9\,\",\,\"x,y,z\,\"\}\texttt{-define\,f2\,-formula\,\"del\,Pezzo\,F11\,\",\,\"x,y,z\,\"\}\texttt{-define\,f3\,-formula\,\"del\,Pezzo\,F13a\,\",\,\"x,y,z\,\"\}\texttt{-define\,f4\,-formula\,\"del\,Pezzo\,F13b\,\",\,\"x,y,z\,\"\}\texttt{-with\,P\,-do\}\texttt{-projective\_space\,activity\,}\texttt{-move\,two\,lines\,in\,hyperplane\,stabilizer\,text\,}\texttt{"1,0,0,0,0,0,1\,\",\,"0,1,0,1,0,0,1,0\,\"}\texttt{-end}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13a\,points\,txt:\}\texttt{-define\,F\,-finite\_field\,-q\,16\,-end}\texttt{-define\,P\,-projective\_space\,-3\,-F\,-end}\texttt{-define\,f1\,-formula\,\"del\,Pezzo\,F9\,\",\,\"x,y,z\,\"\}\texttt{-define\,f2\,-formula\,\"del\,Pezzo\,F11\,\",\,\"x,y,z\,\"\}\texttt{-define\,f3\,-formula\,\"del\,Pezzo\,F13a\,\",\,\"x,y,z\,\"\}\texttt{-define\,f4\,-formula\,\"del\,Pezzo\,F13b\,\",\,\"x,y,z\,\"\}\texttt{-with\,P\,-do\}\texttt{-projective\_space\,activity\,}\texttt{-move\,two\,lines\,in\,hyperplane\,stabilizer\,text\,}\texttt{"1,0,0,0,0,0,1\,\",\,"0,1,0,1,0,0,1,0\,\"}\texttt{-end}\texttt{\$
\texttt{\$\,del\,Pezzo\,F13a\,points\,txt:\}
-define del_Pezzo9 -collection:"f1".\ 
-define del_Pezzo11 -collection:"f2".\ 
-define del_Pezzo13 -collection:"f3,f4".\ 
-with P -do\ 
-projective_space_activity\ 
-analyze del_Pezzo_surface del_Pezzo13.".\ 
-end\ 
pdflatex del_Pezzo_F169a_report.tex\ 
pdflatex del_Pezzo_F169b_report.tex\ 
open del_Pezzo_F169a_report.pdf\ 
open del_Pezzo_F169b_report.pdf\ 
-dot -Tpng del_Pezzo_F169a.gv > del_Pezzo_F169a.png\ 
-dot -Tpng del_Pezzo_F169b.gv > del_Pezzo_F169b.png\ 
-writes del_Pezzo_F169a_points.txt

Section 4.10: Geometric Objects

SECTION GEOMETRIC OBJECTS:
elliptic_quadric_0void_q8:
$(ORBITER\_PATH)orbiter.out-v.2$
-define-F-finite_field-q.8-end-
-define-P-projective_space-3-F-end-
-define-O-geometric_object-P-
-elliptic_quadric_0void-
-with-O-do-combinatorial\_object\_activity-save-
-end-

#ovoid.q8.txt
#-65-points

ovoid_ST.q8:
$(ORBITER\_PATH)orbiter.out-v.2$
-define-F-finite_field-q.8-end-
-define-P-projective_space-3-F-end-
-define-O-geometric_object-P-
-ovoid_ST-
-end-

#ovoid_ST.q8.txt

Edge_curve_17:
$(ORBITER\_PATH)orbiter.out-v.2$
-define-F-finite_field-q.17-end-
-define-P-projective_space-2-F-end-
-define-C-geometric_object-P-
-projective\_variety."Edge_17":"Edge\_q17"-
-monomial\_type\_PART-
-end-

#Edge_q17.txt
#combinatorial\_object.create::init-created-a-set-of-size\_12
#(4,7,16,19,20,23,32,35,89,100,244,251)
Edge_curve_17_line_type:
  echo $(FILE_Q17) > edge_q17.csv
  $(ORBITER_PATH)orbiter.out -v 2
  -define F -finite_field -q 17 -end
  -define P -projective_space 2 -F -end
  -define C -geometric_object P
  -projective_variety "Edge_q17" "Edge\_q17"
  -monomial_typePART
  -end
  -with C -do
  -combinatorial_object_activity
  -line_type
  -end
  -print_symbols

Edge_curve_23_line_type:
  $(ORBITER_PATH)orbiter.out -v 2
  -define F -finite_field -q 23 -end
  -define P -projective_space 2 -F -end
  -define E -geometric_object P
  -set $(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
# Chapter 5: Group Theory

## Section 5.1: Permutation Groups

```plaintext
C13:
$\text{(ORBITER_PATH)}$orbiter.out\-v\-10\-
$\text{-define-gens}\-\text{-vector}\-\text{-dense}\$(\text{GEN_C13})\-\text{-end}\-
$\text{-define-G}\-\text{-permutation}\text{-group}\-
$\text{-bsgs-C13-C}_{13}\cdot13\cdot13\cdot 1\cdot 1\-
$\text{-define}\-\text{-end}\-
$\text{-with-G}\-\text{-do}\-
$\text{-group}\text{-theoretic}\text{-activity}\-
$\text{-export}\text{-orbiter}\-
$\text{-end}\-
$\text{-with-G}\-\text{-do}\-
$\text{-group}\text{-theoretic}\text{-activity}\-
$\text{-report}\-
$\text{-end}\-
$\text{-with-G}\-\text{-do}\-
$\text{-group}\text{-theoretic}\text{-activity}\-
$\text{-save}\text{-elements}\text{-csv}\"C13\_elts.csv\"\-
$\text{-end}\-
$\text{pdflatex}\text{-C13}\_report.tex\-
$\text{open}\text{-C13}\_report.pdf\-

Symmetric_4:
$\text{(ORBITER_PATH)}$orbiter.out\-v\-10\-
$\text{-define-G}\-\text{-permutation}\text{-group}\-\text{-symmetric}\text{-group}\_4\-\text{-end}\-
$\text{-with-G}\-\text{-do}\-
$\text{-group}\text{-theoretic}\text{-activity}\-
$\text{-export}\text{-orbiter}\-
$\text{-end}\-
$\text{-with-G}\-\text{-do}\-
$\text{-group}\text{-theoretic}\text{-activity}\-
$\text{-save}\text{-elements}\text{-csv}\"C13\_elts.csv\"\-
$\text{-end}\-
$\text{pdflatex}\text{-C13}\_report.tex\-
$\text{open}\text{-C13}\_report.pdf\-
```

```


# ToDo:

```
Symmetric_4_stab:

$($ORBITER_PATH)orbiter.out -v 2.0$

-define-G-permutation_group -symmetric_group 4 -end

-with-G-do

-group_theoretic_activity

-orbits_on_points

-stabilizer_of_orbit_rep 0

-end

$($ORBITER_PATH)orbiter.out -v 2$

-define-gens-vector-file Perm4_stab_orb_0_gens.csv -end

-define-G-permutation_group

-bsgs Perm4_stab_orb 0 "Sym3" 4 6 "0,1,2" 2 gens -end

-define-Gr-modified_group-from-G

-restricted_action "1,2,3"

-end

-with-Gr-do

-group_theoretic_activity

-report

-end

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
```
C13 as subgroup:

```bash
$ (ORBITER_PATH) orbiter.out -v -10 \
define G - permutation_group - symmetric_group 13 \
define G - linear_group - PGL 3 F \
with G - do \
subgroup_by_generators C13 13 1 $(GEN C13) - end \
group_theoretic_activity \
end \
egroup_theoretic_activity \
end \
group_theoretic_activity \
end \
group_theoretic_activity \
end \
save_elements_csv "C13 elts.csv" \
end 
```

**SECTION LINEARGROUPS:**

```bash
PGL 3 2:
$ (ORBITER_PATH) orbiter.out -v 2 \
define F - finite_field - q 2 - end \
define G - linear_group - PGL 3 F - end \
with G - do \
group_theoretic_activity \
end 
```

```bash
PGL 4 2:
```
\begin{verbatim}
0434 ▷ $(ORBITER_PATH)orbiter.out-v.2:\n0435 ▷ ▷ -define F: finite_field-q 2-end:\n0436 ▷ ▷ ▷ -define G: linear_group-PGL.4-F-end:\n0437 ▷ ▷ ▷ -with G: do:\n0438 ▷ ▷ ▷ -group_theoretic_activity:\n0439 ▷ ▷ ▷ ▷ -report:\n0440 ▷ ▷ ▷ -end
0441 ▷ pdflatex PGL.4.2_report.tex
0442 ▷ open PGL.4.2_report.pdf
0443
0444
0445
0446
0447 AGL.1.27:
0448 ▷ $(ORBITER_PATH)orbiter.out-v.2:\n0449 ▷ ▷ -define F: finite_field-q 27-end:\n0450 ▷ ▷ ▷ -define G: linear_group-AGL.1-F-end:\n0451 ▷ ▷ ▷ -with G: do:\n0452 ▷ ▷ ▷ -group_theoretic_activity:\n0453 ▷ ▷ ▷ ▷ -report:\n0454 ▷ ▷ ▷ -end
0455 ▷ pdflatex AGL.1.27_report.tex
0456 ▷ open AGL.1.27_report.pdf
0457
0458 #▷ ▷ -group_table:\n0459
0460
0461
0462 PGL.4.5:
0463 ▷ $(ORBITER_PATH)orbiter.out-v.2:\n0464 ▷ ▷ -define F: finite_field-q 5-end:\n0465 ▷ ▷ ▷ -define G: linear_group-PGL.4-F-end:\n0466 ▷ ▷ ▷ -with G: do:\n0467 ▷ ▷ ▷ -group_theoretic_activity:\n0468 ▷ ▷ ▷ ▷ -report:\n0469 ▷ ▷ ▷ -end
0470 ▷ pdflatex PGL.4.5_report.tex
0471 ▷ open PGL.4.5_report.pdf
0472
0473
0474
0475
0476
0477 PGL.4.2 wd:
0478 ▷ $(ORBITER_PATH)orbiter.out-v.12:\n0479 ▷ ▷ -define G: linear_group-PGL.4.2-wedge_detached-end:\n0480 ▷ ▷ -with G: do:\n\end{verbatim}
\begin{verbatim}
4081  ▶  ▶  -group_theoretic_activity\`
4082  ▶  ▶  ▶  -report\`
4083  ▶  ▶  -end
4084  ▶  pdflatex PGL_4_2_Wedge_4_0_detached_report.tex
4085  ▶  open-PGL_4_2_Wedge_4_0_detached_report.pdf
4086
4087
4088
4089
4090  PGL_4_2_wd_reverse:
4091  ▶  $(ORBITER\_PATH)orbiter.out-v-12\`
4092  ▶  ▶  -linear_group-PGL_4_2-wedge_detached-end\`
4093  ▶  ▶  -group_theoretic_activity\`
4094  ▶  ▶  ▶  -reverse_isomorphism_exterior_square\`
4095  ▶  ▶  -end
4096
4097  ▶
4098  PGGL_3_4:
4099  ▶  $(ORBITER\_PATH)orbiter.out-v-3\`
4100  ▶  ▶  -define-G-linear_group-PGGL_3_4-end\`
4101  ▶  ▶  -with-G-do\`
4102  ▶  ▶  -group_theoretic_activity\`
4103  ▶  ▶  ▶  -report\`
4104  ▶  ▶  ▶  -sylow\`
4105  ▶  ▶  ▶  -classes\`
4106  ▶  ▶  -end
4107  ▶  pdflatex PGGL_3_4_report.tex
4108  ▶  open-PGGL_3_4_report.pdf
4109
4110
4111
4112  PGGL_3_8:
4113  ▶  $(ORBITER\_PATH)orbiter.out-v-5\`
4114  ▶  ▶  -define-G-linear_group-PGGL_3_8-end
4115
4116
4117  PGGL_3_8_report:
4118  ▶  $(ORBITER\_PATH)orbiter.out-v-3\`
4119  ▶  ▶  -define-G-linear_group-PGGL_3_8-end\`
4120  ▶  ▶  -with-G-do\`
4121  ▶  ▶  -group_theoretic_activity\`
4122  ▶  ▶  ▶  -report\`
4123  ▶  ▶  -end
4124  ▶  pdflatex PGGL_3_8_report.tex
4125  ▶  open-PGGL_3_8_report.pdf
4126
4127
\end{verbatim}
4128
4129  PG0_5_2:
4130  ▶ $(ORBITER_PATH)orbiter.out-v.2\$
4131  ▶ ▶ -define F -finite_field -q 2 -end\$
4132  ▶ ▶ -define G -linear_group -PGO 5 -F -end\$
4133  ▶ ▶ -with G -do\$
4134  ▶ ▶ -group_theoretic_activity\$
4135  ▶ ▶ ▶ -report\$
4136  ▶ ▶ -end
4137  ▶ pdflatex- PG0_5_2_report.tex
4138  ▶ open- PG0_5_2_report.pdf
4139
4140  PGG0_5_4:
4141  ▶ $(ORBITER_PATH)orbiter.out-v.2\$
4142  ▶ ▶ -define F4 -finite_field -q 4 -end\$
4143  ▶ ▶ -define G -linear_group -PGG0 5 -F4 -end\$
4144  ▶ ▶ -with G -do\$
4145  ▶ ▶ -group_theoretic_activity\$
4146  ▶ ▶ ▶ -report\$
4147  ▶ ▶ -end
4148  ▶ pdflatex- PGG0_5_4_report.tex
4149  ▶ open- PGG0_5_4_report.pdf
4150
4151
4152
4153  PGOp_6_2:
4154  ▶ $(ORBITER_PATH)orbiter.out-v.2\$
4155  ▶ ▶ -define F -finite_field -q 2 -end\$
4156  ▶ ▶ -define G -linear_group -PGOp 6 -F -end\$
4157  ▶ ▶ -with G -do\$
4158  ▶ ▶ -group_theoretic_activity\$
4159  ▶ ▶ ▶ -report\$
4160  ▶ ▶ -end
4161  ▶ pdflatex- PGOp_6_2_report.tex
4162  ▶ open- PGOp_6_2_report.pdf
4163
4164  PGOm_6_2:
4165  ▶ $(ORBITER_PATH)orbiter.out-v.2\$
4166  ▶ ▶ -define F -finite_field -q 2 -end\$
4167  ▶ ▶ -define G -linear_group -PGOm 6 -F -end\$
4168  ▶ ▶ -with G -do\$
4169  ▶ ▶ -group_theoretic_activity\$
4170  ▶ ▶ ▶ -report\$
4171  ▶ ▶ -end
4172  ▶ pdflatex- PGOm_6_2_report.tex
4173  ▶ open- PGOm_6_2_report.pdf
4174
## Section 5.3: Subgroups

### J1:
```bash
$ (ORBITER_PATH) orbiter.out -v 3
```
```bash
define F finite_field q 2 end
```
```bash
define G linear_group PGL 7 11 Janko1 end
```
```bash
with G do
```
```bash
 group_theoretic_activity
```
```bash
report
```
```bash
end
```
```bash
pdflatex PGL 7 11 Subgroup Janko1 report.tex
```
```bash
open PGL 7 11 Subgroup Janko1 report.pdf
```
```bash
PGL 3 11 singer:
```
4222  > $(ORBITER_PATH)orbiter.out -v 3 \n4223  > define G - linear group - PGL 3 11 - singer 19 - end \n4224  > -with G - do \n4225  > -group theoretic activity \n4226  > -report \n4227  > -end \n4228  > pdflatex PGL 3 11 Singer 3 11 19 report.tex \n4229  > open PGL 3 11 Singer 3 11 19 report.pdf \n4230 \n4231 PGL 3 11 singer and frobenius: \n4232  > $(ORBITER_PATH)orbiter.out -v 3 \n4233  > define G - linear group - PGL 3 11 - singer and frobenius 19 - end \n4234  > -with G - do \n4235  > -group theoretic activity \n4236  > -report \n4237  > -end \n4238  > pdflatex PGL 3 11 Singer and Frob 3 11 19 report.tex \n4239  > open PGL 3 11 Singer and Frob 3 11 19 report.pdf \n4240 \n4241 PG 2 4 order 21: \n4242  > $(ORBITER_PATH)orbiter.out -v 5 \n4243  > define G - linear group - PGL 3 4 - end \n4244  > -with G - do \n4245  > -group theoretic activity \n4246  > -search element of order 21 \n4247  > -report \n4248  > -end \n4249 \n4250 \n4251 \n4252 quaternion: \n4253  > $(ORBITER_PATH)orbiter.out -v 30 \n4254  > define G - linear group - SL 2 3 \n4255  > subgroup by generators "quaternion". "8". 3 \n4256  > "1, 1, 1, 2, 2, 1, 1, 1, 0, 2, 1, 0" \n4257  > -end \n4258  > -with G - do \n4259  > -group theoretic activity \n4260  > -print elements tex \n4261  > -group table \n4262  > -report \n4263  > -end \n4264  > pdflatex GL 2 3 Subgroup quaternion 8 elements.tex \n4265  > open GL 2 3 Subgroup quaternion 8 elements.pdf \n4266  > pdflatex GL 2 3 Subgroup quaternion 8 report.tex \n4267  > open GL 2 3 Subgroup quaternion 8 report.pdf \n4268
4269
4270
cube_group:
4272 ▷ $(ORBITER_PATH)orbiter.out-v.3:\
4273 ▷ ▷ -define-gens-vector-dense\
4274 ▷ ▷ ▷ "0,1,0,2,0,0,0,0,1,\"
4275 ▷ ▷ ▷ 0,0,1,0,1,0,2,0,0,\"
4276 ▷ ▷ ▷ 2,0,0,0,1,0,0,0,1".\n4277 ▷ ▷ -end:\
4278 ▷ ▷ -define-G-linear_group-GL-3-3:\
4279 ▷ ▷ -subgroup_by_generators:"cube":"48":3-3\n4280 ▷ ▷ ▷ gens\n4281 ▷ ▷ ▷ -end:\n4282 ▷ ▷ -with-G-do:\n4283 ▷ ▷ ▷ -group_theoretic_activity.\n4284 ▷ ▷ ▷ ▷ -print_elements.tex\n4285 ▷ ▷ ▷ ▷ -report.\n4286 ▷ ▷ ▷ -end\n4287 ▷ ▷ pdflatex-GL_3_3_Subgroup_cube_48_report.tex
4288 ▷ ▷ open-GL_3_3_Subgroup_cube_48_report.pdf
4289 ▷ ▷ pdflatex-GL_3_3_Subgroup_cube_48_elements.tex
4290 ▷ ▷ open-GL_3_3_Subgroup_cube_48_elements.pdf
4291
tetra_group:
4293 ▷ $(ORBITER_PATH)orbiter.out-v.3:\n4294 ▷ ▷ -define-G-linear_group-GL-3-3:\n4295 ▷ ▷ ▷ -subgroup_by_generators:"tetra":"12":2-2\n4296 ▷ ▷ ▷ ▷ "0,1,0,0,0,1,1,0,0,-0,0,1,2,0,0,0,2,0".\n4297 ▷ ▷ ▷ ▷ -end:\n4298 ▷ ▷ ▷ -with-G-do:\n4299 ▷ ▷ ▷ -group_theoretic_activity.\n4300 ▷ ▷ ▷ ▷ -print_elements.tex\n4301 ▷ ▷ ▷ ▷ -report.\n4302 ▷ ▷ ▷ ▷ -end\n4303 ▷ ▷ ▷ pdflatex-GL_3_3_Subgroup_tetra_12_report.tex
4304 ▷ ▷ ▷ open-GL_3_3_Subgroup_tetra_12_report.pdf
4305 ▷ ▷ ▷ pdflatex-GL_3_3_Subgroup_tetra_12_elements.tex
4306 ▷ ▷ ▷ open-GL_3_3_Subgroup_tetra_12_elements.pdf
4307
tetra_group:
4311 ▷ $(ORBITER_PATH)orbiter.out-v.3:\n4312 ▷ ▷ -define-gens-vector-compact.\n4313 ▷ ▷ ▷ $(GENERATORS_HESSE_GROUP).\n4314 ▷ ▷ ▷ -end.\n4315
523
define G=linear group=PGGL 3.4
subgroup by generators="Hesse"."432".7-gens.
-end
-with G-do
-group_theoretic_activity
-report
-end

pdflatex PGGL 3.4 Subgroup_Hesse 432_report.tex
open PGGL 3.4 Subgroup_Hesse 432_report.pdf

#Hesse group:
#1,0,0,0,1,0,0,0,1,0,3,2,3,2,0,
#1,0,0,0,1,0,0,3,1,2,0,1,0,1,3,0,
#1,0,0,0,1,0,0,0,1,1,0,0,0,0,1,1,
#1,0,0,0,2,2,0,0,2,0,0,0,0,0,1,0,
#1,0,0,0,2,3,1,0,2,0,1,0,3,1,3,1,0,
#0,1,1,0,2,0,0,0,1,3,2,0,2,1,1,2,1,
#1,1,0,0,0,0,3,0,3,2,2,0,1,2,3,3,1,

Weyl E8:
$(ORBITER_PATH)orbiter.out-v.3
-definition-gens-vector-dense
$(GENERATORS_WEYL_GROUP_E8)
-end

define G=linear group=GL 8.3
subgroup by generators
"Weyl_E8"."696729600".2
$(GENERATORS_WEYL_GROUP_E8)
-end
-with G-do
-group_theoretic_activity
-report
-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:

```bash
> $(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"\n
> -end\n
> -with G=do\n
> -group_theoretic_activity\n
> -report\n
> -end

> pdflatex-PGL_3_9_Subgroup_U_3_3_6048_report.tex

> open-PGL_3_9_Subgroup_U_3_3_6048_report.pdf

PGL_2_3:

```bash
> $(ORBITER_PATH)orbiter.out -v.3\n
> -define G=linear_group=PGL.2.3 -end\n
> -with G=do\n
> -group_theoretic_activity\n
> -report\n
> -group_table\n
> -end

> pdflatex-PGL_2_3_group_table_order_24.tex

> pdflatex-PGL_2_3_report.tex

> open-PGL_2_3_group_table_order_24.pdf

> open-PGL_2_3_report.pdf

```

#Co3 from Conway et al., 1985 (ATLAS)

#order=.495766656000

#Co3 from the paper by Suleiman and Wilson, 1997

Co3 from the paper by Suleiman and Wilson, 1997
Co3:
$\text{define } F \text{-finite field } q^2 \text{-end}$
$\text{-define } g1 \text{-vector } F \text{-format } 22 \text{-compact } (\text{CONWAY GEN1}) \text{-end}$
$\text{-define } g2 \text{-vector } F \text{-format } 22 \text{-dense } (\text{Ree gen1}) \text{-end}$
$\text{-define } G \text{-linear group } \text{PGL } 7 \cdot F \text{-end}$
$\text{-define } G \text{-linear group } \text{PGL } 22.2 \text{-end}$
$\text{-define } G \text{-subgroup by generators } \text{"Co3" } 495766656000 \text{-2.gens}$
$\text{-define } G \text{-subgroup by generators } \text{"Ree } 27" \text{-10073444472 } \text{-2.gens}$
$\text{-define } G \text{-with } G \text{-do}$
$\text{-group_theoretic_activity}$
$\text{-report}$
$\text{-end}$
$\text{pdflatex PGL_22.2.Subgroup_Co3.495766656000.report.tex}$
$\text{open PGL_22.2.Subgroup_Co3.495766656000.report.pdf}$

Ree_27:
$\text{define } F \text{-finite field } q^{27} \text{-override polynomial } "34" \text{-end}$
$\text{-define } g1 \text{-vector } F \text{-format } 7 \text{-dense } (\text{Ree gen1}) \text{-end}$
$\text{-define } g2 \text{-vector } F \text{-format } 7 \text{-dense } (\text{Ree gen2}) \text{-end}$
$\text{-define } G \text{-linear group } \text{PGL } 7 \cdot F \text{-end}$
$\text{-define } G \text{-subgroup by generators } \text{"Ree } 27" \text{-10073444472 } \text{-2.gens}$
$\text{-define } G \text{-with } G \text{-do}$
$\text{-group_theoretic_activity}$
$\text{-report}$
$\text{-end}$

#needs a lot of memory to run!

T3 on tensors:
$\text{define } G \text{-linear group } \text{GL } d \cdot q \cdot \text{wr Sym } n \cdot 2 \cdot 2 \cdot 3 \text{-end}$
4456 \>> -on_tensors\end\  
4457 \>> -with_G\-do\  
4458 \>> group_theoretic_activity\  
4459 \>> \report\  
4460 \>> -end\  
4461 \> pdflatex\ · \GL_2_2\_wreath\_Sym3\_report.tex  
4462 \> open\-GL_2_2\_wreath\_Sym3\_report.pdf  
4463 \>  
4464  
4465  
4466 T3r1:\  
4467 \> $(\text{ORBITER\_PATH})\text{orbiter.out}-v.4\  
4468 \> \define \G  
4469 \> \-linear_group\-GL_d.q\_wr\_Sym.n\cdot2\cdot2\cdot3\  
4470 \> \> \-on\_rank\_one\_tensors\-end\  
4471 \> \> \-with\_G\-do\  
4472 \> \> group_theoretic_activity\  
4473 \> \> \report\  
4474 \> \> -end\  
4475 \> pdflatex\ · \GL_2_2\_wreath\_Sym3\_report.tex  
4476 \> open\-GL_2_2\_wreath\_Sym3\_report.pdf  
4477  
4478  
4479  
4480  
4481  
4482 T4\_on\_tensors:\  
4483 \> $(\text{ORBITER\_PATH})\text{orbiter.out}-v.4\  
4484 \> \define \G  
4485 \> \-linear_group\-GL_d.q\_wr\_Sym.n\cdot2\cdot2\cdot4\  
4486 \> \> \-on\_tensors\-end\  
4487 \> \> \-with\_G\-do\  
4488 \> \> group_theoretic_activity\  
4489 \> \> \report\  
4490 \> \> -end\  
4491 \> pdflatex\ · \GL_2_2\_wreath\_Sym4\_report.tex  
4492 \> open\-GL_2_2\_wreath\_Sym4\_report.pdf  
4493 \>  
4494 \>  
4495 T4r1:\  
4496 \> $(\text{ORBITER\_PATH})\text{orbiter.out}-v.4\  
4497 \> \define \G  
4498 \> \-linear_group\-GL_d.q\_wr\_Sym.n\cdot2\cdot2\cdot4\  
4499 \> \> \-on\_rank\_one\_tensors\-end\  
4500 \> \> \-with\_G\-do\  
4501 \> \> group_theoretic_activity\  
4502 \> \> \report\  
527
### Section 5.6: Group-Theoretic Activities

**SECTION_GROUP_THEORETIC_ACTIVITIES:**

**PGL\_3\_2\_elements:**

```bash
$(ORBITER\_PATH)orbi\_test.out..v.5/\n```

**PGL\_3\_4\_singer:**

```bash
$(ORBITER\_PATH)orbi\_test.out..v.5/\n```

**GL\_2\_8\_multiply:**

```bash
$(ORBITER\_PATH)orbi\_test.out..v.5/\n```
GL_2.8_multiply:

GL_2.8_inv:

GL_2.8_power:

PGL3.2_classes:
normal_forms_PGL_4.4:

PGL_4.4_2A_rank:

PGL_4.4_2A_unrank:

ToDo:

cent_2A:
PGL(4,5) rank:

define G linear group PGL(4,5) end

with G do

group theoretic activity

element rank:0,0,0,1,2,3,0,1,0,3,4,4,0,1,2,1 end

end

PGL(4,5) unrank:

define G linear group PGL(4,5) end

with G do

group theoretic activity

element unrank:701459351 end

end

#eigen 3A:

$(ORBITER_PATH)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" end

#eigen 3B:

$(ORBITER_PATH)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" end

#element-of-order-31-in-PGL(4,5):

int data[] = {1,0,0,0,0,1,0,0,0,0,1,0,1,3,0};
normal_forms_PGL_4_5:
 $(ORBITER_PATH)orbiter.out-v.7:\
 -define-G-linear_group-PGL.4.5-end\
 -with-G-do\
 -group_theoretic_activity\
 -classes_based_on_normal_form\
 -end
 pdflatex-PGL_4_5_classes_normal_form.txt
 open-PGL_4_5_classes_normal_form.pdf

SECTION_GROUP_THEORETIC_ACTIVITIES_BASED_ON_MAGMA:

PGGL_2_4_classes:
 $(ORBITER_PATH)orbiter.out-v.3:\
 -define-G\n -linear_group-PGGL.2.4-end\n -with-G-do\n -group_theoretic_activity\n -classes-end
 $(MAGMA_PATH)/magma-PGGL_2_4_classes.magma
 $(MAGMA_PATH)/magma-PGGL_2_4_classes.magma
 open-PGL_2_4_classes_out.pdf

PGGL_2_4_cent_2A:
4736 $(ORBITER_PATH)orbiter.out -v 3$
4737  define-G $
4738  linear_group -PGGL 2 4 -end $
4739  with-G -do $
4740  group_theoretic_activity $
4741  centralizer_of_element "2A" "1,0,0,1,1" $
4742  report $
4743  end
4744 $(MAGMA_PATH)/magma -element 2A -centralizer.magma
4745 $(ORBITER_PATH)orbiter.out -v 6 $
4746  define-G $
4747  linear_group -PGGL 2 4 -end $
4748  with-G -do $
4749  group_theoretic_activity $
4750  centralizer_of_element "2A" "1,0,0,1,1" $
4751  report $
4752  end
4753 pdflatex PGGL 2 4 elt 2A -centralizer.tex
4754 open-PGGL 2 4 elt 2A -centralizer.pdf
4755
4756
4757
4758
4759
4760
4761 PGGL 3 4 classes:
4762 $(ORBITER_PATH)orbiter.out -v 3 $
4763  define-G $
4764  linear_group -PGGL 3 4 $
4765  end $
4766  with-G -do $
4767  group_theoretic_activity $
4768  centralizer -classes $
4769  end
4770 pdflatex PGGL 3 4 classes out.tex
4771 open-PGGL 3 4 classes out.pdf
4772
4773
4774
4775 #1,3,3,1,3,2,3,0,3,0, #
4776 is an element of order 21
4777
4778
4779 # subgroup_by_generators "singer" "21" 1 $
4780 # "0,1,0,0,0,1,-2,1,1" $
4781
4782
classes_PGGL_4_4:
  $(ORBITER_PATH)orbiter.out-v.3\$
  -magma_path/usr/local/magma/\$
  -define-G\$
  -linear_group-PGGL_4_4-end\$
  -with-G-do\$
  -group_theoretic_activity\$
  -classes\$
  -end

#group.order.1974067200=2^13*3^4*5^2*7*17

#the-find_subgroup-command-is-too-specialized

subgroups_PGL_4_5:
  $(ORBITER_PATH)orbiter.out-v.6\$
  -define-G\$
  -linear_group-PGL_4_5-end\$
  -with-G-do\$
  -group_theoretic_activity\$
  -find_subgroup.3\$
  -end
  pdflatex-PGL_4_5_report.tex
  open-PGL_4_5_report.pdf

classes_PGL_4_5:
  $(ORBITER_PATH)orbiter.out-v.6\$
  -define-G\$
  -linear_group-PGL_4_5-end\$
  -with-G-do\$
  -group_theoretic_activity\$
  -classes\$
  -end
  pdflatex-PGL_4_5_classes_out.tex
  open-PGL_4_5_classes_out.pdf

#163-classes
4830  #two-classes-of-elements-of-order-3
4831  #Order-of-element=3-Class-size=310000-Centralizer-order=93600-Normalizer-order=187200
4832  #of-order-3-and-with-0-fixed-points.
4833  #0,1,0,2,0,1,2,1,4,2,3,1,2,0,4,3,
4834  #Class-size=10075000-Centralizer-order=2880-Normalizer-order=5760
4835  #of-order-3-and-with-6-fixed-points.
4836  #0,0,0,1,2,3,0,1,0,3,4,4,0,1,2,1,
4837  PGL_4.5_3B_class_again:
4838  $\langle ORBITER PATH \rangle$ orbiter.out -v 6 -define G:
4839  -linear_group $\langle -$GL-4.5$\rangle$ -end$
4840  -with G$ -do$
4841  -group_theoretic_activity$ -do$
4842  -conjugacy_class_of$
4843  "0,0,0,1,-2,3,0,1,-0,3,4,4,-0,1,2,1$ -end
4844  search_primitive_poly_q5_deg3:
4845  $\langle ORBITER PATH \rangle$ orbiter.out -v 6$
4846  -search_for_primitive_polynomial_in_range 5:5:3:3
4847  OK, we found an irreducible and primitive polynomial $X^3 + X^2 + 2$
4848  GL_3.5_singer_power:
4849  $\langle ORBITER PATH \rangle$ orbiter.out -v 6 -define G$
4850  -linear_group $\langle -$GL-3.5$\rangle$ -end$
4851  -with G$ -do$
4852  -group_theoretic_activity$ -do$
4853  -raise_to_the_power$
4854  "0,1,0,-0,0,1,-3,0,4"31$ -end$
4855  pdflatex GL_3.5_power.tex
4856  open GL_3.5_power.pdf
4857  PGL_4.5_norm_31:
4858  $\langle ORBITER PATH \rangle$ orbiter.out -v 6 -define G$
4859  -linear_group $\langle -$GL-4.5$\rangle$ -end$
4860  -with G$ -do$
4861  -group_theoretic_activity$ -do$
4862  -normalizer_of_cyclic_subgroup "31" -end$
4863  "2,0,0,0,-0,0,1,0,-0,0,0,0,1,-0,3,0,4$ -end
4864  pdflatex normalizer_of_31_in_PGL_4.5.tex
4865  open normalizer_of_31_in_PGL_4.5.pdf
Normalizer of \( \mathbb{Z}_{22} \) in \( \text{PGL}_2(9) \):

- \((\text{ORBITER\_PATH})\text{orbiter.out}\) -v 4
- \text{-define}\( G \) -linear_group -PGL_2.9
- \text{-subgroup\_by\_generators}\( \mathbb{Z}_{22} \cdot 4 \)
- \text{-end}\( "2,0,0,1,0,1,1,0"\)
- \text{-with}\( G \) -do
- \text{-group\_theoretic\_activity}\( G \)
- \text{-normalizer}\( G \)
- \text{-end}

\text{pdflatex} \( \text{PGL}\_2.9\) Subgroup \( \mathbb{Z}_{22} \cdot 4 \) normalizer.tex
\text{open} \( \text{PGL}\_2.9\) Subgroup \( \mathbb{Z}_{22} \cdot 4 \) normalizer.pdf

T3r1 orbits:

- \((\text{ORBITER\_PATH})\text{orbiter.out}\) -v 4
- \text{-define}\( G \)
- \text{-linear\_group}\( \text{-GL}\_d\_q\_wr\_Sym\_n\cdot 2.2.3\)
- \text{-on\_rank\_one\_tensors}\( G \)
- \text{-with}\( G \) -do
#pdflatex GL 2 2 wreath Sym3 report.tex
#open GL 2 2 wreath Sym3 report.pdf

pdflatex · GL 2 2 wreath Sym3 report.tex
open · GL 2 2 wreath Sym3 report.pdf

T3r1 orbits draw:
$\text{(ORBITER PATH)orbiter.out}$ · v · 3
-draw layered graph -GL 2 2 wreath Sym3 res27 0.layered graph
-radius 500 -spanning tree -embedded -line width 1.1 -x stretch 1.4 -scale 0.25
-end

#pdflatex GL 2 2 wreath Sym3 report.tex
#open GL 2 2 wreath Sym3 report.pdf

write · GL 2 2 wreath Sym3 res27 0.layered graph

2C orbit under PGGL 4 4.elements coded.csv:
$\text{(ORBITER PATH)orbiter.out}$ · v · 6
-define G · linear group -PGGL 4 4 -end
-with G · do
-group theoretic activity
-conjugacy class of element
"2C"."1,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0"
-end

#class of size 64260
#creates:
2C orbit under PGGL 4 4.csv
2C orbit under PGGL 4 4.txt
2C orbit under PGGL 4 4.elements coded.csv
2C orbit under PGGL 4 4.transporter.csv
1:33 on Mac
#User time: 2:59 on Mac
PGGL 4 4 subgroups of type 2C 2C: 2C orbit under PGGL 4 4.elements coded.csv
$\text{(ORBITER PATH)orbiter.out}$ · v · 6
-define G · linear group -PGGL 4 4
-subgroup by generators "centralizer 2C"."30720".9"
The distribution of orbit lengths is: (1, 2, 15, 20, 24^3, 30, 40, 240, 256, 480, 512, 960^2, 1280, 1920^2, 2560^4, 3840, 5120, 6144^3, 7680, 10240)

found 29 conjugacy classes

User time: 0:57

orbits on conics q13:

orbits.cubic.curves.q2:

orbits.cubic.curves.q3:

orbits.on.polynomials.2:

orbits_on_group_elements_under_conjugation:

orbits_on_group_elements_under_conjugation_after_classes.compute_all_point_orbits

open-subgroups_of_order_4.pdf
orbits_cubic_curves_q2_with_draw_tree:

```
#pdflatex poly_orbits_d3_n3_q2.tex
#open poly_orbits_d3_n3_q2.pdf
```

draw_tree:

```
$(ORBITER_PATH)orbiter.out -v:4 -
  -draw_options:-yout:500000:-radius:15:-nodes_empty:
  -line_width:0.5:-y_stretch:0.25:-embedded:-end:
  -define-G:-linear_group:-PGL:3:2:-end:
  -with-G:-do:
  -group_theoretic_activity:
  -orbits_on_polynomials:3:
  -orbits_on_polynomials_draw_tree:6:
  -end:
```

t4_orbits:

```
pdflatex poly_orbits_d3_n3_q2_get_ranks:
#pdflatex poly_orbits_d3_n3_q2.tex
#open poly_orbits_d3_n3_q2.pdf
```

GL2_2_wreath_Sym4_orbits:

```
$(ORBITER_PATH)orbiter.out -v:4 -
  -define-G:
  -linear_group:-GL:2:2_wreath_Sym:4:-end:
  -on_tensors:-end:
  -with-G:-do:
  -group_theoretic_activity:
  -report:
  -orbits_on_points:
  -end
```

draw_tree:

```
$(ORBITER_PATH)orbiter.out -v:4 -
  -draw_options:-yout:500000:-radius:15:-nodes_empty:
  -line_width:0.5:-y_stretch:0.25:-embedded:-end:
  -define-G:-linear_group:-PGL:4:2:-end:
  -with-G:-do:
  -group_theoretic_activity:
  -orbits_on_polynomials:3:
  -orbits_on_polynomials_draw_tree:6:
  -end:
```

csv_file:

```
$(ORBITER_PATH)orbiter.out -v:4 -
  -draw_options:-yout:500000:-radius:15:-nodes_empty:
  -line_width:0.5:-y_stretch:0.25:-embedded:-end:
  -define-G:-linear_group:-PGL:4:2:-end:
  -with-G:-do:
  -group_theoretic_activity:
  -orbits_on_polynomials:3:
  -orbits_on_polynomials_draw_tree:6:
  -end:
```

tensors:

```
$(ORBITER_PATH)orbiter.out -v:4 -
  -define-G:
  -linear_group:-GL:2:2_wreath_Sym:4:-end:
  -on_tensors:-end:
  -with-G:-do:
  -group_theoretic_activity:
  -report:
  -orbits_on_points:
  -end
```

csv_file:

```
$(ORBITER_PATH)orbiter.out -v:4 -
  -csv_file:poly_orbits_d3_n3_q2.csv:0
```

generate_ranks:

```
pdflatex poly_orbits_d3_n3_q2.csv:0
#open poly_orbits_d3_n3_q2.csv:0
```

GL2_2_wreath_Sym4_res65535_orbits:

```
pdflatex GL:2:2_wreath_Sym:4_res65535_orbits.tex
open GL:2:2_wreath_Sym:4_res65535_orbits.pdf
```
T4r1\_orbits:
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -4\$

\> -define\ G\ -linear\ group\ -GL\ d\ q\ wr\ Sym\ n\ 2\ 2\ 4\$
\> -on\ rank\ one\ tensors\ -end\$
\> -with\ G\ -do\$
\> -group\_theoretic\_activity\$
\> -orbits\_on\ points\$
\> -export\_trees\$
\> -report\$
\> -end

#pdflatex\ -open\ GL\ 2\ 2\ wreath\ Sym\ 4\_orbits\ report.tex
#open\ GL\ 2\ 2\ wreath\ Sym\ 4\_orbits\ report.pdf

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

\> -draw\_layered\_graph$
\> GL\ 2\ 2\ wreath\ Sym\ 4\_res\ 81\ 0\_layered\_graph$
\> -radius\ 400\ -spanning\_tree\ -embedded$
\> -line\_width\ 1.1\ -x\_stretch\ 2.5\ -scale\ 0.15$
\> -end

#pdflatex\ GL\ 2\ 2\ wreath\ Sym\ 3\_orbits\ report.tex
#open\ GL\ 2\ 2\ wreath\ Sym\ 3\_orbits\ report.pdf

T4r1\_orbits\_4$
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -4$

\> -orbiter\_path\ $(\text{ORBITER\_PATH})$
\> -define\ G\ -linear\ group\ -GL\ d\ q\ wr\ Sym\ n\ 2\ 2\ 4$
\> -on\ rank\ one\ tensors\ -end$
\> -with\ G\ -do$
\> -group\_theoretic\_activity$
\> -poset\_classification\_control\ -problem\_label\ T4r1\_W$
\> -bit\_depth\ 4\ -draw\_options\ -end\ -draw\ poset\ -report\ -end$
\> -orbits\_on\ subsets\ 4$
\> -report$
\> -end

#pdflatex\ T4r1\ poset.tex
#open\ T4r1\ poset.pdf
PGGL\_2\_8\_on\_conic\_orbits:

\$(\text{ORBITER\_PATH})\text{orbiter.out}\!-\!v\!-\!4\$

\$\text{define}\text{-}\!G\!$

\$\text{-linear}\text{-}G\text{-PGGL\_2\_8\_PGL2OnConic\_end}\$

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

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

\$\text{-orbits}\text{-on}\text{-}points\$

\$\text{-report}\$

\$\text{-end}\$

pdf\_latex\: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\!-\!4\$

\$\text{define}\text{-}\!G\!$

\$\text{-linear}\text{-}G\text{-PGGL\_7\_8\_end}\$

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

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

\$\text{-report}\$

\$\text{-orbits}\text{-on}\text{-}points\$

\$\text{-end}\$

#1\!-\!min\!-\!31\!-\!sec\!-\!on\!Mac

SECTION POSET CLASSIFICATION:

poset \_of \_4\_subsets:

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

\$\text{-orbiter}\text{-path}\text{-}$(\text{ORBITER\_PATH})\$

\$\text{-define}\text{-}G\text{-linear}\text{-}G\text{-PGL\_2\_3\_identity}\text{-}G\text{-end}\$

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

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

---
poset classification control

- problem_label:poset_4
- W-depth:4
- draw_options:-radius:200
- report:-end
- draw_poset:
- end
- orbits_on_subsets:4
- report:
- end

poset_of_4subsets_draw:

$\text{(ORBITER PATH)}$ orbiter.out:-v:3
- draw_layered_graph
- poset_4_poset_lvl_4.layered_graph:
- radius:300:-embedded:-line_width:1.1
- y_stretch:0.9:-scale:0.25
- end

pdflatex:poset_4_poset_lvl_4.draw.tex
open:poset_4_poset_lvl_4.draw.pdf

poset_of_5subsets:

$\text{(ORBITER PATH)}$ orbiter.out:-v:3
- orbiter_path:$\text{(ORBITER PATH)}$
- define:
  G:-linear_group:-PGL:2:-identity_group:-end
- with:
  G:-do
- group_theoretic_activity:
- poset_classification_control:
- problem_label:poset_5:
- W-depth:5:-draw_options:-radius:150:-end
- report:-end:-draw_poset:-end
- orbits_on_subsets:5
- report:
- end

pdflatex:poset_5_poset.tex
open:poset_5_poset.pdf

poset_of_5subsets_draw:

$\text{(ORBITER PATH)}$ orbiter.out:-v:3
- draw_layered_graph
- poset_5_poset_lvl_5.layered_graph:
- radius:300:-embedded
V_3.2_trivial:
$(ORBITER_PATH)oriber.out:-v.5:\$

-define-G-linear_group-PGL.3.2-identity_group:-end:\$

-group.theoretic_activity:\$

-poset_classification_control:\$

-problem_label-V_3.2_trivial:\$

- depth.3.-node_label_is_element:\$

-draw_options:\$

-radius.-200.-embedded:\$

-end:\$

-report.-end:\$

-draw_poset:\$

-end:\$

-orbits_on_subspaces.3:\$

-report:\$

-end:\$

$(ORBITER_PATH)oriber.out:-v.5:\$

-draw_layered_graph-PGL.3.2-Identity_3.2_poset_lvl.3.layered_graph\$

-radius.-300.-embedded-line_width.1.1.-y_stretch.0.9.-scale.0.25\$

-pdflatex-PGL.3.2-Identity_3.2_report.tex

-open-PGL.3.2-Identity_3.2_report.pdf

-pdflatex-PGL.3.2-Identity_3.2_poset_lvl.3_draw.tex

-open-PGL.3.2-Identity_3.2_poset_lvl.3_draw.pdf

-pdflatex-PGL.3.2-Identity_3.2_poset.tex

-open-PGL.3.2-Identity_3.2_poset.pdf

V_4.2_trivial:
$(ORBITER_PATH)oriber.out:-v.5:\$

-define-G-linear_group-PGL.4.2-identity_group:-end:\$

-group.theoretic_activity:\$

-poset_classification_control:\$
Section 6.3: Orbits on Subsets

\text{PG\_2\_2 subsets:}
$\$(\text{ORBITER\_PATH})\text{orbiter.out }-v.3\$
$\$(\text{ORBITER\_PATH})\text{-orbiter.path}\$
$\$(\text{ORBITER\_PATH})\text{-define}\_F\_finite_field\_q.2\$
$\$(\text{ORBITER\_PATH})\text{-define}\_G\_linear_group\_PGL\_3\_F\$
$\$(\text{ORBITER\_PATH})\text{-with}\_G\_do\$
$\$(\text{ORBITER\_PATH})\text{-group.theoretic.activity}\$
$\$(\text{ORBITER\_PATH})\text{-poset.classification.control}\$
$\$(\text{ORBITER\_PATH})\text{-problem.label}\_PGL\_3\_2\$
$\$(\text{ORBITER\_PATH})\text{-depth.7}\$
$\$(\text{ORBITER\_PATH})\text{-draw.options}\$
$\$(\text{ORBITER\_PATH})\text{-radius.200}\$
$\$(\text{ORBITER\_PATH})\text{-end}\$
$\$(\text{ORBITER\_PATH})\text{-report.end}\$
$\$(\text{ORBITER\_PATH})\text{-draw.poset}\$
$\$(\text{ORBITER\_PATH})\text{-end}\$
$\$(\text{ORBITER\_PATH})\text{-orbits.on.subspaces.7}\$
$\$(\text{ORBITER\_PATH})\text{-report.end}\$
$\$(\text{ORBITER\_PATH})\text{-draw.poset}\$
$\$(\text{ORBITER\_PATH})\text{-end}\$
$\$\text{pdflatex}\_\text{PGL\_3\_2}\_\text{poset.lvl.7}.tex\$

\text{PGL\_4\_2 Identity 4.2 report:}

\text{PGL\_4\_2 Identity 4.2 report.pdf}

\text{PGL\_4\_2 Identity 4.2 tree_lvl.4.tex}

\text{PGL\_4\_2 Identity 4.2 tree_lvl.4.pdf}

\text{PGL\_4\_2 Identity 4.2 poset.tex}

\text{PGL\_4\_2 Identity 4.2 poset.pdf}
5304 $\#\mathrm{PG}(3,2)$ has $2^3+2^2+2^1+1=15$ points.
5305 $\#\mathrm{PG}(3,3)$ has $3^3+3^2+3^1+1=27+9+3+1=40$ points.

5309 $\mathrm{PG}_3,2.$ subsets again:
5310 $\$(\text{ORBITER PATH})\text{orbiter.out}-v:3\$
5311 $\text{-orbiter_path}$$(\text{ORBITER PATH})\$
5312 $\text{-define} F$finite field$q=2$end$
5313 $\text{-define} G$linear group$\mathrm{PGL}_{4,F}$end$
5314 $\text{-with} G$do$
5315 $\text{-group_theoretic_activity}$
5316 $\text{-poset_classification_control}$
5317 $\text{-problem_label}$PGL$_{4,2}$
5318 $\text{-depth}15$
5319 $\text{-draw_options}$
5320 $\text{-radius}200$embedded
5321 $\text{-end}$
5322 $\text{-report}$end
5323 $\text{-draw_poset}$
5324 $\text{-orbits}$on subsets$15$
5325 $\text{-report}$
5326 $\text{-end}$
5327 $\$\text{pdflatex}\mathrm{PGL}_{4,2}\text{poset.lvl.15.tex}$
5328 $\$\text{open-PGL}_{4,2}\text{poset.lvl.15.pdf}$
5329 $\$\text{pdflatex}\mathrm{PGL}_{4,2}\text{poset.tex}$
5330 $\$\text{open-PGL}_{4,2}\text{poset.pdf}$
5331 $\$\text{pdflatex}\mathrm{PGL}_{4,2}\text{poset.tex}$
5332 $\$\text{open-PGL}_{4,2}\text{poset.pdf}$
5333 $\$\text{pdflatex}\mathrm{PGL}_{4,2}\text{poset.detailed.lvl.15.tex}$
5334 $\$\text{open-PGL}_{4,2}\text{poset.detailed.lvl.15.pdf}$
5335 5336 5337 5338 $\mathrm{PG}_3,2.$ subsets:
5339 $\$(\text{ORBITER PATH})\text{orbiter.out}-v:3\$
5340 $\text{-orbiter_path}$$(\text{ORBITER PATH})\$
5341 $\text{-define} F$finite field$q=2$end$
5342 $\text{-define} G$linear group$\mathrm{PGL}_{4,F}$end$
5343 $\text{-with} G$do
PGL_3_2_singer:
$(ORBITER_PATH)orbiter.out-v.3\$
-define-G-linear_group-PGL_3_2-singer.1-end\$
-with-G-do\$
group_theoretic_activity\$
-poset_classification_control\$
-problem_label-PGL_3_2_singer.1-W-depth.7\$
-draw_poset\$
-report-end\$
-end\$
-orbits_on_subsets.7\$
-report\$
-end\$
pdflatex-PGL_3_2_singer.1_poset.tex
open-PGL_3_2_singer.1_poset.pdf

PGL_3_2_on_lines:
$(ORBITER_PATH)orbiter.out-v.3\$
-orbit_path=$(ORBITER_PATH)\$
-define-G-linear_group-PGL_3_2-on_k_subspaces.2-end\$
-with-G-do\$
group_theoretic_activity\$
-poset_classification_control\$
-problem_label-PGL_3_2_lines-W-depth.7\$
-draw_poset\$
-report-end\$
-end\$
-orbits_on_subsets.7\$
-report\$
-end
PGL_2.5_on_subsets:
$\text{\$(ORBITER\_PATH)orbiter.out\-v.10\$
\text{\-orbiter_path\$(ORBITER\_PATH)\$
\text{\-define\textit{G}\-linear\_group\-PGL\_2.5\-end\$
\text{\-with\textit{G}\-do\$
\text{\-group_theoretic\_activity\$
\text{\-poset\_classification\_control\$
\text{\-problem\_label\textit{PGL}\_2.5\-\textit{W}\-depth\_6\$
\text{\-draw\_poset\$
\text{\-draw\_options\-radius\_200\-end\$
\text{\-report\-end\$
\text{\-end\$
\text{\-orbits\_on\_subsets\_6\$
\text{\-report\$
\text{\-end\$
\text{\-end\$
\text{\textit{pdflatex}\textit{PGL}\_2.5\_poset.tex$
\text{\textit{open-PGL}\_2.5\_poset.pdf$
\text{\textit{PGL}\_2.7_on_subsets:
$\text{\$(ORBITER\_PATH)orbiter.out\-v.10\$
\text{\-orbiter_path\$(ORBITER\_PATH)\$
\text{\-define\textit{G}\-linear\_group\-PGL\_2.7\-end\$
\text{\-with\textit{G}\-do\$
\text{\-group_theoretic\_activity\$
\text{\-poset\_classification\_control\$
\text{\-problem\_label\textit{PGL}\_2.7\-\textit{W}\-depth\_8\$
\text{\-draw\_poset\$
\text{\-draw\_options\-radius\_200\-end\$
\text{\-report\-end\$
\text{\-end\$
\text{\-orbits\_on\_subsets\_8\$
\text{\-report\$
\text{\-end\$
\text{\textit{pdflatex}\textit{PGL}\_2.7\_poset.tex$
\text{\textit{open-PGL}\_2.7\_poset.pdf$
\text{\textit{PGL}\_2.8_on_subsets:
$\text{\$(ORBITER\_PATH)orbiter.out\-v.10\$
\text{\-orbiter_path\$(ORBITER\_PATH)\$
\text{\-define\textit{G}\-linear\_group\-PGGL\_2.8\-end\$
\text{\-with\textit{G}\-do\$
\text{\-group_theoretic\_activity\$
\text{\textit{pdflatex}\textit{PGGL}\_2.8\_poset.tex$
\text{\textit{open-PGL}\_2.8\_poset.pdf$
5438  ▷ ▷ ▷ -poset_classification_control\ 
5439  ▷ ▷ ▷ ▷ -problem_label:PGGL_2.8:-W:-depth:9\ 
5440  ▷ ▷ ▷ ▷ -draw_poset: \ 
5441  ▷ ▷ ▷ ▷ -draw_options:-radius:200:-end \ 
5442  ▷ ▷ ▷ ▷ -report:-end \ 
5443  ▷ ▷ ▷ -end: \ 
5444  ▷ ▷ ▷ -orbits_on_subsets:9 \ 
5445  ▷ ▷ ▷ -report: \ 
5446  ▷ ▷ -end \ 
5447  ▷ pdflatex:PGGL_2.8_poset.tex \ 
5448  ▷ open:PGGL_2.8_poset.pdf \ 
5449  
5450 5451  PGGL_2.9.on_subsets: \ 
5452  ▷ $(ORBITER_PATH)orbiter.out:-v:10\ 
5453  ▷ -orbiter_path:$(ORBITER_PATH)\ 
5454  ▷ -define-G:-linear_group:-PGGL_2.9:-end\ 
5455  ▷ -with-G:-do\ 
5456  ▷ ▷ -group_theoretic_activity: \ 
5457  ▷ ▷ ▷ -poset_classification_control\ 
5458  ▷ ▷ ▷ ▷ -problem_label:PGGL_2.9:-W:-depth:10\ 
5459  ▷ ▷ ▷ ▷ -draw_poset:\ 
5460  ▷ ▷ ▷ ▷ -draw_options:-radius:200:-end\ 
5461  ▷ ▷ ▷ ▷ -report:-end\ 
5462  ▷ ▷ ▷ -end: \ 
5463  ▷ ▷ ▷ -orbits_on_subsets:10 \ 
5464  ▷ ▷ ▷ -report: \ 
5465  ▷ ▷ -end \ 
5466  ▷ pdflatex:PGGL_2.9_poset.tex \ 
5467  ▷ open:PGGL_2.9_poset.pdf \ 
5468  
5469 5470  PGGL_2.11.on_subsets: \ 
5471  ▷ $(ORBITER_PATH)orbiter.out:-v:10\ 
5472  ▷ -orbiter_path:$(ORBITER_PATH)\ 
5473  ▷ -define-G:-linear_group:-PGGL_2.11:-end\ 
5474  ▷ -with-G:-do\ 
5475  ▷ ▷ -group_theoretic_activity: \ 
5476  ▷ ▷ ▷ -poset_classification_control\ 
5477  ▷ ▷ ▷ ▷ -problem_label:PGGL_2.11:-W:-depth:12\ 
5478  ▷ ▷ ▷ ▷ -draw_poset:\ 
5479  ▷ ▷ ▷ ▷ -draw_options:-radius:200:-end\ 
5480  ▷ ▷ ▷ ▷ -report:-end\ 
5481  ▷ ▷ ▷ -end: \ 
5482  ▷ ▷ ▷ -orbits_on_subsets:12 \ 
5483  ▷ ▷ ▷ -report: \ 
5484  ▷ ▷ -end
5485 \> \texttt{pdflatex-PGL.2.11_poset.tex}
5486 \> \texttt{open-PGL.2.11_poset.pdf}
5487
5488
5489
5490 \texttt{PGGL.2.16_on_subsets:}
5491 \> \$(\texttt{ORBITER}\_\texttt{PATH})\texttt{orbiter.out}\texttt{-v.3:\}
5492 \> \> \texttt{-orbiter_path}\texttt{$(\texttt{ORBITER}\_\texttt{PATH})\}\}
5493 \> \> \> \texttt{-define}\texttt{G}\texttt{-linear}\texttt{_group}\texttt{-PGGL.2.16-end}\texttt{\}
5494 \> \> \> \texttt{-with}\texttt{G}\texttt{-do}\texttt{\}
5495 \> \> \> \texttt{-group_theoretic_activity}\texttt{\}
5496 \> \> \> \texttt{-poset_classification_control}\texttt{\}
5497 \> \> \> \> \texttt{-problem_label}\texttt{-PGGL.2.16-}\texttt{-W-}\texttt{-depth.10-}\texttt{\}
5498 \> \> \> \> \texttt{-draw_poset-}\texttt{\}
5499 \> \> \> \> \texttt{-report-}\texttt{-end}\texttt{\}
5500 \> \> \> \> \> \texttt{-end}\texttt{\}
5501 \> \> \> \> \> \texttt{-orbits_on_subsets.10-}\texttt{\}
5502 \> \> \> \> \> \> \texttt{-report-}\texttt{\}
5503 \> \> \> \> \> \> \> \texttt{-end}\texttt{\}
5504 \> \> \texttt{pdflatex-PGL.2.16_poset.tex}
5505 \> \texttt{open-PGL.2.16_poset.pdf}
5506
5507
5508 \texttt{PGGL.2.32_on_subsets:}
5509 \> \$(\texttt{ORBITER}\_\texttt{PATH})\texttt{orbiter.out}\texttt{-v.3:\}
5510 \> \> \texttt{-orbiter_path}\texttt{$(\texttt{ORBITER}\_\texttt{PATH})\}\}
5511 \> \> \> \texttt{-define}\texttt{G}\texttt{-linear}\texttt{_group}\texttt{-PGGL.2.32-end}\texttt{\}
5512 \> \> \> \texttt{-with}\texttt{G}\texttt{-do}\texttt{\}
5513 \> \> \> \texttt{-group_theoretic_activity}\texttt{\}
5514 \> \> \> \> \texttt{-poset_classification_control}\texttt{\}
5515 \> \> \> \> \> \texttt{-problem_label}\texttt{-PGGL.2.32-}\texttt{-W-}\texttt{-depth.8-}\texttt{\}
5516 \> \> \> \> \> \> \texttt{-draw_poset-}\texttt{\}
5517 \> \> \> \> \> \> \> \texttt{-report-}\texttt{-end}\texttt{\}
5518 \> \> \> \> \> \> \> \> \texttt{-end}\texttt{\}
5519 \> \> \> \> \> \> \> \> \> \texttt{-orbits_on_subsets.8-}\texttt{\}
5520 \> \> \> \> \> \> \> \> \> \> \texttt{-report-}\texttt{\}
5521 \> \> \> \> \> \> \> \> \> \> \> \texttt{-end}\texttt{\}
5522 \> \> \texttt{pdflatex-PGL.2.32_poset.tex}
5523 \> \texttt{open-PGL.2.32_poset.pdf}
5524
5525
5526 \texttt{PG.3.4_subsets:}
5527 \> \$(\texttt{ORBITER}\_\texttt{PATH})\texttt{orbiter.out}\texttt{-v.3:\}
5528 \> \> \texttt{-orbiter_path}\texttt{$(\texttt{ORBITER}\_\texttt{PATH})\}\}
5529 \> \> \> \texttt{-define}\texttt{G}\texttt{-linear}\texttt{_group}\texttt{-PGGL.4.4-end}\texttt{\}
5530 \> \> \> \texttt{-with}\texttt{G}\texttt{-do}\texttt{\}
5531 \> \> \> \texttt{-group_theoretic_activity}\texttt{\}
549
5532 -poset_classification_control\ 
5533 -problem_label:PGGL_4.4\ 
5534 -depth:5\ 
5535 -draw_poset\ 
5536 -draw_options\ 
5537 -radius:200\ 
5538 -end\ 
5539 -report\-end\ 
5540 -end\ 
5541 -orbits_on_subsets:5\ 
5542 -report\ 
5543 -end\ 
5544 pdflatex PGGL_4.4_poset.tex 
5545 open PGGL_4.4_poset.pdf 
5546 
5547 PGGL_2.9_orbits: 
5548 $(\text{ORBITER\_PATH})\text{orbiter.out}-v.3\ 
5549 \text{-orbiter_path:$(\text{ORBITER\_PATH})}\ 
5550 \text{-define G-\text{linear\_group}-PGGL_2.9-end}\ 
5552 \text{-with G-do}\ 
5553 -group_theoretic_activity\ 
5554 -poset_classification_control\ 
5555 -problem_label:PGGL_2.9-W-depth:5\ 
5556 -report\-end\ 
5557 -draw_poset\ 
5558 -draw_options--radius:200-end\ 
5559 -end\ 
5560 -orbits_on_subsets:5\ 
5561 -report\ 
5562 -end\ 
5563 pdflatex PGGL_2.9_poset.tex 
5564 open PGGL_2.9_poset.pdf 
5565 
5566 PG_3.7_lines: 
5568 $(\text{ORBITER\_PATH})\text{orbiter.out}-v.7\ 
5569 \text{-orbiter_path:$(\text{ORBITER\_PATH})}\ 
5570 \text{-define F-\text{finite\_field}-q.7-end}\ 
5571 \text{-define G-\text{linear\_group}-PGL_4-F-on_k\_subspaces:2-end}\ 
5572 \text{-with G-do}\ 
5573 -group_theoretic_activity\ 
5574 -poset_classification_control\ 
5575 -problem_label:7_lines\ 
5576 -depth:3-report\-end\ 
5577 -draw_poset\ 
5578 -draw_options--radius:200-embedded-end\ 
5579 pdflatex PGGL_4.4_poset.tex 
5580 open PGGL_4.4_poset.pdf
Section 6.4: Orbits on Subspaces

SECTION ORBITS ON SUBSPACES:

PGO_5_2_on_subsets:

PGO_5_2_on_subsets:

subspaces_Op_4_2:

subspaces_Op_4_2:
PGL\_4\_2 on subspaces:
\begin{verbatim}
$(ORBITER_PATH)orbiter.out-v.5\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-singer-1-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-singer-1-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-singer-1-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-singer-1-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-singer-1-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}

\begin{verbatim}
-define-G-linear_group-PGL\_4\_2-singer-1-end\end{verbatim}

\begin{verbatim}
-with-G-do\end{verbatim}

\begin{verbatim}
-launch\end{verbatim}
PGL₂ Singer on subspaces:

$(\text{ORBITER \ PATH})$orbiter.out -v 5

- orbiter_path$(\text{ORBITER \ PATH})$

-define G -linear_group-PGL₂ -singer 1-end

-with G -do

-group_theoretic_activity

- node_label_is_element

-draw_poset

-draw_options -radius 150-end

-problem_label-PGL₂ -singer

-W-depth 8-report-end

-end

-orbits on subspaces 8

-report

-end

-Orbits on subspaces: 8

-report

-end

Op_6₂ orbits on subspaces:

$(\text{ORBITER \ PATH})$orbiter.out -v 5

- orbiter_path$(\text{ORBITER \ PATH})$

-define G -linear_group-PGL₂ -orthogonal 1-end

-with G -do

-group_theoretic_activity

- node_label_is_element

-draw_poset

-draw_options -radius 200-end

-problem_label-Op_6₂ -W

-depth 6-report-end

-end

-orbits on subspaces 6

-report

-end

Op_6₂ orbits on subspaces: 6

-report

-end

Op_6₂ orbits on subspaces: 6

-report

-end

Op_6₂ orbits on subspaces: 6

-report

-end

Op_6₂ orbits on subspaces: 6

-report

-end

Op_6₂ orbits on subspaces: 6
Op₆₃ orbits on subspaces:

```
$(ORBITER_PATH)orbiter.out -v 5
```

```
-orbiter_path$(ORBITER_PATH)
```

```
-define G-linear_group-PGL₆₃-orthogonal 1-end
```

```
-with G-do
```

```
-group_theoretic_activity
```

```
-define
```

```
-G-linear_group-PGL₆₃-orthogonal 1-end
```

```
-with G-do
```

```
-group_theoretic_activity
```

```
-define
```

```
-G-linear_group-PGL₆₃-orthogonal 1-end
```

```
-with G-do
```

```
-node_label_is_element
```

```
-draw_poset
```

```
-draw_options -radius 200 -end
```

```
-problem_label Op₆₃-W
```

```
-depth 6-report -end
```

```
-end
```

```
pdflatex PGL₆₃_Orthogonal_plus_6₃_report.tex
```

```
open PGL₆₃_Orthogonal_plus_6₃_report.pdf
```

# June 3, 2020 on Mac: 0.0 sec

Op₆₁₁ orbits on subspaces:

```
$(ORBITER_PATH)orbiter.out -v 5
```

```
-orbiter_path$(ORBITER_PATH)
```

```
-draw_options -nodes_empty -end
```

```
-define G-linear_group-PGL₆₁₁-orthogonal 1-end
```

```
-with G-do
```

```
-group_theoretic_activity
```

```
-define
```

```
-G-linear_group-PGL₆₁₁-orthogonal 1-end
```

```
-with G-do
```

```
-node_label_is_element
```

```
-draw_poset
```

```
-draw_options -radius 200 -end
```

```
-problem_label Op₆₁₁-W
```

```
-depth 6-report -end
```

```
-end
```

```
pdflatex PGL₆₁₁_Orthogonal_plus_6₁₁_report.tex
```

```
open PGL₆₁₁_Orthogonal_plus_6₁₁_report.pdf
```

```
5767 # June 3, 2020 on Mac: 12 sec
5768
5769
5770 0p_8_2_orbits_on_subspaces:
5771  ▶ $(ORBITER_PATH)orbiter.out -v 5
5772  ▶ -orbiter_path $(ORBITER_PATH)
5773  ▶ -define G -linear_group -PGL 8 2 -orthogonal 1 -end
5774  ▶ -with G -do
5775  ▶ -group_theoretic_activity
5776  ▶ -poset_classification_control
5777  ▶ -node_label_is_element
5778  ▶ -draw_poset -draw_options -radius 200 -end
5779  ▶ -problem_label 0p_8_2 -W -depth 8 -report -end
5780  ▶ -end
5781  ▶ -orbits_on_subspaces 8
5782  ▶ -report
5783  ▶ -end
5784  ▶ pdflatex PGL_8_2_Orthogonal_plus_8_2_poset.tex
5785  ▶ open PGL_8_2_Orthogonal_plus_8_2_poset.pdf
5786
5787
5788
5789  PGO 7 2_on_subspaces:
5790  ▶ $(ORBITER_PATH)orbiter.out -v 20
5791  ▶ -orbiter_path $(ORBITER_PATH)
5792  ▶ -define F -finite_field -q 2 -end
5793  ▶ -define G -linear_group -PGL 7 F -orthogonal 0 -end
5794  ▶ -with G -do
5795  ▶ -group_theoretic_activity
5796  ▶ -poset_classification_control
5797  ▶ -node_label_is_element
5798  ▶ -draw_poset
5799  ▶ -draw_options -radius 200 -end
5800  ▶ -problem_label 0 7 2
5801  ▶ -W -depth 7
5802  ▶ -report -end
5803  ▶ -end
5804  ▶ -orbits_on_subspaces 7
5805  ▶ -report
5806  ▶ -end
5807  ▶ pdflatex PGL_7_2_Orthogonal_7_2_poset.tex
5808  ▶ open PGL_7_2_Orthogonal_7_2_poset.pdf
5809
5810
5811
5812
5813

555
SECTION ARCS AND CAPS IN PROJECTIVE SPACES:

PGL_3_27:
$(ORBITER_PATH)orbiter.out-v.5$

-define-G-
-linear_group-PGL_3_27-end
-with-G-do-
-group.theoretic_activity-
-report-
-end

pdflatex-PGL_3_27_report.tex
open-PGL_3_27_report.pdf

AGGL_2_27:
$(ORBITER_PATH)orbiter.out-v.5$

-define-G-
-linear_group-AGGL_2_27-end
-with-G-do-
-group.theoretic_activity-
-report-
-end

pdflatex-AGGL_2_27_report.tex
open-AGGL_2_27_report.pdf

hyperoval_4_classify:
$(ORBITER_PATH)orbiter.out-v.4$

-define-F-finite_field-q.4-end

-define-P-projective_space-2-F-end

-with-P-do-

-projective_space_activity-

-classify_arcs-

-poset_classification_control-

-problem_label-hyperoval_q4-

-W-depth-6-

-report-end-

-end-

-target_size-6-
5861 \> \> \> \> -d.2:\
5862 \> \> \> \> -end:\
5863 \> \> -end
5864 \> pdflatex hyperoval_q4_poset.tex
5865 \> open-hyperoval_q4_poset.pdf
5866
5867
5868
5869
5870 hyperoval_8classify:
5871 \> $(ORBITER_PATH)orbiter.out-v.4:\
5872 \> -orbiter_path$(ORBITER_PATH)\n
5873 \> \> -define F-finite_field-q.8-end\n
5874 \> \> -define F-projective_space-2-F-end\n
5875 \> \> -with P-do\n
5876 \> \> -projective_space_activity-\n
5877 \> \> \> -classify_arcs-\n
5878 \> \> \> \> -poset_classification_control-\n
5879 \> \> \> \> \> -problem_label-hyperoval_q8-\n
5880 \> \> \> \> \> -W-depth.10-\n
5881 \> \> \> \> \> -report-end-\n
5882 \> \> \> \> \> -draw_poset-\n
5883 \> \> \> \> \> -draw_options-\n
5884 \> \> \> \> \> \> -radius.200-\n
5885 \> \> \> \> \> \> -end-\n
5886 \> \> \> \> -end-\n
5887 \> \> \> -target_size.10-\n
5888 \> \> \> -d.2-\n
5889 \> \> -end\n
5890 \> -end

5891 \> pdflatex hyperoval_q8_poset.tex

5892 \> open-hyperoval_q8_poset.pdf

5893

5894

5895

5896

5897 frame_stabilizer_PGGL:
5898 \> $(ORBITER_PATH)orbiter.out-v.4-\n
5899 \> \> -define G-\n
5900 \> \> -linear_group-PGGL-3-8-end-\n
5901 \> \> -with G-do-\n
5902 \> \> -group_theoretic_activity-\n
5903 \> \> \> -poset_classification_control-\n
5904 \> \> \> \> -problem_label-frame_q8-W-depth.4-\n
5905 \> \> \> \> \> -draw_options-end-\n
5906 \> \> \> \> \> \> -report-end-\n
5907 \> \> \> \> \> \> -end-\n
557
frame_stabilizer_PGL:
$\{(\text{ORBITER PATH})\text{orbiter.out}-v.4\}$
 DEFINE-G-
 LINEAR_GROUP-PGL.3.8-END-
 WITH-G-DO-
 GROUP_THEORETIC_ACTIVITY-
 ORBITER_PATH-$(\text{ORBITER PATH})-
 DEFINE-F-FINITE_FIELD-Q.16-END-
 DEFINE-P-PROJECTIVE_SPACE.2.F-END-
 WITH-P-DO-
 PROJECTIVE_SPACE_ACTIVITY-
 CLASSIFY_ARCS-
 TARGET_SIZE.4-
 Q.8-
 N.3-
 D.2-
 END-
 -end
 pdflatex-frame_q8_poset.tex
 open-frame_q8_poset.pdf

hyperoval.16_classify:
$\{(\text{ORBITER PATH})\text{orbiter.out}-v.4\}$
 ORBITER_PATH-$\{(\text{ORBITER PATH})\}$
 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-FRAME_Q8-W-DEPTH.4-
 DRAW_OPTIONS-END-
 REPORT-END-
 END-
 pdflatex-frame_q8_poset.tex
 open-frame_q8_poset.pdf
We found 17028 non-conical 6-subsets

Eckardt point number distribution: $\{13^{252}, 9^{720}, 5^{2304}, 3^{12}\}$
hyperoval_{16,2} \text{ nonconical type:}

\begin{verbatim}
 hyperoval_{16,2}.nonconical_type:
\$ (ORBITER_PATH) orbiter.out -v 2 \\
\$ -define F = finite_field -q 16 -end \\
\$ -define P = projective_space -2 -F -end \\
\$ -define H = geometric_object -P \\
\$ -set $(HYPERVAL_{16,16320}) -end \\
\$ -with H =16 -2 -do \\
\$ -combinatorial_object_activity \\
\$ -save \\
\$ -end \\
\$ -with H =16 -2 -do \\
\$ -combinatorial_object_activity \\
\$ -non_conical_type \\
\$ -end \\
\$ -print_symbols
\end{verbatim}

\#We found 6188 = \binom{17}{5} \text{ non-conical 6-subsets}

\#Eckardt point number distribution: \$45^68, 13^2040, 5^4080\$

\#neighbors of 0 with 4 removed.csv

\#Row, C0, C1, C2, C3

\# 0, 2, 3, 9, 10
\# 1, 1, 3, 9, 7, 8
\# 2, 10, 12, 13, 15
\# 3, 1, 5, 10, 11
\# 4, 3, 6, 13
\# 5, 8, 9, 11, 12
\# 6, 11, 13, 17
\# 7, 10, 14, 16
\# 8, 1, 13, 16
\# 9, 2, 13, 14
\# 10, 2, 15, 17
\# 11, 6, 10, 17
\# 12, 6, 9, 15
\# 13, 2, 11, 16
\# 14, 5, 14, 17
\# 15, 5, 15, 16
\# 16, 1, 12, 14
\# 17, 2, 5, 12
\# 18, 3, 12, 16, 17
\# 19, 3, 11, 15
\#END
hyperoval

$\text{(ORBITER\_PATH)}\text{orbiter.out}\cdot-v\cdot2\cdot$

-define\ G\ -graph\ -disjoint\_sets\_graph\$
-neighbors\_of\_0\_with\_4\_removed\_\cdot\_csv\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$

-\text{-end}\cdot$
6095  ▶ $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot 4:\$
6096  ▶ ▶ -define\cdot F\cdot-\text{finite\_field}\cdot-\text{q16}\cdot-\text{end}\cdot$
6097  ▶ ▶ -define\cdot P\cdot-\text{projective\_space}\cdot 2\cdot F\cdot-\text{end}\cdot$
6098  ▶ ▶ -with\cdot P\cdot-\text{do}\cdot$
6099  ▶ ▶ -\text{projective\_space\_activity}\cdot$
6100  ▶ ▶ ▶ -\text{classify\_arcs}\cdot$
6101  ▶ ▶ ▶ ▶ -\text{poset\_classification\_control}\cdot$
6102  ▶ ▶ ▶ ▶ ▶ -\text{problem\_label}\cdot\text{nc\_arcs\_q16\_d2}\cdot$
6103  ▶ ▶ ▶ ▶ ▶ ▶ -W\cdot-\text{depth}\cdot 6\cdot$
6104  ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{report}\cdot-\text{end}\cdot$
6105  ▶ ▶ ▶ ▶ ▶ ▶ -\text{end}\cdot$
6106  ▶ ▶ ▶ ▶ ▶ -\text{target}\cdot \text{size}\cdot 6\cdot$
6107  ▶ ▶ ▶ ▶ ▶ ▶ -d\cdot 2\cdot$
6108  ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{conic\_test}\cdot$
6109  ▶ ▶ ▶ ▶ ▶ ▶ -\text{end}\cdot$
6110  ▶ ▶ ▶ ▶ -\text{end}\cdot$
6111  ▶ ▶ \text{pdflatex\_nc\_arcs\_q16\_d2\_poset.tex}
6112  ▶ ▶ open\cdot\text{nc\_arcs\_q16\_d2\_poset.pdf}
6113
6114
6115  ▶ #User\cdot\text{time}:: 0\cdot 00
6116
6117
6118
6119  ▶ \text{nc\_arcs\_32\_E13:}
6120  ▶ ▶ $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\cdot 4:\$
6121  ▶ ▶ ▶ -\text{orbiter\_path}\cdot $(\text{ORBITER\_PATH})\cdot$
6122  ▶ ▶ ▶ -define\cdot F\cdot-\text{finite\_field}\cdot-\text{q32}\cdot-\text{end}\cdot$
6123  ▶ ▶ ▶ -define\cdot P\cdot-\text{projective\_space}\cdot 2\cdot F\cdot-\text{end}\cdot$
6124  ▶ ▶ ▶ -with\cdot P\cdot-\text{do}\cdot$
6125  ▶ ▶ ▶ -\text{projective\_space\_activity}\cdot$
6126  ▶ ▶ ▶ ▶ -\text{classify\_arcs}\cdot$
6127  ▶ ▶ ▶ ▶ ▶ -\text{poset\_classification\_control}\cdot$
6128  ▶ ▶ ▶ ▶ ▶ ▶ -\text{problem\_label}\cdot\text{nc\_arcs\_q32\_d2}\cdot$
6129  ▶ ▶ ▶ ▶ ▶ ▶ ▶ -W\cdot-\text{depth}\cdot 6\cdot$
6130  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{draw\_poset}\cdot-\text{draw\_options}\cdot-\text{end}\cdot$
6131  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{report}\cdot-\text{end}\cdot$
6132  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{end}\cdot$
6133  ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{target}\cdot \text{size}\cdot 6\cdot$
6134  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{test\_nb\_Eckardt\_points}\cdot 13\cdot$
6135  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -d\cdot 2\cdot$
6136  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{conic\_test}\cdot$
6137  ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{end}\cdot$
6138  ▶ ▶ ▶ ▶ -\text{end}\cdot$
6139  ▶ ▶ \text{pdflatex\_nc\_arcs\_q32\_d2\_poset.tex}
6140  ▶ ▶ open\cdot\text{nc\_arcs\_q32\_d2\_poset.pdf}
6141
562
#User.time: 0:00

6142

6143

6144

6145

6146

6147 F64_work:
6148 $#(ORBITER_PATH)orbiter.out -v.3\n6149 #define F -finite_field -q 64 -end \n6150 #define f -formula "f"."f"."a*a+a".\n6151 #with F do -finite_field_activity \n6152 #evaluate f: a=2 -end
6153
6154 F64_frob:
6155 $#(ORBITER_PATH)orbiter.out -v.3\n6156 #define F -finite_field -q 64 -end \n6157 #define f -formula "f"."f"."a*a*a*a*a*a*a".\n6158 #with F do -finite_field_activity \n6159 #evaluate f: a=61 -end
6160
6161
6162 #surfaces with 13 Eckardt points have OCN=0,98,99
6163
6164 surface_64_0:
6165 $#(ORBITER_PATH)orbiter.out -v.3\n6166 #define F -finite_field -q 64 -end \n6167 #define P -projective_space:3 F -end \n6168 #with P do \n6169 #projective_space_activity \n6170 #define_surface S -q 64 -catalogue 0 \n6171 #end \n6172 #end \n6173 #with S do \n6174 #cubic_surface_activity \n6175 #report \n6176 #report_with_group \n6177 #end
6178 pdfflatex surface_catalogue_q64_iso0_with_group.tex
6179 open surface_catalogue_q64_iso0_with_group.pdf
6180
6181
6182
6183
6184 #makes it slow:
6185 #test_nb_Eckardt_points:13 \n6186 #report: select_orbits_by_level:6:select_orbits_by_stabilizer_order_multiple_of:24 -end \n6187 #User.time: 0:3
nc_arcs_128:
$($ORBITER_PATH$)orbiter.out -v 4 \
  -define F:finite_field q:128 -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 q128 d2 -W:depth 6: \
  -report -select_orbits_by_level 6: \
  -select_orbits_by_stabilizer_order multiple_of 24: \
  -end: \
  -end: \
  -target_size 6: \
  -d: 2: \
  -conic_test: \
  -end: \
  -end \
  -test nb Eckardt points 13: \
  -d: 2: \
  -conic_test: \
  -end: \
  -end

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 -W:depth 6: \
  -report -end: \
  -target_size 6: \
  -test nb Eckardt points 13: \
  -d: 2: \
  -conic_test: \
  -end: \
  -end

#User time: 0:52
Example_F64:

```
$\text{(ORBITER_PATH)orbiter.out-v.3:\}
\text{-define-F\-finite_field\-q\-64\-end\}
\text{-define-P\-projective_space\-3\-F\-end\}
\text{-with-P\-do\}
\text{-projective_space\-activity\}
\text{-define_surface-S64\_abcd\-52\_8\_8\_52\-q\-64\}
\text{-family_general\_abcd\-52\_8\_8\_52\-end\}
\text{-with\-S64\_abcd\-52\_8\_8\_52\-do\}
\text{-cubic_surface\-activity\}
\text{-report\}
\text{-end\}
\text{pdflatex\-surface\_family\_general\_abcd\_q64\_a52\_b8\_c8\_d52\_report\_tex}
```

six_arcs_4_nbE13:

```
$\text{(ORBITER_PATH)orbiter.out-v.3:\}
\text{-define-F\-finite_field\-q\-4\-end\}
\text{-define-P\-projective_space\-2\-F\-end\}
\text{-with-P\-do\}
\text{-projective_space\-activity\}
\text{-control\_six_arcs\-problem\_label\_sixarcs\_q4\-end\}
\text{-six_arcs\_not\_on\_conic\-filter\_by\_nb\_Eckardt\_points\_13\-end}
```

six_arcs_8_nbE13:

```
$\text{(ORBITER_PATH)orbiter.out-v.3:\}
\text{-define-F\-finite_field\-q\-8\-end\}
\text{-define-P\-projective_space\-2\-F\-end\}
\text{-with-P\-do\}
\text{-projective_space\-activity\}
\text{-control\_six_arcs\-problem\_label\_sixarcs\_q8\-end\}
\text{-six_arcs\_not\_on\_conic\-filter\_by\_nb\_Eckardt\_points\_13\-end}
```
six_arcs_16_nbE13:
- $(ORBITER\ PATH)\ orbiter.out\ -v.3\$
- \define\ F\ -finite\ field\ -q.16\ -end\$
- \define\ P\ -projective\ space\ -2.F\ -end\$
- \with\ P\ -do\$
- \projective\ space\ activity\$
- \control\ six\ arcs\ -problem\ label\ -sixarcs.q16\ -end\$
- \six\ arcs\ not\ on\ conic\ -filter\ by\ nb\ Eckardt\ points\ 13\ -end\$

six_arcs_32_nbE13:
- $(ORBITER\ PATH)\ orbiter.out\ -v.3\$
- \define\ F\ -finite\ field\ -q.32\ -end\$
- \define\ P\ -projective\ space\ -2.F\ -end\$
- \with\ P\ -do\$
- \projective\ space\ activity\$
- \control\ six\ arcs\ -problem\ label\ -sixarcs.q32\ -end\$
- \six\ arcs\ not\ on\ conic\ -filter\ by\ nb\ Eckardt\ points\ 13\ -end\$

six_arcs_64_nbE13:
- $(ORBITER\ PATH)\ orbiter.out\ -v.3\$
- \define\ F\ -finite\ field\ -q.64\ -end\$
- \define\ P\ -projective\ space\ -2.F\ -end\$
- \with\ P\ -do\$
- \projective\ space\ activity\$
- \control\ six\ arcs\ -problem\ label\ -sixarcs.q64\ -end\$
- \six\ arcs\ not\ on\ conic\ -filter\ by\ nb\ Eckardt\ points\ 13\ -end\$

-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:
- $(ORBITER\ PATH)\ orbiter.out\ -v.3\$
- \define\ F\ -finite\ field\ -q.128\ -end\$
- \define\ P\ -projective\ space\ -2.F\ -end\$
- \with\ P\ -do\$
- \projective\ space\ activity\$
- \control\ six\ arcs\ -problem\ label\ -sixarcs.q128\ -end\$
- \six\ arcs\ not\ on\ conic\ -filter\ by\ nb\ Eckardt\ points\ 13\ -end\$

1-min 39.3 sec
12-arcs, ago: 4^3, 24^9

-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_256_nbE13:
- $(ORBITER\ PATH)\ orbiter.out\ -v.3\$

566
# 27 minutes on ripoff

# User-time::29:11 on ripoff 7/30/21

five_arcs.q13:

```
$ (ORBITER_PATH) orbiter.out -v 3
```

```
define F finite_field -q 3 -end
```

```
define P projective_space -2 F -end
```

```
with P do
```

```
control six_arcs -problem_label sixarcs_q3 -end
```

```
six_arcs_not_on_conic -filter by nb Eckardt_points -13 -end
```

```
six arcs not on conic filter by nb Eckardt points 13 -end
```

```
report -end
```

```
target size 5
```

```
d 2
```

```
end
```

```
pdflatex five_arcs_q13 poset.tex
```

```
open five_arcs_q13 poset.pdf
```

### Section 6.6: Cubic Curves

```
SECTION CUBIC CURVES:
```

cubic_curves_PG 2 4:
```
$ (ORBITER_PATH) orbiter.out -v 3
```

```
orbit_path $(ORBITER_PATH)
```

```
define F finite_field -q 3 -end
```

```
def F finite_field -q 256 -end
```

```
define P projective_space -2 F -end
```

```
with P do
```

```
control six_arcs -problem_label sixarcs_q256 -end
```

```
six_arcs_not_on_conic -filter by nb Eckardt_points -13 -end
```

```
six arcs not on conic filter by nb Eckardt points 13 -end
```

```
six arcs filter by nb Eckardt points 13 -end
```

```
report -end
```

```
target size 5
```

```
d 2
```

```
end
```

```
pdflatex six_arcs_q256 poset.tex
```

```
open six_arcs_q256 poset.pdf
```

567
\begin{verbatim}
6376  \texttt{-define-P-projective_space-2-F-end\}
6377  \texttt{-with-P-do\}
6378  \texttt{-projective_space_activity\}
6379  \texttt{-classify_cubic_curves-q-4\}
6380  \texttt{-target_size-9-n-3-d-3\}
6381  \texttt{-poset_classification_control\}
6382  \texttt{-problem_label-cc_4-W-depth-9\}
6383  \texttt{-draw_poset\}
6384  \texttt{-draw_options-radius-200-embedded-end\}
6385  \texttt{-report-end\}
6386  \texttt{-end\}
6387  \texttt{-end\}
6388  \texttt{pdflatex cc_4_poset.tex}
6389  \texttt{open-cc_4_poset.pdf}
6390  \texttt{pdflatex cc_4_poset_lvl_9.tex}
6391  \texttt{#open-cc_4_poset_lvl_9.pdf}
6392  \texttt{pdflatex Cubic_curves_q4.tex}
6393  \texttt{#open-Cubic_curves_q4.pdf}
6394
6395  \texttt{cubic_curves_PG_2_4.draw:}
6396  \texttt{$(ORBITER\_PATH)\texttt{orbiter.out-v.3\}
6397  \texttt{-draw_labeled_graph-cc_4_poset_lvl_9_labeled_graph\}
6398  \texttt{-radius-300-embedded-line_width-1.1\}
6399  \texttt{-y_stretch-0.9-scale-0.25\}
6400  \texttt{-paths_in_between-6.4-9.0\}
6401  \texttt{-end\}
6402  \texttt{pdflatex cc_4_poset_lvl_9_draw.tex}
6403  \texttt{open-cc_4_poset_lvl_9_draw.pdf}
6404
6406  \texttt{cubic_curves_PG_2_8:}
6407  \texttt{$(ORBITER\_PATH)\texttt{orbiter.out-v.3-define-G\}
6408  \texttt{-define-F-finite_field-q-8-end\}
6409  \texttt{-define-P-projective_space-2-F-end\}
6410  \texttt{-with-P-do\}
6411  \texttt{-projective_space_activity\}
6412  \texttt{-classify_cubic_curves-q-8-target_size-9-n-3-d-3\}
6413  \texttt{-poset_classification_control\}
6414  \texttt{-problem_label-cc_8-W-depth-9\}
6415  \texttt{-draw_options-radius-200-embedded-end\}
6416  \texttt{-recognize\"0,1,2,3,35,28\"\}
6417  \texttt{-recognize\"1,2,3,51,28,61,46,71,40\"\}
6418  \texttt{-draw_poset\}
6419  \texttt{-Kramer_Mesner_matrix-6.9\}
6420  \texttt{-end\}
6421  \texttt{-end\}
6422  \texttt{$(ORBITER\_PATH)\texttt{orbiter.out-v.2-draw_matrix\}
\end{verbatim}
cubic_curves_PG_28_draw:

\$ (ORBITER_PATH) orbiter.out -v 3 \$

\$ draw_layered_graph \$

\$ cc_8_poset_lvl_9.layered_graph \$

\$ radius 2 - embedded - line_width 0.01 \$

\$ y_stretch 1.3 - scale 0.5 \$

\$ paths_in_between 0.7 - 0.9 \$

\$ - end \$

\$ pdflatex cc_8_poset_lvl_9_draw.tex \$

\$ open cc_8_poset_lvl_9_draw.pdf \$

\$ cc_8_poset_lvl_9.layered_graph \$

\$ cc_8_poset_detailed_lvl_9.layered_graph \$

SECTION CUBIC SURFACES CREATION:

surface 4 0:

\$ (ORBITER_PATH) orbiter.out -v 3 \$
...
surface_8_0_clean:

$\text{(ORBITER\_PATH)}\text{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\{1,4,4,0,6,0,0,6,2,0,1,7,0,4,0,0\}-

-transform\{4,4,0,0,0,1,0,1,0,0,0,0,0,1,0\}-

-transform_inverse\{2,0,0,0,2,0,0,0,2,0,1,1,2,3,0\}-

-end-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:

$\text{(ORBITER\_PATH)}\text{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\{1,0,0,0,1,0,6,0,0,1,6,0,0,0,1,0\}-

-transform_inverse\{3,1,1,0,0,1,0,0,0,0,1,0,0,0,0,1,0\}-

-transform_inverse\{2,0,0,0,1,0,0,0,1,0,0,0,0,1,0\}-

-end-\
\input{group.tex}

\begin{verbatim}
Eckardt_13:
\$\text{(ORBITER\ PATH)orbiter.out:-v.3}\$
\$\text{-define F-finite_field-q.13-end}\$
\$\text{-define P-projective_space-3-F-end}\$
\$\text{-with P-do}\$
\$\text{-projective_space_activity}\$
\$\text{-define_surface_S_q13-q.13}\$
\$\text{-family_Eckardt.3.1-end}\$
\$\text{-end}\$
\$\text{-with S_q13-do}\$
\$\text{-cubic_surface_activity}\$
\$\text{-report}\$
\$\text{-report_with_group}\$
\$\text{-end}\$
\end{verbatim}

\begin{verbatim}
surface_13.0:
\$\text{(ORBITER\ PATH)orbiter.out:-v.3}\$
\$\text{-define F-finite_field-q.13-end}\$
\$\text{-define P-projective_space-3-F-end}\$
\$\text{-with P-do}\$
\end{verbatim}
-projective_space_activity\ 
-define_surface-S13_0-q13-catalogue-0-end\ 
-end\ 
-with-S13_0-do-\ 
cubic_surface_activity\ 
-report-\ 
-report_with_group-\ 
end-\ 
pdflatex-surface_catalogue_q13_iso0_report.tex\ 
open-surface_catalogue_q13_iso0_report.pdf\ 

surface_16_0:\ 
$(\text{ORBITER}\_PATH)\text{orbiter.out} -v.3-\ 
-define-F-finite_field-q16-end-\ 
-define-F-projective_space-3F-end-\ 
-with-F-do-\ 
-projective_space_activity-\ 
-define_surface-S16_0-q16-catalogue-0-\ 
-transform-"1,0,0,0,1,0,12,0,0,1,12,0,0,0,1,0"-\ 
-transform-"15,11,4,0,0,12,0,0,12,0,0,0,0,1,3"-\ 
-end-\ 
end-\ 
-with-S16_0-do-\ 
cubic_surface_activity-\ 
-report-\ 
-report_with_group-\ 
end-\ 
pdflatex-surface_catalogue_q16_iso0_with_group.tex\ 
open-surface_catalogue_q16_iso0_with_group.pdf\ 

# clean equation for Tekirdag-2:\ 

surface_16_0:\ 
$(\text{ORBITER}\_PATH)\text{orbiter.out} -v.3-\ 
-define-F-finite_field-q16-end-\ 
-define-F-projective_space-3F-end-\ 
-with-F-do-\ 
-projective_space_activity-\ 
-define_surface-S16_0-q16-catalogue-0-\ 
-transform-"1,0,0,0,1,0,12,0,0,1,12,0,0,0,1,0"-\ 
-transform-"15,11,4,0,0,12,0,0,12,0,0,0,0,1,3"-\ 
-end-\ 
end-\ 
-with-S16_0-do-\ 
cubic_surface_activity-\ 
-report-\ 
-report_with_group-\ 
end-\ 
pdflatex-surface_catalogue_q16_iso0_with_group.tex\ 
open-surface_catalogue_q16_iso0_with_group.pdf\ 

# transform inverse:3,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0.\ 
# transform inverse:13,12,1,0,12,13,1,0,0,0,1,0,0,0,1,0.\ 
# transform inverse:1,0,0,0,0,1,0,0,12,12,1,0,0,0,0,1,0.\ 
# transform inverse:12,0,0,0,12,0,0,0,0,1,0,0,0,1,0.\ 

# rank of lines: (-66591, -26737, -4093, -69904, -28376, -26470, -70160, -69855, -26208, -26470, -847, -369, -72430, -529, -30293, -70068, -2178, -48736, -25595, -25209, -22193, -49862, -274)\ 

# Rank of points on Klein quadric: (-29181, -4677, -29950, 33, -62496, 429, -1, 9205, -37, -29964, -29364, -21501, -4656, -54735, -5425, -30105, -754, -6680, -13354, -758, -30106, -0, -29209, -48736, -25595, -33780, -105, -31694, -0, -51784, -0, -29209, -48736, -25595, -33780, -4657)\ 

574
Tekirdag-1:

G13_8:

F13_8:

Tekirdag-2:
\-define\text{-}P\-\text{-}projective\text{-}space\text{-}3\text{-}F\-\text{-}end\textbackslash
\-define\text{-}T2\-\text{-}family\text{-}F13\text{-}2\-\text{-}q16\-\text{-}end\textbackslash
\-define\text{-}surface\text{-}T2\-\text{-}family\text{-}F13\text{-}2\-\text{-}q\text{-}16\textbackslash
\-end\textbackslash
\-define\text{-}T3\-\text{-}family\text{-}F13\text{-}2\-\text{-}q32\textbackslash
\-end\textbackslash
\-define\text{-}K1\-\text{-}family\text{-}F13\text{-}2\-\text{-}q64\textbackslash
\-end\textbackslash
\-define\text{-}P\-\text{-}projective\text{-}space\text{-}3\text{-}F\-\text{-}end\textbackslash
\-define\text{-}T2\-\text{-}do\textbackslash
\-cubic\text{-}surface\text{-}activity\textbackslash
\-report\textbackslash
\-report\text{-}with\text{-}group\textbackslash
\-end\textbackslash
\-define\text{-}F\-\text{-}finite\text{-}field\text{-}q32\-\text{-}end\textbackslash
\-define\text{-}P\-\text{-}projective\text{-}space\text{-}3\text{-}F\-\text{-}end\textbackslash
\-define\text{-}T2\-\text{-}do\textbackslash
\-cubic\text{-}surface\text{-}activity\textbackslash
\-report\textbackslash
\-report\text{-}with\text{-}group\textbackslash
\-end\textbackslash
\-define\text{-}F\-\text{-}finite\text{-}field\text{-}q64\-\text{-}end\textbackslash
\-define\text{-}P\-\text{-}projective\text{-}space\text{-}3\text{-}F\-\text{-}end\textbackslash
\-define\text{-}T1\-\text{-}do\textbackslash
\-cubic\text{-}surface\text{-}activity\textbackslash
\-report\textbackslash
\-report\text{-}with\text{-}group\textbackslash
\-end\textbackslash
\-define\text{-}K1\-\text{-}family\text{-}F13\text{-}2\-\text{-}q64\textbackslash
\-end\textbackslash
\-define\text{-}P\-\text{-}projective\text{-}space\text{-}3\text{-}F\-\text{-}end\textbackslash
\-define\text{-}T2\-\text{-}do\textbackslash
\-cubic\text{-}surface\text{-}activity\textbackslash
\-report\textbackslash
\-report\text{-}with\text{-}group\textbackslash
\-end\textbackslash
\-define\text{-}T3\-\text{-}family\text{-}F13\text{-}2\-\text{-}q32\textbackslash
\-end\textbackslash
\-define\text{-}K1\-\text{-}family\text{-}F13\text{-}2\-\text{-}q64\textbackslash
\-end\textbackslash
\-define\text{-}P\-\text{-}projective\text{-}space\text{-}3\text{-}F\-\text{-}end\textbackslash
\-define\text{-}T2\-\text{-}do\textbackslash
\-cubic\text{-}surface\text{-}activity\textbackslash
\-report\textbackslash
\-report\text{-}with\text{-}group\textbackslash
\-end
Kapadokya-2:

F13_64b:

define F -finite field -q 64 -end

define P -projective space 3 F -end

with P -do

-projective_space_activity

define_surface K2 -family F13 18 -q 64 -end

-end

define P -projective space 3 F -end

with P -do

-projective_space_activity

define_surface K2 -family F13 18 -q 64 -end

-end

Colorado1:

define F -finite field -q 128 -end

define P -projective space 3 F -end

-with P -do

-projective_space_activity

define_surface CO-1 -q 128 -catalogue 0

-transform_inverse "1,0,0,0,1,0,96,0,0,1,96,0,0,1,0"

-end

-with CO-1 -do

-cubic_surface_activity

-report

-report_with_group

-end

Colorado2:

define F -finite field -q 128 -end

define P -projective space 3 F -end

-with P -do

-projective_space_activity

Colorado-1,2,3:

# recognize the arcs from Colorado-1,2,3:
define_surface CO-2 -q 128 -catalogue 926;
transform_inverse "1,0,0,0,1,0,32,0,0,1,32,0,0,0,1,0";
end;
end;
with CO-2 -do;
cubic_surface_activity;
end;
end;
with CO-2 -do;
cubic_surface_activity;
end;
end;
with CO-2 -do;
cubic_surface_activity;
end;
end;

Colorado3:

$(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 -q 128 -catalogue 928
transform_inverse "1,0,0,0,1,0,59,0,0,1,59,0,0,0,1,0"
end;
end;
with CO-3 -do;
cubic_surface_activity;
end;
end;
with CO-3 -do;
cubic_surface_activity;
end;
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;
end;
with CO-1 -do;
cubic_surface_activity
end;
end;
with CO-1 -do;
cubic_surface_activity
end;
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\-
-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 F13\-14\-q\-128\-end\-
-with CO\-3\-do\-
-cubic_surface_activity\-
-report\-
-report_with_group\-
-end

move_two_lines:
$(ORBITER\_PATH)orbiter.out\-v\-5\-
-define F\-finite_field\-q\-8\-end\-
-with F\-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\-
-define F\-finite_field\-q\-7\-end\-
-with F\-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

579
\documentclass{article}
\usepackage{amsmath}

\begin{document}

\section*{F_{abcd} Eckardt q31:}
\begin{verbatim}
\$ (ORBITER_PATH) orbiter.out -v 3 \\
define F - finite_field - q 31 - end \\
define P - projective_space - 3 F - end \\
with P - do \\
projective_space_activity \\
\end{verbatim}

\section*{F_{abcd} sweep 4 27 q7:}
\begin{verbatim}
\$ (ORBITER_PATH) orbiter.out -v 3 \\
define F - finite_field - q 7 - end \\
with F - do \\
finite_field_activity \\
parse_and_evaluate "Fabcd" "X0, X1, X2, X3" \\
\end{verbatim}

\end{document}
F_alpha_beta_gamma_delta_q7_overwrite_group:

\$(\text{ORBITER\_PATH})\text{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
 DEFINE_surface_F_2345 -q 7 \n
 -by_equation "F alpha beta gamma delta" \n
 DF {alpha, beta, gamma, delta}"x0,x1,x2,x3"\n $F_{\text{ALPHA,BETA,GAMMA,DELTA}}\$
 $\text{alpha}=2,\beta=3,\gamma=4,\delta=5$\n $\text{delta}=2,\beta=3,\gamma=4,\delta=5\text{D}$\n override_group 6-2\n override_group 5,0,0,3,6,0,0,1,1,3,0,5,5,0,3,\n 1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4\n -end \n -end \n -end \n-cubic_surface_activity \n-report \n-report_with_group \n-end

\text{pdflatex surface_equation_F_alpha_beta_gamma_delta_q7_report.tex}
\text{open surface_equation_F_alpha_beta_gamma_delta_q7_report.pdf}
\text{pdflatex surface_equation_F_alpha_beta_gamma_delta_q7_with_group.tex}
\text{open surface_equation_F_alpha_beta_gamma_delta_q7_with_group.pdf}

#cubic_surfaces_with_15_lines:

F_alpha_beta_gamma_delta_sweep_4_q3:

\$(\text{ORBITER\_PATH})\text{orbiter.out}-v.3\$
\n-define F -finite_field -q 3 -end \n-define P -projective_space -3 F -end \n-with P -do \n-projective_space_activity \n
# cubic surfaces with 15 lines:

F_\alpha_\beta_\gamma_\delta q7:

- by equation:
  \[ F(\alpha, \beta, \gamma, \delta, x0, x1, x2, x3) \]
  \[ \alpha=2, \beta=3, \gamma=4, \delta=5 \]

- define F - finite field q7:

- define P - projective space 3 F:

- with P - do projective space activity:

- classify surfaces with double sixes:

User time: 0:30

348 parameter sets

F_\alpha_\beta_\gamma_\delta q7_points.txt

F_\alpha_\beta_\gamma_\delta q7_sweep.csv

F_\alpha_\beta_\gamma_\delta q7_sweep4_15_data.csv

F_\alpha_\beta_\gamma_\delta q7.recognize:

- by equation:
  \[ F(\alpha, \beta, \gamma, \delta, x0, x1, x2, x3) \]
  \[ \alpha=2, \beta=3, \gamma=4, \delta=5 \]

- define F - finite field q49:

- define P - projective space 3 F:

- with P - do:

- classify surfaces with double sixes:

582
classification of cubic surfaces with double sixes activity
recognize
by equation "F_alpha_beta_gamma_delta".
"x0,x1,x2,x3".
$(F\_ALPHA\_BETA\_GAMMA\_DELTA)\$
"alpha=2, beta=1, gamma=1, delta=2".
"D\alpha=2, \beta=1, \gamma=1, \delta=2".
"alpha=2, beta=1, gamma=2, delta=2".
"D\alpha=2, \beta=1, \gamma=2, \delta=2".
"alpha=2, beta=1, gamma=5, delta=2".
"D\alpha=2, \beta=1, \gamma=5, \delta=2".
"alpha=2, beta=1, gamma=1, delta=3".
"D\alpha=2, \beta=1, \gamma=1, \delta=3".
> > > -end\n> > -end\n> > -with Surf.-do\n> > -classification_of_cubic_surfaces_with_double_sixes_activity\n> > -recognize\n> > -q.49\n> > -by_equation "F_alpha_beta_gamma_delta"\n> > "DF{\alpha,\beta,\gamma,\delta}\D"."x0,x1,x2,x3"\n> > $(F_ALPHA_BETA_GAMMA_DELTA)\n> > "alpha=2,\beta=1,\gamma=2,\delta=3"\n> > "\Dalpha=2,\beta=1,\gamma=2,\delta=3\D"\n> > -end\n> > -end\n> > -with Surf.-do\n> > -classification_of_cubic_surfaces_with_double_sixes_activity\n> > -recognize\n> > -q.49\n> > -by_equation "F_alpha_beta_gamma_delta"\n> > "DF{\alpha,\beta,\gamma,\delta}\D"."x0,x1,x2,x3"\n> > $(F_ALPHA_BETA_GAMMA_DELTA)\n> > "alpha=2,\beta=1,\gamma=4,\delta=3"\n> > "\Dalpha=2,\beta=1,\gamma=4,\delta=3\D"\n> > -end\n> > -end\n> > surf49_recognize:\n> > $(ORBITER_PATH)oribiter.out-v.3\n> > -define F- finite_field -q.49 -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\n> > -classification_of_cubic_surfaces_with_double_sixes_activity\n> > -recognize\n> > -q.49\n> > -by_coefficients "2,5,1,16,4,10,1,18,4,8,4,9,3,11,4,14"\n> > -end\n> > -end\n> > -print symbols
McKean_15lines_q7:

```bash
$(ORBITER_PATH)orbiter.out -v 3
define F -finite_field -q 7 -end
define P -projective_space 3 F -end
with P -do
  -projective_space_activity
end
end

define surface
  -by_coefficients $(SURFACE_MCKEAN_15_LINES) -q 7
end
end

#pdflatex -surface_by_coefficients_q7_report.tex
#open -surface_by_coefficients_q7_report.pdf

# 2-Eckardt points

F.4.4.3.3_q7:

$(ORBITER_PATH)orbiter.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_S
  -do
    -projective_space_activity
    -define_surface -q 7 -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:

$(ORBITER_PATH)orbiter.out -v 3
define F -finite_field -q 7 -end
define P -projective_space 3 F -end
with P -do
  -projective_space_activity
  -sweep nothing
```
-define_surface-q7-by_equation:\n"F_{\alpha,\beta,\gamma,\delta}\":\n"\Delta\{\alpha,\beta,\gamma,\delta}\Delta\":\n"x_0,x_1,x_2,x_3\":\n$(F_{\alpha,\beta,\gamma,\delta})\nF_{\alpha,\beta,\gamma,\delta}=$
"alpha=2,beta=3,\gamma=4,\delta=5\"$
"\Delta\alpha=2,\beta=3,\gamma=4,\delta=5\Delta\"$
end\nend\n
# Section 7.2: Cubic Surfaces and Quartic Curves.

SECTION_CUBIC_SURFACES_AND_QUARTIC_CURVES:

quartic_curve_9_0_report::
$\$(ORBITER\_PATH)oritzer.out-v.3\$
-define_F-finite_field-q9-end:\n-define_P-projective_space-2F-end:\n-with_P-do:\n-projective_space_activity:\n-define_quartic_curve_C-q9:\n-catalogue-0-end:\n-end:\n-with_C-do:\n-quartic_curve_activity:\n-report:\n-end:
pdflatex_quartic_curve_catalogue_q9_iso0_report.tex
open-quartic_curve_catalogue_q9_iso0_report.pdf

quartic_curve_13_0_report::
$\$(ORBITER\_PATH)oritzer.out-v.3\$
-define_F-finite_field-q13-end:\n-define_P-projective_space-2F-end:\n-with_P-do:\n-projective_space_activity:\n-define_quartic_curve_C-q13-catalogue-0-end:\n-end\n
7265 \> \> -with:C-do\>
7266 \> \> \> -quartic_curve_activity\>
7267 \> \> \> \> -report\>
7268 \> \> \> -end
7269 \> pdflatex-quartic_curve_catalogue_q13_iso0_report.tex
7270 \> open-quartic_curve_catalogue_q13_iso0_report.pdf
7271
7272
7273 quartic_curve_13_1_report::
7274 \> $(ORBITER_PATH)orbiter.out-v.3\>
7275 \> \> -define-F--finite_field-q.13-end\>
7276 \> \> -define-P--projective_space-2-F-end\>
7277 \> \> -with-P-do\>
7278 \> \> \> -projective_space_activity\>
7279 \> \> \> \> -define_quartic_curve:C--q.13--catalogue:1-end\>
7280 \> \> \> -end\>
7281 \> \> -with:C-do\>
7282 \> \> \> -quartic_curve_activity\>
7283 \> \> \> \> -report\>
7284 \> \> \> -end
7285 \> pdflatex-quartic_curve_catalogue_q13_iso1_report.tex
7286 \> open-quartic_curve_catalogue_q13_iso1_report.pdf
7287
7288
7289
7290 surface_4_0_quartic_curves:
7291 \> $(ORBITER_PATH)orbiter.out-v.3\>
7292 \> \> -define-F--finite_field-q.4-end\>
7293 \> \> -define-P--projective_space-3-F-end\>
7294 \> \> -with-P-do\>
7295 \> \> -projective_space_activity\>
7296 \> \> -define_surface:S4.0--q.4--catalogue:0-end\>
7297 \> \> -end\>
7298 \> \> -with:S4.0-do\>
7299 \> \> -cubic_surface_activity\>
7300 \> \> \> -report\>
7301 \> \> \> \> -report_with_group\>
7302 \> \> \> \> -all_quartic_curves\>
7303 \> \> \> -end
7304 \> pdfflatex-surface_catalogue_q4_iso0_report.tex
7305 \> open-surface_catalogue_q4_iso0_report.pdf
7306 \> pdfflatex-surface_catalogue_q4_iso0_with_group.tex
7307 \> open-surface_catalogue_q4_iso0_with_group.pdf
7308 \> #pdflatex-surface_catalogue_q4_iso0_quartics.tex
7309 \> #open-surface_catalogue_q4_iso0_quartics.pdf
7310
7311
NB_CUBIC_SURFACES_Q7=1

quartic_curves_q7:

quartic_curves_q7_classify:

NB_CUBIC_SURFACES_Q13=4

quartic_curves_q13:
7359 ▸ ▸ -define F=finite_field_q13-end\ 
7360 ▸ ▸ -define P=projective_space_3-F-end\ 
7361 ▸ ▸ -loop L=0-$\langle$NB_CUBIC_SURFACES_Q13$\rangle$.41\ 
7362 ▸ ▸ ▸ -with Q=F-end\ 
7363 ▸ ▸ ▸ ▸ -projective_space_activity\ 
7364 ▸ ▸ ▸ ▸ ▸ -define_surface S13_L=q13-catalogue_L-end\ 
7365 ▸ ▸ ▸ ▸ ▸ -end\ 
7366 ▸ ▸ ▸ ▸ -end_loop\ 
7367 ▸ ▸ ▸ -print_symbols\ 
7368 ▸ ▸ ▸ -loop L=0-$\langle$NB_CUBIC_SURFACES_Q13$\rangle$.41\ 
7369 ▸ ▸ ▸ ▸ -with S13_L-do\ 
7370 ▸ ▸ ▸ ▸ -cubic_surface_activity\ 
7371 ▸ ▸ ▸ ▸ ▸ -export_all_quartic_curves\ 
7372 ▸ ▸ ▸ ▸ ▸ -end\ 
7373 ▸ ▸ ▸ ▸ -end_loop\ 
7374 ▸ ▸ ▸ -print_symbols
7375 ▸ ▸ ▸ #pdflatex-surface_catalogue_q13_iso0_quartics.tex
7376 ▸ ▸ ▸ #open-surface_catalogue_q13_iso0_quartics.pdf
7377 ▸ ▸ ▸ #pdflatex-surface_catalogue_q13_iso1_quartics.tex
7378 ▸ ▸ ▸ #open-surface_catalogue_q13_iso1_quartics.pdf
7379 ▸ ▸ ▸ #pdflatex-surface_catalogue_q13_iso2_quartics.tex
7380 ▸ ▸ ▸ #open-surface_catalogue_q13_iso2_quartics.pdf
7381 ▸ ▸ ▸ #pdflatex-surface_catalogue_q13_iso3_quartics.tex
7382 ▸ ▸ ▸ #open-surface_catalogue_q13_iso3_quartics.pdf
7383
7384
7385 #surface_catalogue_q13_iso0_quartics.csv
7386
7387 quartic_curves_q13_classify:
7388 ▸ $(ORBITER_PATH)orbiter.out-v.3-
7389 ▸ ▸ -list_arguments-
7390 ▸ ▸ -define F=finite_field_q13-end-
7391 ▸ ▸ -define P=projective_space_2-F-end-
7392 ▸ ▸ -with P-do-
7393 ▸ ▸ -projective_space_activity-
7394 ▸ ▸ ▸ -classify_quartic_curves_with_substructure-
7395 ▸ ▸ ▸ ▸ surface_catalogue_q13_iso%d_quartics.csv-
7396 ▸ ▸ ▸ ▸ ▸ 3.4.4-quartic_curves_q13-
7397 ▸ ▸ ▸ ▸ ▸ -end-
7398 ▸ ▸ ▸ ▸ -print_symbols-
7399
7400 #q13
7401 #The-number-of-types-of-quartic-curves.is.2
7402 #idx.:ago
7403 #0,:24
7404 #1,:48
7405
589
quartic_curves_q17:

> $(ORBITER_PATH)orbiter.out -v.3:\
>   -list_arguments:\
>   -define F -finite_field -q.17 -end:\
>   -define P -projective_space -3 F -end:\
>   -loop L 0 $(NB_CUBIC_SURFACES Q17) 1:\
>   -with P -do:\
>   -projective_space activity:\
>   -define_surface S17 %L -q.17 -catalogue %L -end:\
>   -end:\
>   -print_symbols:\
>   -loop L 0 $(NB_CUBIC_SURFACES Q17) 1:\
>   -with S17 %L -do:\
>   -cubic_surface activity:\
>   -report:\
>   -export_all_quartic_curves:\
>   -end:\
>   -end_loop:\

> -print_symbols

> #pdfLaTeX surface_catalogue_q17_iso0_report.tex
> #open surface_catalogue_q17_iso0.pdf

quartic_curves_q17.classify:

> $(ORBITER_PATH)orbiter.out -v.3:\
>   -list_arguments:\
>   -define F -finite_field -q.17 -end:\
>   -define P -projective_space -2 F -end:\
>   -with P -do:\
>   -projective_space activity:\
>   -classify_quartic_curves_with_substructure:\
>   -surface_catalogue_q17_iso%d_quartics.csv:\
>   -$ (NB_CUBIC_SURFACES Q17) 3 4 quartic_curves_q17\
>   -end:\
>   -print_symbols:\

#User.time: 2:33
#q17
#The number of types of quartic curves is 7
#idx --- ago
#0: 24
 quartic_curves_q19:
  $(ORBITER_PATH)orbiter.out-v.3\$
  -list_arguments\$
  -define-F-finite_field-q.19-end\$
  -define-P-projective_space-3-F-end\$
  -loop-L.0-$\{NB_CUBIC_SURFACES_Q19\}.1\$
  -with-P-do\$
  -projective_space_activity\$
  -define_surface-S19_%L-q.19-catalogue_%L-end\$
  -end\$
  -end_loop\$
  -print_symbols\$
  -loop-L.0-$\{NB_CUBIC_SURFACES_Q19\}.1\$
  -with-S19_%L-do\$
  -cubic_surface_activity\$
  -report\$
  -export_all_quartic_curves\$
  -end\$
  -end_loop\$
  -print_symbols
  pdflatex:surface_catalogue_q19_iso0_report.tex
  open:surface_catalogue_q19_iso0_report.pdf

 quartic_curves_q19_classify:
  $(ORBITER_PATH)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\$
  -classify_quartic_curves_with_substructure\$
  surface_catalogue_q19_iso%d_quartics.csv
  $(NB_CUBIC_SURFACES_Q19).4-4_quartic_curves_q19\$
  -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\$
  \$\text{-list_arguments}\$
  \$\text{-define}\ F\ -\text{finite\ field}\ -q\ 19\ -end}\$
  \$\text{-define}\ P\ -\text{projective\ space\ 2\ F}\ -end\$
  \$\text{-with}\ P\ -do\$
  \$\text{-projective\ space\ activity}\$
  \$\text{-set\ stabilizer\ 4}\$
  \$\text{-define}\ \text{surface\ catalogue\ q19\ iso\ quartics\ csv}\$
  \$\text{-end}\$
  \$\text{-print\ symbols}\$
  \$\text{-end}\$

surface_13_0_quartics:
  \$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ -3\$
  \$\text{-define}\ F\ -\text{finite\ field}\ -q\ 13\ -end}\$
  \$\text{-define}\ P\ -\text{projective\ space\ 3\ F}\ -end\$
  \$\text{-with}\ P\ -do\$
  \$\text{-projective\ space\ activity}\$
  \$\text{-define}\ \text{surface\ S13_0\ -q\ 13\ -catalogue\ 0\ -end}\$
  \$\text{-end}\$
  \$\text{-cubic\ surface\ activity}\$
  \$\text{-export\ all\ quartic\ curves}\$
  \$\text{-end}\$
  \text{pdflatex}\ \text{surface\ catalogue\ q13\ iso0\ quartics.tex}\n  \text{open}\ \text{surface\ catalogue\ q13\ iso0\ quartics.pdf}\n
surface_13_1_quartics:
  \$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ -3\$
  \$\text{-define}\ F\ -\text{finite\ field}\ -q\ 13\ -end}\$
  \$\text{-define}\ P\ -\text{projective\ space\ 3\ F\ -end}\$
  \$\text{-with}\ P\ -do\$
quartic_curve_13_2_group:
-define_surface-S13.1-q13-catalogue.1-end-
-end-
-with_S13.1-do-
-cubic_surface_activity-
-define_surface-S13.1-q13-catalogue.1-end-
-export_all_quartic_curves-
-end
pdflatex_surface_catalogue_q13_iso1_quartics.tex
open_surface_catalogue_q13_iso1_quartics.pdf

quartic_curve_13_2
-define_G-linear_group-PGL3.13-
-subgroup_by_generators="quartic_13_2"24336.5-
1,0,0,1,0,1,2,12-
1,0,0,1,7,0,11,0,7-
1,0,0,8,12,1,4,11,1-
1,0,0,11,4,1,2,10,0-
0,1,0,11,3,0,4,9,1-
-restricted_action="29,86,97,154"-
-end-
-with_G-do-
-group_theoretic_activity-
-report-
-end
pdflatex_PGL3.13_Subgroup_quartic_13_2_24336_report.tex
open-PGL3.13_Subgroup_quartic_13_2_24336_report.pdf
#pdflatex-PGGL3.4_report.tex
#open-PGGL3.4_report.pdf

surface_25_0:
$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v.3$

- define F: finite field -q25-end
- define P: projective space 3 F-end
- with P:-do-
- projective space activity
- define_surface S25_0:-q25:-catalogue 0:-end-
end
- with S25_0:-do-
cubic_surface_activity
- report
- export all quartic curves:
- end

qdflatex\_surface\_catalogue\_q25\_iso0\_quartics.tex
open\_surface\_catalogue\_q25\_iso0\_quartics.pdf

quartic\_curve\_25\_report::
$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v.3$
- define F: finite field -q25-end
- define P: projective space 2 F-end
- loop L0-18:1
- with P:-do-
- projective_space_activity
- define quartic curve QC25_\%L
- q:25:-catalogue \%L-end
- end
- end_loop
- print_symbols
- loop L0-18:1
- with QC25_\%L:-do-
- quartic\_curve\_activity
- report
- end
- end_loop
- print_symbols
qdflatex\_quartic\_curve\_catalogue\_q25\_iso0\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso1\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso2\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso3\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso4\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso5\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso6\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso7\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso8\_report.tex
qdflatex\_quartic\_curve\_catalogue\_q25\_iso9\_report.tex
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--quartic_curve_catalogue_q25.pdf

#open-quartic_curve_catalogue_q25_iso0_report.pdf

quartic_curve_13_table:
$\langle\text{ORBITER\ PATH}\rangle\text{orbiter.out}-v.3\
-d\text{efineF}\text{-finite_field}\_q.13\_end\
-d\text{efineP}\text{-projective_space}\_2\_P\_end\
-w\text{ithP}\_do\
-\text{projective_space}\_activity\
-\text{table}\_of\_quartic\_curves\
-end

#quartic_curves_q13_info.csv

quartic_curve_19_table:
$\langle\text{ORBITER\ PATH}\rangle\text{orbiter.out}-v.3\$
\begin{verbatim}
7688  \triangleright  \triangleright  -define F\texttt{-finite_field}\texttt{-q 19\texttt{-end}}
7689  \triangleright  \triangleright  -define P\texttt{-projective_space2}\texttt{F\texttt{-end}}
7690  \triangleright  \triangleright  \texttt{-with P\texttt{-do}}
7691  \triangleright  \triangleright  \texttt{-projective_space}\texttt{activity}\texttt{.}
7692  \triangleright  \triangleright  \texttt{-table_of}\texttt{quartic}\texttt{curves}\texttt{.}
7693  \triangleright  \triangleright  \texttt{-end}
7694
7695  \texttt{quartic\texttt{curve19}}\texttt{.table}\texttt{latex}\texttt{:}
7696  \triangleright  $(\texttt{ORBITER}\texttt{\ PATH})\texttt{orbiter.out}\texttt{\ \texttt{-v 3}}
7697  \triangleright  \texttt{-csv}\texttt{\_file_latex1}\texttt{:quartic\texttt{\_curves\_q19}\texttt{\_info}}\texttt{.csv}
7698  \triangleright  ~/\texttt{bin/\texttt{tth}}\texttt{:quartic\texttt{\_curves\_q19}\texttt{\_info}}\texttt{.tex}
7699
7700
7701  \texttt{quartic\texttt{\_curve25}}\texttt{.table}\texttt{:}
7702  \triangleright  $(\texttt{ORBITER}\texttt{\ PATH})\texttt{orbiter.out}\texttt{\ \texttt{-v 3}}
7703  \triangleright  \texttt{-define F\texttt{-finite_field}\texttt{-q 25\texttt{-end}}}
7704  \triangleright  \texttt{-define P\texttt{-projective_space2}\texttt{F\texttt{-end}}}
7705  \triangleright  \texttt{-with P\texttt{-do}}
7706  \triangleright  \texttt{-projective_space}\texttt{activity}\texttt{.}
7707  \triangleright  \texttt{-table_of}\texttt{quartic}\texttt{curves}\texttt{.}
7708  \triangleright  \texttt{-end}
7709
7710
7711  \#\texttt{quartic\texttt{\_curves\_q25}\texttt{\_info}}\texttt{.csv}
7712
7713  \texttt{quartic\texttt{\_curve27}}\texttt{.table}\texttt{:}
7714  \triangleright  $(\texttt{ORBITER}\texttt{\ PATH})\texttt{orbiter.out}\texttt{\ \texttt{-v 3}}
7715  \triangleright  \texttt{-define F\texttt{-finite_field}\texttt{-q 27\texttt{-end}}}
7716  \triangleright  \texttt{-define P\texttt{-projective_space2}\texttt{F\texttt{-end}}}
7717  \triangleright  \texttt{-with P\texttt{-do}}
7718  \triangleright  \texttt{-projective_space}\texttt{activity}\texttt{.}
7719  \triangleright  \texttt{-table_of}\texttt{quartic}\texttt{curves}\texttt{.}
7720  \triangleright  \texttt{-end}
7721
7722
7723  \#\texttt{quartic\texttt{\_curves\_q27}\texttt{\_info}}\texttt{.csv}
7724
7725  \texttt{quartic\texttt{\_curve29}}\texttt{.table}\texttt{:}
7726  \triangleright  $(\texttt{ORBITER}\texttt{\ PATH})\texttt{orbiter.out}\texttt{\ \texttt{-v 3}}
7727  \triangleright  \texttt{-define F\texttt{-finite_field}\texttt{-q 29\texttt{-end}}}
7728  \triangleright  \texttt{-define P\texttt{-projective_space2}\texttt{F\texttt{-end}}}
7729  \triangleright  \texttt{-with P\texttt{-do}}
7730  \triangleright  \texttt{-projective_space}\texttt{activity}\texttt{.}
7731  \triangleright  \texttt{-table_of}\texttt{quartic}\texttt{curves}\texttt{.}
7732  \triangleright  \texttt{-end}
7733
7734
\end{verbatim}
quartic curves

quartic curves

$\text{\#quartic\_curves\_q29\_info.csv}$

quartic_curve_31.table:

$\text{\$\text{(ORBITER\_PATH)orbiter.out\_v.3\}}$

$\text{\-define\_F\_finite\_field\_q\_31\_end\}}$

$\text{\-define\_P\_projective\_space\_2\_F\_end\}}$

$\text{\-with\_P\_do\}}$

$\text{\-define\_projective\_space\_activity\}}$

$\text{\-table\_of\_quartic\_curves\}}$

$\text{\-end\}}$

$\text{\#quartic\_curves\_q31\_info.csv}$

surface_25_12:

$\text{\$\text{(ORBITER\_PATH)orbiter.out\_v.3\}}$

$\text{\-define\_F\_finite\_field\_q\_25\_end\}}$

$\text{\-define\_P\_projective\_space\_3\_F\_end\}}$

$\text{\-with\_P\_do\}}$

$\text{\-define\_projective\_space\_activity\}}$

$\text{\-define\_surface\_S25\_12\_q\_25\_catalogue\_12\_end\}}$

$\text{\-end\}}$

$\text{\-with\_S25\_12\_do\}}$

$\text{\-cubic\_surface\_activity\}}$

$\text{\-report\}}$

$\text{\-report\_with\_group\}}$

$\text{\-end\}}$

$\text{\-pdflatex\_surface\_catalogue\_q25\_iso12\_with\_group.tex}$

$\text{\-open\_surface\_catalogue\_q25\_iso12\_with\_group.pdf}$

$\text{\surface\_25\_12\_t1:}$

$\text{\$\text{(ORBITER\_PATH)orbiter.out\_v.3\}}$

$\text{\-define\_F\_finite\_field\_q\_25\_end\}}$

$\text{\-define\_P\_projective\_space\_3\_F\_end\}}$

$\text{\-with\_P\_do\}}$

$\text{\-define\_projective\_space\_activity\}}$

$\text{\-define\_surface\_S25\_12\_q\_25\_catalogue\_12\_end\}}$

$\text{\-transform\:"1,0,0,16,-0,1,0,18,-0,0,1,8,0,0,1,1,0".\}}$

$\text{\-end\}}$

$\text{\-end\}}$

$\text{\-with\_S25\_12\_do\}}$

$\text{\-cubic\_surface\_activity\}}$
\begin{verbatim}
7782 \triangleright \triangleright \triangleright -report\backslash
7783 \triangleright \triangleright -report_with_group\backslash
7784 \triangleright -end
7785 \triangleright pdflatex surface_catalogue_q25_iso12_with_group.tex
7786 \triangleright open surface_catalogue_q25_iso12_with_group.pdf
7787
7788
7789 \texttt{surface}_25.12_t2:
7790 \triangleright $(\text{ORBITER\_PATH})\text{orbiter.out}:\text{-v.3}\backslash
7791 \triangleright -defineF-\text{finite\_field}-q25-\text{-end}\backslash
7792 \triangleright -defineP-\text{projective\_space}-3F-\text{-end}\backslash
7793 \triangleright -withP-\text{-do}\backslash
7794 \triangleright -\text{projective\_space\_activity}\backslash
7795 \triangleright \triangleright \triangleright -\text{define\_surface\_S25.12-}\text{-q25}-\text{-catalogue\_12}\backslash
7796 \triangleright \triangleright -\text{transform\_inverse}-16,0,1,0,\ldots,1,0,0,1,1,0\backslash
7797 \triangleright \triangleright -\text{-end}\backslash
7798 \triangleright \triangleright -end\backslash
7799 \triangleright \triangleright -with\_S25.12-\text{-do}\backslash
7800 \triangleright \triangleright -\text{-cubic\_surface\_activity}\backslash
7801 \triangleright \triangleright \triangleright -\text{-report}\backslash
7802 \triangleright \triangleright \triangleright -\text{-report\_with\_group}\backslash
7803 \triangleright \triangleright -end\backslash
7804 \triangleright pdflatex surface_catalogue_q25_iso12_with_group.tex
7805 \triangleright open surface_catalogue_q25_iso12_with_group.pdf
7806
7807
7808
7809 \texttt{surface}_25.12_t3:
7810 \triangleright $(\text{ORBITER\_PATH})\text{orbiter.out}:\text{-v.3}\backslash
7811 \triangleright -defineF-\text{finite\_field}-q25-\text{-end}\backslash
7812 \triangleright -defineP-\text{projective\_space}-3F-\text{-end}\backslash
7813 \triangleright -withP-\text{-do}\backslash
7814 \triangleright -\text{projective\_space\_activity}\backslash
7815 \triangleright \triangleright \triangleright -\text{define\_surface\_S25.12-}\text{-q25}-\text{-catalogue\_12}\backslash
7816 \triangleright \triangleright \triangleright -\text{transform\_inverse}-16,0,1,0,\ldots,1,0,0,1,1,0\backslash
7817 \triangleright \triangleright \triangleright -\text{-end}\backslash
7818 \triangleright \triangleright \triangleright -with\_S25.12-\text{-do}\backslash
7819 \triangleright \triangleright -\text{-cubic\_surface\_activity}\backslash
7820 \triangleright \triangleright \triangleright -\text{-report}\backslash
7821 \triangleright \triangleright \triangleright -\text{-report\_with\_group}\backslash
7822 \triangleright \triangleright -end\backslash
7823 \triangleright pdflatex surface_catalogue_q25_iso12_with_group.tex
7824 \triangleright open surface_catalogue_q25_iso12_with_group.pdf
7825
7826
7827
7828
598
\end{verbatim}
surface_25_12_t4:
▷ $(\text{ORBITER\ PATH})\text{orbiter.out}$ -v.3
▷ -define\-finite_field\-q.25\-end
▷ -define\-projective_space\-3F\-end
▷ -with\-P\-do
▷ -projective_space_activity
▷ -define\-surface\-S25\_12\-q.25\-catalogue\-12
▷ -transform\"1,0,0,16,\-0,1,0,18,\-0,0,1,8,\-0,0,1,1,\-0\"\¬
▷ -transform\_inverse\"16,0,1,0,3,5,1,0,0,0,1,0,0,0,1,0\"\¬
▷ -transform\"3,0,0,0,1,0,0,0,1,0,0,0,0,1,0\"\¬
▷ -transform\_inverse\"1,0,0,0,\-0,1,0,0,\-0,0,1,0,13,2,2,1,0\"\¬
▷ -end
▷ -end
▷ -with\-S25\_12\-do
▷ -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_t5:
▷ $(\text{ORBITER\ PATH})\text{orbiter.out}$ -v.3
▷ -define\-finite_field\-q.25\-end
▷ -define\-projective_space\-3F\-end
▷ -with\-P\-do
▷ -projective_space_activity
▷ -define\-surface\-S25\_12\-q.25\-catalogue\-12
▷ -transform\"1,0,0,16,\-0,1,0,18,\-0,0,1,8,\-0,0,1,1,\-0\"\¬
▷ -transform\_inverse\"16,0,1,0,3,5,1,0,0,0,1,0,0,0,1,0\"\¬
▷ -transform\"3,0,0,0,1,0,0,0,1,0,0,0,0,1,0\"\¬
▷ -transform\_inverse\"1,0,0,0,\-0,1,0,0,\-0,0,1,0,13,2,2,1,0\"\¬
▷ -end
▷ -end
▷ -with\-S25\_12\-do
▷ -cubic\_surface\_activity
▷ -report
▷ -report\_with\_group
▷ -end
▷ pdflatex\-surface\_catalogue\_q25\_iso12\_with\_group.tex
▷ open\-surface\_catalogue\_q25\_iso12\_with\_group.pdf
7876  PG_2.25:
7877 ▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\$
7878 ▷ ▷ -define-F-finite_field-q\,25\,-end\$
7879 ▷ ▷ -define-P-projective_space-2-F\,-end\$
7880 ▷ ▷ -with-P-do-projective_space_activity\,-cheat_sheet\,-end
7881 ▷ pdflatex:PG_2.25.tex
7882 ▷ open-PG_2.25.pdf
7883
7884
7885
7886  PG_2.25_lines:
7887 ▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v\,\cdot\,5\,$\$
7888 ▷ ▷ -orbiter_path$(\text{ORBITER\_PATH})\,$$
7889 ▷ ▷ ▷ -define-G-linear_group-PGGL\,3\,\cdot\,25\,-on_k\_subspaces\,2\,-end\$
7890 ▷ ▷ ▷ -with-G\,-do\,$$
7891 ▷ ▷ ▷ -group_theoretic_activity\,$$
7892 ▷ ▷ ▷ ▷ -poset_classification_control\,$$
7893 ▷ ▷ ▷ ▷ ▷ -problem_label-PGGL\,3\,\cdot\,25\,$$
7894 ▷ ▷ ▷ ▷ ▷ -depth3\,-draw_poset\,-draw_options\,-radius\,200\,-end\,-report\,-end\,$$
7895 ▷ ▷ ▷ ▷ ▷ -recognize"0,25,650"$
7896 ▷ ▷ ▷ ▷ ▷ -recognize"430,16,364"
7897 ▷ ▷ ▷ ▷ ▷ -end\,$$
7898 ▷ ▷ ▷ ▷ -orbits_on_subsets\,3\,$$
7899 ▷ ▷ ▷ ▷ -end\,$$
7900 ▷ ▷ \,\,\,$$
7901 ▷ pdflatex:PGGL\,3\,\cdot\,25\,\_\text{poset}.tex
7902 ▷ open-PGGL\_3\_25\_\text{poset}.pdf
7903
7904  surface_25_12_t6:
7905 ▷ $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v\,\cdot\,3\,$$
7906 ▷ ▷ -define-F-finite_field-q\,25\,-end\,$$
7907 ▷ ▷ -define-P-projective_space-3-F\,-end\,$$
7908 ▷ ▷ -with-P\,-do\,$$
7909 ▷ ▷ -projective_space_activity\,$$
7910 ▷ ▷ ▷ -define_surface-S25\,12\,-q\,25\,-catalogue\,12\,$$
7911 ▷ ▷ ▷ -transform"1,0,0,16,0,1,0,18,0,0,1,8,0,1,1,1"$
7912 ▷ ▷ ▷ -transform_inverse"16,0,1,0,3,5,1,0,0,0,1,0,0,0,1,0"$
7913 ▷ ▷ ▷ -transform"3,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1"$
7914 ▷ ▷ ▷ -transform_inverse"1,0,0,0,0,1,0,0,0,0,1,0,13,2,2,1"$
7915 ▷ ▷ ▷ -transform_inverse"1,0,0,0,0,1,0,0,13,1,1,0,0,0,1,0"$
7916 ▷ ▷ ▷ -transform"3,8,8,0,22,13,22,0,14,19,15,0,0,0,1,1"$
7917 ▷ ▷ ▷ -transform_inverse"16,0,0,0,0,16,0,0,21,21,21,0,0,0,1,0"$
7918 ▷ ▷ ▷ -end\,$$
7919 ▷ ▷ ▷ -end\,$$
7920 ▷ ▷ -with-S25\,12\,-do\,$$
7921 ▷ ▷ -cubic_surface_activity\,$$
7922 ▷ ▷ ▷ -report\,$$

600
PG_2_25_stab_of_triangle:

```latex
> $(\text{ORBITER\_PATH})\text{orbiter}\_\text{out}\_v\_5\backslash$
> -orbiter\_path\$(\text{ORBITER\_PATH})\backslash$
> -define-G-\text{linear}\_\text{group}-\text{PGGL\_3}\_25\backslash$
> -\text{subgroup}\_\text{by}\_\text{generators}''\text{triangle}\_\text{stab}''-6912.7\backslash$
> "1,0,0,0,1,0,0,0,1,1,\backslash$
> 1,0,0,0,13,0,0,0,13,1,\backslash$
> 1,0,0,0,4,0,0,0,6,1,\backslash$
> 1,0,0,0,17,0,0,0,13,0,\backslash$
> 1,0,0,0,18,0,0,0,4,1,\backslash$
> 1,0,0,0,11,0,1,0,0,\backslash$
> 0,1,0,0,0,20,14,0,0,0\"\backslash$
> -end\backslash$
> -with-G-do\backslash$
> -group\_theoretic\_activity\backslash$
> -\text{poset}\_\text{classification}\_\text{control}\backslash$
> -\text{problem}\_\text{label}\_\text{PGGL}\_3\_25\backslash$
> -\text{depth}\_3\_\text{draw}\_\text{poset}\_\text{draw}\_\text{options}\_\text{radius}\_200-\text{end}\backslash$
> -\text{recognize}.''8,44,226''\backslash$
> -end\backslash$
> -orbits\_on\_subsets\_3\backslash$
> -end\backslash

surface_25_12_t7:

```
surface_25_12_t8:
    $(ORBITER\ PATH)orbiter.out-v.3\$
    \texttt{-define F-finite_field-q.25-}\end\$
    \texttt{-define P-projective_space-3-F-}\end\$
    \texttt{-with P-}\do\$
    \texttt{-projective_space_activity-}\end\$
    \texttt{-define_surface_S25_12-q.25-}\catalogue\12-\end\$
    \texttt{-transform"1,0,0,16,0,1,0,18,0,0,1,8,0,0,1,1,0"-}\end\$
    \texttt{-transform_inverse"16,0,1,0,3,5,1,0,0,0,1,0,0,0,1,0-}\end\$
    \texttt{-transform"3,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0-}\end\$
    \texttt{-transform_inverse"1,0,0,0,1,0,0,0,1,0,1,0,13,2,2,1,1,0-}\end\$
    \texttt{-transform_inverse"1,0,0,0,1,0,0,13,1,1,0,0,0,0,1,0-}\end\$
    \texttt{-transform"3,8,8,0,22,13,22,0,14,19,15,0,0,0,0,1,1-}\end\$
    \texttt{-transform_inverse"16,0,0,0,0,16,0,0,21,21,21,0,0,0,0,1,0-}\end\$
    \texttt{-transform_inverse"1,0,0,0,0,5,0,0,0,17,0,0,0,0,1,1,1-}\end\$
    \texttt{-transform inverse"1,0,0,0,0,1,0,0,1,0,0,0,0,0,24,0-}\end\$
    \texttt{-with S25.12-}\do\$

surface_25_12_t9:
    $(ORBITER\ PATH)orbiter.out-v.3\$
    \texttt{-define F-finite_field-q.25-}\end\$
    \texttt{-define P-projective_space-3-F-}\end\$
    \texttt{-with P-}\do\$
    \texttt{-projective_space_activity-}\end\$
    \texttt{-define_surface_S25_12-q.25-}\catalogue\12-\end\$
    \texttt{-transform"1,0,0,16,0,1,0,18,0,0,1,8,0,0,1,1,0-}\end\$

602
inverse surface with q25 iso12 space 12 curves:

\begin{verbatim}
8017 \verb|\^{-\text{transform\_inverse}}|"16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,1,0";
8018 \verb|\^{-\text{transform}}|"3,0,0,0,0,1,0,0,-0,1,0,0,0,1,0";
8019 \verb|\^{-\text{transform\_inverse}}|"1,0,0,0,0,1,0,0,-0,1,0,13,2,2,1,0";
8020 \verb|\^{-\text{transform\_inverse}}|"1,0,0,0,0,1,0,0,-0,1,0,0,1,0";
8021 \verb|\^{-\text{transform}}|"3,8,8,0,22,13,2,2,0,14,19,15,0,0,0,1,1";
8022 \verb|\^{-\text{transform\_inverse}}|"16,0,0,0,-0,16,0,0,21,21,21,0,-0,0,0,1,0";
8023 \verb|\^{-\text{transform}}|"1,0,0,0,0,5,0,0,-0,0,17,0,-0,0,1,1";
8024 \verb|\^{-\text{transform}}|"1,0,0,0,0,1,0,0,-0,1,0,0,0,0,24,0";
8025 \verb|\^{-\text{transform}}|"0,1,0,0,-0,0,1,0,-0,0,0,1,1,0,0,0,0";
8026 \verb|\end{-\text{do}}|;
8027 \verb|\end{-\text{do}}|;
8028 \verb|\-\text{with\_S25\_12}\-\text{do}\;
8029 \verb|\-\text{cubic\_surface\_activity}\;
8030 \verb|\-\text{report}\;
8031 \verb|\-\text{report\_with\_group}\;
8032 \verb|\end{-\text{end}}\;
8033 \verb|\text{pdflatex\_surface\_catalogue\_q25\_iso12\_with\_group\_tex}\;
8034 \verb|\text{open\_surface\_catalogue\_q25\_iso12\_with\_group\_pdf}\;
8035 
8036 
8037 
8038 
8039 
8040 \verb|\text{surface\_25\_12\_t8\_quartic\_curves}:\;
8041 \verb|\$\{\text{ORBITER\_PATH}\text{orbi}ter\_\text{out}\-\text{v}\-3\}|
8042 \verb|\defineF\-\text{finite\_field}\-q\-25\-\text{end}\;
8043 \verb|\defineF\-\text{projective\_space}\-3\-F\-\text{end}\;
8044 \verb|\with\-F\-\text{do}\;
8045 \verb|\-\text{projective\_space\_activity}\;
8046 \verb|\defineF\-\text{surface}\-S25\_12\-q\-25\-\text{-catalogue}\-12\;
8047 \verb|\-\text{transform\_inverse}}|"16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,1,0";
8048 \verb|\-\text{transform\_inverse}}|"16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,1,0";
8049 \verb|\-\text{transform}}|"3,0,0,0,0,1,0,0,-0,1,0,0,0,0,1,0";
8050 \verb|\-\text{transform\_inverse}}|"1,0,0,0,0,1,0,0,-0,1,0,13,2,2,1,0";
8051 \verb|\-\text{transform\_inverse}}|"1,0,0,0,0,1,0,0,-0,1,0,13,2,2,1,0";
8052 \verb|\-\text{transform}}|"3,8,8,0,22,13,2,2,0,14,19,15,0,0,0,1,1";
8053 \verb|\-\text{transform\_inverse}}|"16,0,0,0,-0,16,0,0,21,21,21,0,-0,0,0,1,0";
8054 \verb|\-\text{transform}}|"1,0,0,0,0,5,0,0,-0,0,17,0,-0,0,0,1,1";
8055 \verb|\-\text{transform}}|"1,0,0,0,0,1,0,0,-0,0,1,0,0,0,0,24,0";
8056 \verb|\end{-\text{do}}|;
8057 \verb|\end{-\text{do}}|;
8058 \verb|\-\text{with\_S25\_12}\-\text{do}\;
8059 \verb|\-\text{cubic\_surface\_activity}\;
8060 \verb|\-\text{all\_quartic\_curves}\;
8061 \verb|\end{-\text{do}}|;
8062 \verb|\-\text{with\_S25\_12}\-\text{do}\;
8063 \verb|\-\text{cubic\_surface\_activity}\;

603
quartic_curve_13_0_surface:
$\{(ORBITER\_PATH)\text{orbiter.out-}v.3\}$

define F - finite_field - q.13 - end
define P - projective_space:2 F - end
with P - do
  define quartic_curve QC13_0 - q.13 - catalogue:0 - end
end
with QC13_0 - do
  quartic_curve_activity
create quartic_curve
end
end
with quartic_curve - do
  cubic_surface_activity
report
end

#surface_equation: 0, 0, 9, 0, 3, 0, 1, 0, 12, 7, 0, 4, 4, 0, 7, 12, 6, 2, 5, 10
#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

quartic_curve_13_0_surface_create:
$\{(ORBITER\_PATH)\text{orbiter.out-}v.3\}$

define F - finite_field - q.13 - end
define P - projective_space:3 F - end
with P - do
  projective_space_activity
  define_surface S - by_coefficients
  "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"
  -q.13 - end
  end
with S - do
  cubic_surface_activity
  report
end
end
with cubic_surface_activity - do
  report
end

with quartic_curve - do
  cubic_surface_activity
report
end

#Section.7.3:Cubic.Surfaces.Classification

SECTION_CUBIC_SURFACES_CLASSIFICATION:
surface_classify_q4:

```
> $(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> with Surf27 - do \n> classification_of_cubic_surfaces_with_double_sixes_activity \n> report - end \n> end \n> define P - projective_space - 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 \n> pdflatex Surfaces_q4.tex \n> open Surfaces_q4.pdf
```

# time: 0:00

```
surface_classify_q4_arc_lifting_two_lines:

```
> $(ORBITER_PATH)orbiter.out -v 10 \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> control six_arcs - problem_label sixarcs q4 - end \n> classify_surfaces_through_arcs_and_two_lines \n> end \n> pdflatex surfaces_arc_lifting_4.tex \n> open surfaces_arc_lifting_4.pdf
```

```
surface_classify_q7:

```
> $(ORBITER_PATH)orbiter.out -v 5 \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 Surf27 W end \n> with Surf27 - do \n```
8158  \ -classification_of_cubic_surfaces_with_double_sixes_activity\ 
8159  \ -report.-end\ 
8160  \ -end\ 
8161  \ -print_symbols 
8162  \ pdflatex Surfaces_q7.tex
8163  \ open Surfaces_q7.pdf
8164
8165
8166  \ surface_classify_q13:\n8167  \ $(\text{ORBITER\ PATH})\text{orbiter.out}-v.5\ 
8168  \ -defineF-\text{finite_field}-q.13.-end\ 
8169  \ -defineF-\text{projective_space:3F.-end}\ 
8170  \ -withP.-do\ 
8171  \ -projective_space_activity\ 
8172  \ -classify_surfaces_with_double_sixes:C.-W.-end\ 
8173  \ -end\ 
8174  \ -withC.-do\ 
8175  \ -classify_surfaces_with_double_sixes_activity\ 
8176  \ -report.-end\ 
8177  \ -end\ 
8178  \ -print_symbols 
8179  \ pdflatex Surfaces_q13.tex
8180  \ open Surfaces_q13.pdf
8181
8182
8183
8184  #-----------------------------------------------------------------------------------------
8185  # Section 7.4: Cubic Surfaces - Isomorphism Testing and Recognition
8186
8187
8188  SECTION_CUBIC_SURFACES_ISOMORPHISM_TESTING_AND_RECOGNITION:
8189
8190
8191  \ surface_recognize_q7_abcd_2.3.3.4:\n8192  \ $(\text{ORBITER\ PATH})\text{orbiter.out}-v.3\ 
8193  \ -defineF-\text{finite_field}-q.7.-end\ 
8194  \ -defineF-\text{projective_space:3F.-end}\ 
8195  \ -withP.-do\ 
8196  \ -projective_space_activity\ 
8197  \ -classify_surfaces_with_double_sixes:Surf.-W.-end\ 
8198  \ -end\ 
8199  \ -withSurf.-do\ 
8200  \ -classification_of_cubic_surfaces_with_double_sixes_activity\ 
8201  \ -recognize\ 
8202  \ -q.7\ 
8203  \ -family_general_abcd_2.3.3.4\ 
8204  \ -end\ 

606
surface_isomorph_16:
 $(ORBITER_PATH)orbiter.out-v.3\ 
 -define F-finite_field-q16-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\ 
 -isomorphism_testing\ 
 -q16-by_coefficients\ 
 "1,5,1,8,1,9,1,10,1,11,1,12,6,14,6,15,7,18,7,19"-end\ 
 -q16-by_coefficients\ 
 "13,6,3,8,3,11,13,13,1,19"-end\ 
 -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-q8-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\ 
 -q8\ 
 "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 3 F - end 
with P - do 
projective_space_activity 
$\text{classify\_surfaces\_with\_double\_sixes}\_Surf27 - W - end 
$\text{end} 
with Surf27 - do 
$\text{classification\_of\_cubic\_surfaces\_with\_double\_sixes}$ activity 
$\text{identify_F13} 
$\text{end} 
with $\text{Surf27} - do 
$$\text{classification\_of\_cubic\_surfaces\_with\_double\_sixes}$ activity 
$\text{identify_F13} 
$\text{end} 
with $\text{Surf27} - do 
``` 

F_sweep_15_q7:

```
$(ORBITER_PATH)/orbiter.out -v 20 \
define F - finite_field -q 13 - end \
define P - projective_space 3 F - end 
with P - do 
projective_space_activity 
$\text{sweep} \\
\text{end} 
with Surf27 - do 
$\text{classification\_of\_cubic\_surfaces\_with\_double\_sixes}$ activity 
$\text{end} 
with Surf27 - do 
``` 

surf_sweep_Cayley_13:

```
$(ORBITER_PATH)/orbiter.out -v 3 \
define F - finite_field -q 13 - end \
define P - projective_space 3 F - end 
with P - do 
projective_space_activity 
$\text{sweep} \\
\text{end} 
with Surf27 - do 
$\text{classification\_of\_cubic\_surfaces\_with\_double\_sixes}$ activity 
$\text{end} 
with Surf27 - do 
``` 

sweep 15 q 7:

```
$\text{sweep 4\_15\_lines}\_sweep_4\_15\_lines\_q7 - q 7 \\
\text{by\_equation} \"F\_alpha\_beta\_gamma\_delta\" \\
"\{\alpha,\beta,\gamma,\delta\}\"x_0, x_1, x_2, x_3\" \\
$F\{\alpha,\beta,\gamma,\delta\} \\
\text{by\_equation} \"alpha=2,\beta=1,\gamma=2,\delta=3\" \\
\text{by\_equation} \"\{\alpha,\beta,\gamma,\delta\}\" \text{end} \\
$\text{end} 
``` 

608
# Section 7.5: Cubic Surfaces of Dickson type

SECTION CUBIC SURFACES DICKSON:

D6 q2:

```
$\text{(ORBITER_PATH)orbiter.out}\ -v\ 3 \$
```

```
$\text{-define F:\ finite field-q2-end}$
```

```
$\text{-define P:\ projective space\ 3 F\ -end}$
```

```
$\text{-with P:\ -do}$
```

```
$\text{-projective space\ activity}$
```

```
$\text{-define surface\ S\ D6 q2-q 2-by coefficients\$(D6)\ -end}$
```

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

```
$\text{-with S\ D6 q2\ -do}$
```

```
$\text{-cubic surface\ activity}$
```

```
$\text{-define surface\ S\ D6 q2-q 2-by coefficients\$(D6)\ -end}$
```

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

```
$\text{pdflatex surface\ by\ coefficients\ q2\ report.tex}$
```

```
$\text{open\ surface\ by\ coefficients\ q2\ report.pdf}$
```

```
$\text{mv surface\ by\ coefficients\ q2\ points.txt surface\ by\ coefficients\ q2\ D6\ points.txt}$
```

# 1 line over GF(2)

D3 q4:

```
$\text{(ORBITER_PATH)orbiter.out\ -v\ 3}$
```

```
$\text{-define F:\ finite field-q4-end}$
```

```
$\text{-define P:\ projective space\ 3 F\ -end}$
```

```
$\text{-with P:\ -do}$
```

```
$\text{-projective space\ activity}$
```

```
$\text{-define surface\ S\ D3 q4-q 4-by coefficients\$(D3)\ -end}$
```

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

```
$\text{-with S\ D3 q4\ -do}$
```
8345 \triangleright \triangleright \ -cubic_surface_activity\ \\`
8346 \triangleright \triangleright \ \triangleright \ -report\ \\`
8347 \triangleright \triangleright \ -end
8348 \triangleright pdflatex\ surface_by_coefficients_q4_report.tex
8349 \triangleright open_surface_by_coefficients_q4_report.pdf
8350 \triangleright mv\ surface_by_coefficients_q4_points.txt surface_by_coefficients_q4_D3_points.txt
8351 #surface_by_coefficients_q4_points.txt
8352
8353
8354
8355
8356
8357
8358
8359 D4_q8:
8360 \triangleright $(ORBITER\ PATH)\orbiter.out\ -v\ 3 \`
8361 \triangleright \triangleright \ -define\ F\ -finite_field\ -q\ 8\ -end\`
8362 \triangleright \triangleright \ -define\ P\ -projective_space\ -3\ F\ -end\`
8363 \triangleright \triangleright \ -with\ P\ -do\`
8364 \triangleright \triangleright \ -projective_space_activity\`
8365 \triangleright \triangleright \ \triangleright \ -define_surface\ S\ D4\ q8\ -q\ 8\ -by_coefficients\ $(D4)\ -end\`
8366 \triangleright \triangleright \ -end\`
8367 \triangleright \triangleright \ -with\ S\ D4\ q8\ -do\`
8368 \triangleright \triangleright \ -cubic_surface_activity\`
8369 \triangleright \triangleright \ \triangleright \ -report\`
8370 \triangleright \triangleright \ -end
8371 \triangleright pdflatex\ surface_by_coefficients_q8_report.tex
8372 \triangleright open_surface_by_coefficients_q8_report.pdf
8373 \triangleright mv\ surface_by_coefficients_q8_points.txt surface_by_coefficients_q8_D4_points.txt
8374
8375
8376
8377
8378
8379 D6_q4:
8380 \triangleright $(ORBITER\ PATH)\orbiter.out\ -v\ 3 \`
8381 \triangleright \triangleright \ -define\ F\ -finite_field\ -q\ 4\ -end\`
8382 \triangleright \triangleright \ -define\ P\ -projective_space\ -3\ F\ -end\`
8383 \triangleright \triangleright \ -with\ P\ -do\`
8384 \triangleright \triangleright \ -projective_space_activity\`
8385 \triangleright \triangleright \ \triangleright \ -define_surface\ S\ D6\ q4\ -q\ 4\ -by_coefficients\ $(D6)\ -end\`
8386 \triangleright \triangleright \ -end\`
8387 \triangleright \triangleright \ -with\ S\ D6\ q4\ -do\`
8388 \triangleright \triangleright \ -cubic_surface_activity\`
8389 \triangleright \triangleright \ \triangleright \ -report\`
D8_q4:

$(\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v.3:\$

```bash
$define F \text{-finite_field-} q^4 \text{-end}$
```

```bash
$define P \text{-projective_space-} 3F \text{-end}$
```

```bash
$with P \text{-do}$
```

```bash
$\text{-projective_space_activity}$
```

```bash
define surface $S_{\text{D8}} q^4 \_q^4 \_by\_coefficients $(\text{D8}) \text{-end}$
```

```bash
$-end$
```

```bash
with $S_{\text{D8}} q^4 \text{-do}$
```

```bash
-cubic_surface_activity$
```

```bash
-report$
```

```bash
-end$
```

```bash
pdflatex $\text{surface\_by\_coefficients\_q4\_report}$
```

```bash
mv $\text{surface\_by\_coefficients\_q4\_points}.\text{txt}$ $\text{surface\_by\_coefficients\_q4\_D6\_points}.\text{txt}$
```

### D6 has 7 lines over GF(4)

### D1_q8:

```bash
$\text{ORBITER}\_\text{PATH})\text{orbiter.out}\_v.3:\$
```

```bash
$define F \text{-finite_field-} q^8 \text{-end}$
```

```bash
$define P \text{-projective_space-} 3F \text{-end}$
```

```bash
$with P \text{-do}$
```

```bash
$\text{-projective_space_activity}$
```

```bash
define surface $S_{\text{D1}} q^8 \_q^8 \_by\_coefficients $(\text{D1}) \text{-end}$
```

```bash
$-end$
```

```bash
with $S_{\text{D1}} q^8 \text{-do}$
```

```bash
-cubic_surface_activity$
```

```bash
-report$
```

```bash
-end$
```

```bash
pdflatex $\text{surface\_by\_coefficients\_q8\_report}$
```

```bash
mv $\text{surface\_by\_coefficients\_q8\_points}.\text{txt}$ $\text{surface\_by\_coefficients\_q8\_D8\_points}.\text{txt}$
```
D1_q4_with_select_double_six:

D1_q4_with_select_double_six_b:

D1_q4_trans:
$\begin{align*}
&-\text{move\_two\_lines\_in\_hyperplane\_stabilizer\_text}\ \backslash \\
&-"1,0,0,0,0,0,1,1";"0,1,1,1,0,1,0,0"\backslash \\
&-"1,0,0,0,0,0,0,1";"0,1,0,1,1,0,1,0"\backslash \\
&-\text{end}\ \\
&D1\_q4\_with\_select\_double\_six\_c:\ \\
&\$(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_3\backslash \\
&-\text{define}\_F\_finite\_field\_q\_4\_end\backslash \\
&-\text{define}\_P\_projective\_space\_3\_F\_end\backslash \\
&-\text{with}\_P\_do\backslash \\
&D1\_q4\_with\_select\_double\_six\_c:\ \\
&-\text{define}\_S\_D1\_q4\_q\_4\_by\_coefficients\_$(D1)\_end\backslash \\
&-\text{select}\_double\_six\_"3,9,15,19,22,26,4,10,14,18,21,25"\backslash \\
&-\text{select}\_double\_six\_"1,2,3,4,5,0,7,8,9,10,11,6"\backslash \\
&-\text{transform}\_"1,0,0,0,0,1,0,0,0,1,0,0,0,1,1,0"\backslash \\
&-\text{end}\backslash \\
&-\text{end}\backslash \\
&-\text{with}\_S\_D1\_q4\_do\backslash \\
&-\text{cubic\_surface\_activity}\backslash \\
&-\text{report}\backslash \\
&-\text{end}\backslash \\
&-\text{poly\_orbits\_d3\_n3\_q3.pdf}\backslash \\
&\text{poly\_orbits\_d3\_n3\_q3.pdf}\backslash \\
&\#\text{this\_takes\_3\_days\_and\_about\_150\_GB\_memory\_on\_ripoff}\backslash \\
&-\text{orbits\_cubic\_surfaces\_q3}\backslash \\
&-\text{define}\_G\_linear\_group\_PGL\_4\_3\_end\backslash \\
&-\text{with}\_G\_do\backslash \\
&-\text{group\_theoretic\_activity}\backslash \\
&-\text{orbits\_on\_polynomials}\_3\backslash \\
&-\text{end}\backslash \\
&-\text{poly\_orbits\_d3\_n3\_q3.pdf}\backslash \\
&\text{poly\_orbits\_d3\_n3\_q3.pdf}\backslash \\
&\text{orbits\_cubic\_curves\_q2\_again}\backslash \\
&-\text{define}\_G\backslash \\
&-\text{linear\_group}\_\text{PGL\_3}\_2\backslash \\
&\end{align*}$
orbits.cubic_curves_q3:
$\text{($(ORBITER\_PATH)orbiter.out\_v\_4\_}$
$\text{-define-G\_}$
$\text{-define-linear\_group-PGL\_3\_3\_}$
$\text{-end\_}$
$\text{-with-G\_do\_}$
$\text{-group\_theoretic\_activity\_}$
$\text{-orbits\_on\_polynomials\_3\_}$
$\text{-end\_}$
pdflatex poly_orbits_d3_n2_q2.tex
open-poly_orbits_d3_n2_q2.pdf

compute_and_analyze_properties_over_F2

poly_orbits_d3_n3_q2_F2.csv:poly_orbits_d3_n3_q2.csv
$\text{($(ORBITER\_PATH)orbiter.out\_v\_4\_}$
$\text{-define-F\_finite\_field\_q\_2\_end\_}$
$\text{-define-P\_projective\_space\_3\_F\_end\_}$
$\text{-with-P\_do\_}$
$\text{-projective\_space\_activity\_}$
$\text{-table_of_cubic\_surfaces\_compute\_properties\_}$
poly_orbits_d3_n3_q2.csv:2.0
$\text{-end\_}$

Dickson_q2alyze:poly_orbits_d3_n3_q2_F2.csv
$\text{($(ORBITER\_PATH)orbiter.out\_v\_4\_}$
$\text{-define-F\_finite\_field\_q\_2\_end\_}$
$\text{-define-P\_projective\_space\_3\_F\_end\_}$
$\text{-with-P\_do\_}$
$\text{-projective\_space\_activity\_}$
cubic_surface_properties_analyze
poly_orbits_d3_n3_q2_F2.csv:2
$\text{-end\_}$
pdflatex poly_orbits_d3_n3_q2_F2_report.tex
open-poly_orbits_d3_n3_q2_F2_report.pdf
compute and analyze properties over $F_4$

poly_orbits_d3_n3_q2_F4.csv: poly_orbits_d3_n3_q2.csv

define $F$ - finite field - q 4 - end
define $P$ - projective space - 3 F - end
with $P$ - do
projective space activity

table of cubic surfaces compute properties

poly_orbits_d3_n3_q2.csv

Dickson_q4_analyze: poly_orbits_d3_n3_q2_F4.csv

define $F$ - finite field - q 4 - end
define $P$ - projective space - 3 F - end
with $P$ - do
projective space activity
cubic surface properties analyze

poly_orbits_d3_n3_q2_F4.csv

Dickson_q8_analyze: poly_orbits_d3_n3_q2_F8.csv

define $F$ - finite field - q 8 - end
define $P$ - projective space - 3 F - end
with $P$ - do
projective space activity
cubic surface properties analyze

poly_orbits_d3_n3_q2_F8.csv
# compute-and-analyze-properties-over-F16

- define F - finite field - q 16 - end
- define P - projective space - 3 F - end
- with P - do
- projective space activity
- table of cubic surfaces compute properties
- poly orbits d3 n3 q2 F16.csv
- end

Dickson_q16_analyze: poly_orbits_d3_n3_q2_F16.csv

- define F - finite field - q 16 - end
- define P - projective space - 3 F - end
- with P - do
- projective space activity
- cubic_surface_properties_analyze
- poly orbits d3 n3 q2 F16.csv
- end

pdflatex poly_orbits_d3_n3_q2_F16_report.tex

open poly_orbits_d3_n3_q2_F16_report.pdf

# Section 7.6: Cubic Surfaces -- ATLAS and Tables

SECTION_CUBIC_SURFACES_ATLAS_AND_TABLES:

MAKE_TABLE_OF_CUBIC_SURFACES=-define P - projective space - 3 F - end
- with P - do
- projective space activity
- table of cubic surfaces
- end

cubic_surfaces_tables_17:

$(ORBITER_PATH) orbiter.out --v 3

- define F - finite field - q 17 - end

$(MAKE_TABLE_OF_CUBIC_SURFACES)
8670 cubic_surfaces_table_latex_17:\n8671 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.4.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8672 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.7.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8673 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.8.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8674 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.9.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8675 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.11.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8676 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.13.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8677 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.16.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8678 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.17.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8679 cubic_surfaces_tables_up_to_17:\n8680 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.19.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8681 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.23.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8682 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.25.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8683 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.27.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8684 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.29.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8685 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.31.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8686 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.32.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8687 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.37.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8688 cubic_surfaces_tables_19_37:\n8689 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.19.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8690 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.23.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8691 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.25.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8692 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.27.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8693 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.29.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8694 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.31.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8695 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.32.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8696 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.37.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)\n8697 cubic_surfaces_tables_41_and_up:\n8698 ▶ $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q.41.-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
8700  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-43.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8701  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-47.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8702  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-49.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8703  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-53.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8704  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-59.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8705  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-61.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8706  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-64.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8707  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-67.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8708  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-71.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8709  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-73.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8710  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-79.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8711  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-81.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8712  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-83.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8713  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-89.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8714  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-97.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8715  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-101.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8716  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-103.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8717  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-107.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8718  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-109.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8719  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-113.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8720  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-121.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8721  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-127.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
8722  $(ORBITER_PATH)orbiter.out-v.3.-define-F--finite_field--q-128.-end$(MAKE_TABLE
OF_CUBIC_SURFACES)
cubic_surfaces_tables_latex::
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.1-test.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q4_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q7_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q8_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q9_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q11_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q13_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q16_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q17_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q19_info.csv}$

\begin{verbatim}
cubic_surfaces_tables_latex_big::
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q23_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q25_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q27_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q29_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q31_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q32_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q37_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q41_info.csv}$
$\texttt{(ORBITER PATH)orbiter.out-v.3-csv_file_latex.0-table_of_cubic_surfaces_q43_info.csv}$
\end{verbatim}

\begin{verbatim}
$\texttt{pdflatex-quartic_curves_q13_info.tex}$
$\texttt{open-quartic_curves_q13_info.pdf}$
$\texttt{~/bin/tth-quartic_curves_q13_info.tex}$
$\texttt{open-quartic_curves_q13_info.html}$
\end{verbatim}
8753
8754
8755
8756
8757 surface_table:
8758 ▶ $(ORBITER_PATH)orbiter.out-v.3-make_table_of_surfaces
8759 ▶ pdflatex: surfaces_report.tex
8760 ▶ open: surfaces_report.pdf
8761
8762
8763
8764
8765 surface_atlas:
8766 ▶ $(ORBITER_PATH)orbiter.out-v.3-create_surface_atlas.97
8767 ▶ ~/bin/tth-surface_atlas.tex
8768
8769
8770 surface_reports:
8771 ▶ $(ORBITER_PATH)orbiter.out-v.3-
8772 ▶ ▶ orbiter_path $(ORBITER_PATH)-create_surface_reports 4,7,8,9,11
8773
8774
8775
8776
8777
8778
8779
8780
8781 Surface_Orb0_q4:
8782 ▶ $(ORBITER_PATH)orbiter.out-v.3-
8783 ▶ ▶ -linear_group-PGL.4.4-wedge-end-
8784 ▶ ▶ -group_theoretic_activity-
8785 ▶ ▶ -control_six_arcs-end-
8786 ▶ ▶ -define_surface-label_txt:"Orb0_q4"-label_tex:"Rank-$\text{ Orb0}$"-by_rank:$\text{ Orb0}$-2-q.4-
8787 ▶ ▶ -end-
8788 ▶ ▶ -end.
8789
8790
8791
8792
8793
8794
8795
8796
8797 create_dickson_makefiles_q2:
8798 ▶ $(ORBITER_PATH)orbiter.out-v.create_files-file_mask:dickson_q2.%03d-
create_dickson_makefiles_q4:

$(ORBITER_PATH)/orbiter.out -create_files -file_mask dickson_q4_%03d

create_dickson_makefiles_q8:

$(ORBITER_PATH)/orbiter.out -create_files -file_mask dickson_q8_%03d

create_dickson_makefiles_q4:

$(ORBITER_PATH)/orbiter.out -create_files -file_mask dickson_q4_%03d

create_dickson_makefiles_q8:

$(ORBITER_PATH)/orbiter.out -create_files -file_mask dickson_q8_%03d
create_dickson_makefiles_q16:

$ (ORBITER_PATH) orbiter.out --create_files --file_mask dickson_q16_%03d \
- N-141 \n- line_numeric "Surface_Orb%d_q16: " \n- line_numeric "t\D(ORBITER_PATH)/orbiter.out -v 3 " \n- line_numeric "t\t-linear_group-PGL-4:16 -wedge -end " \n- line_numeric "t\t-group_theoretic_activity " \n- line_numeric "t\t-control_six_arcs -end " \n- line_numeric "t\t-define_surface -label_txt "Orb%d_q16 " \n- label_txt "Rank -D(Orb%d) " -by_rank -D(Orb%d) -2 -q 16 " \n- line_numeric \
- line_numeric "t\t-end " \n- line_numeric \
- end 

create_dickson_makefiles_q32:

$ (ORBITER_PATH) orbiter.out --create_files --file_mask dickson_q32_%03d \
- N-141 \n- line_numeric "Surface_Orb%d_q32: " \n- line_numeric "t\D(ORBITER_PATH)/orbiter.out -v 3 " \n- line_numeric "t\t-linear_group-PGL-4:32 -wedge -end " \n- line_numeric "t\t-group_theoretic_activity " \n- line_numeric "t\t-control_six_arcs -end " \n- line_numeric "t\t-define_surface -label_txt "Orb%d_q32 " \n- label_txt "Rank -D(Orb%d) " -by_rank -D(Orb%d) -2 -q 32 " \n- line_numeric \
- line_numeric "t\t-end " \n- line_numeric \
- end 

create_dickson_makefiles_q64:

$ (ORBITER_PATH) orbiter.out --create_files --file_mask dickson_q64_%03d \
- N-141 \n- line_numeric "Surface_Orb%d_q64: " \n- line_numeric "t\D(ORBITER_PATH)/orbiter.out -v 3 " \n- line_numeric "t\t-linear_group-PGL-4:64 -wedge -end " \n- line_numeric "t\t-group_theoretic_activity " \n- line_numeric "t\t-control_six_arcs -end " \n- line_numeric "t\t-define_surface -label_txt "Orb%d_q64 " \n- label_txt "Rank -D(Orb%d) " -by_rank -D(Orb%d) -2 -q 64 " \n- line_numeric \
- line_numeric "t\t-end " \n- line_numeric \
- end 

create_dickson_makefiles_q32:
create_dickson_makefiles_top_line_q2:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q2_top.\n  -N:1:-repeat:141:0:1:"Surface_q2:\"" -command "\tSurface_Orb%d_q2:\"" \n  -end

create_dickson_makefiles_top_line_q4:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q4_top.\n  -N:1:-repeat:141:0:1:"Surface_q4:\"" -command "\tSurface_Orb%d_q4:\"" \n  -end

create_dickson_makefiles_top_line_q8:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q8_top.\n  -N:1:-repeat:141:0:1:"Surface_q8:\"" -command "\tSurface_Orb%d_q8:\"" \n  -end

create_dickson_makefiles_top_line_q16:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q16_top.\n  -N:1:-repeat:141:0:1:"Surface_q16:\"" -command "\tSurface_Orb%d_q16:\"" \n  -end

create_dickson_makefiles_top_line_q32:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q32_top.\n  -N:1:-repeat:141:0:1:"Surface_q32:\"" -command "\tSurface_Orb%d_q32:\"" \n  -end

create_dickson_makefiles_top_line_q64:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q64_top.\n  -N:1:-repeat:141:0:1:"Surface_q64:\"" -command "\tSurface_Orb%d_q64:\"" \n  -end

surfaces_q4_join:
  $(ORBITER_PATH)orbiter.out -v:3.\n  -csv_file join:3\n  -csv_file join:0 Orb0_q4_summary.csv:"Surface"
  -csv_file join:1 Orb1_q4_summary.csv:"Surface"
  -csv_file join:2 Orb2_q4_summary.csv:"Surface"

create_dickson_makefiles_join_q2:
  $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q2_join.\n  -N:1:-repeat:141:0:1:"join:\n\t\D(ORBITER_PATH)/orbiter.out -v:3.\B"\n  -command "\t\t-csv_file join:0 Orb%d_q2_summary.csv:"Surface"\n  -B"
8940 ▷ ▷ -end
8941
8942
8943
8944 create_dickson_makefiles_join_q4:
8945 ▷ $(ORBITER_PATH)orbiter.out -create_files -file_mask dickson_q4_join:
8946 ▷ ▷ -N:1-repea141:0:1:"join:\n't\D(ORBITER_PATH)/orbiter.out -v.3\B"
8947 ▷ ▷ -command:"\t\t-csv_file join Orb%d_q4_summary.csv "Surface"\.B"
8948 ▷ ▷ -end
8949
dickson_q2_latex:
8950 ▷ $(ORBITER_PATH)orbiter.out -csv_file_latex q2.csv
8951
8952
8953
8954
8955 #problem:
8956 #orthogonal::lines on point by line rank i=1366 / 4225 pt=15447347 pt2=15225451
8957 #orthogonal::lines on point by line rank before rank line
8958 #orthogonal::rank line p1=15447347 p2=15225451
8959
create_dickson_atlas:
8960 ▷ $(ORBITER_PATH)orbiter.out -create_dickson_atlas
8961 ▷ ~/bin/tth dickson_surfaces.tex
8962
8963
8964
8965
8966 quartic_curve_tables_23::
8967 ▷ $(ORBITER_PATH)orbiter.out -v.3\n8968 ▷ ▷ -define F -finite_field -q 23 -end\n8969 ▷ ▷ -define P -projective_space 2 F -end\n8970 ▷ ▷ -with P -do\n8971 ▷ ▷ ▷ -projective_space_activity\n8972 ▷ ▷ ▷ ▷ -table_of_quartic_curves\n8973 ▷ ▷ ▷ ▷ -end
8974
8975 quartic_curve_tables::
8976 ▷ $(ORBITER_PATH)orbiter.out -v.3\n8977 ▷ ▷ -define F -finite_field -q 9 -end\n8978 ▷ ▷ -define P -projective_space 2 F -end\n8979 ▷ ▷ -with P -do\n8980 ▷ ▷ ▷ -projective_space_activity\n8981 ▷ ▷ ▷ ▷ -table_of_quartic_curves\n8982 ▷ ▷ ▷ ▷ -end
8983 ▷ ▷ $(ORBITER_PATH)orbiter.out -v.3\n8984 ▷ ▷ -define F -finite_field -q 13 -end\n8985 ▷ ▷ -define P -projective_space 2 F -end\n8986 ▷ ▷ -with P -do\n
quartic_curve_tables_latex::

$\text{(ORBITER PATH)}$ orbiter.out-v.3-

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q9_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q13_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q17_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q19_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q25_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q27_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q29_info.csv

$\text{(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-0-quartic_curves_q31_info.csv

$\text{#(ORBITER PATH)}$ orbiter.out-v.3-csv_file_latex-1-quartic_curves_q9_info.csv

#pdflatex-quartic_curves_q13_info.tex

#open-quartic_curves_q13_info.pdf

#~/bin/tth-quartic_curves_q13_info.tex

#open-quartic_curves_q13_info.html

inverse_mod_a:

$\text{(ORBITER PATH)}$ orbiter.out-v.2-inverse_mod-18059241-58014043
9034
9035  jacobi.35.41:
9036  \$\text{(ORBITER\_PATH)\text{\textbackslash orbiter.out\-v\-5\-\textbackslash jacobi.35.41}
9037  \text{\textbackslash pdflatex\textbackslash jacobi.35.41\textbackslash tex}
9038  \text{\textbackslash open\textbackslash jacobi.35.41.pdf}
9039  
9040  
9041  jacobi.33.41:
9042  \$\text{(ORBITER\_PATH)\text{\textbackslash orbiter.out\-v\-5\-\textbackslash jacobi.33.41}
9043  \text{\textbackslash pdflatex\textbackslash jacobi.33.41\textbackslash tex}
9044  \text{\textbackslash open\textbackslash jacobi.33.41.pdf}
9045  
9046  
9047  jacobi.a:
9048  \$\text{(ORBITER\_PATH)\text{\textbackslash orbiter.out\-v\-5\-\textbackslash jacobi.2221.7817}
9049  
9050  jacobi.5.19:
9051  \$\text{(ORBITER\_PATH)\text{\textbackslash orbiter.out\-v\-5\-\textbackslash jacobi.5.19}
9052  
9053  sqrt_mod.7817:
9054  \$\text{(ORBITER\_PATH)\text{\textbackslash orbiter.out\-v\-2\-\textbackslash square_root_mod.2221.7817}
9055  
9056  
9057  
9058  
9059  
9060  
9061  
9062  
9063  
9064  # Section 8.2: Representation Theory
9065  
9066  SECTION\_REPRESENTATION\_THEORY:
9067  
9068  
9069  
9070  representation\_on\_polynomials\_of\_degree\_3:
9071  \$\text{(ORBITER\_PATH)\text{\textbackslash orbiter.out\-v\-4\-\textbackslash define\textbackslash G\-linear\_group\-PGL.4.3\-end\-\textbackslash with\textbackslash G\-do\-\textbackslash group\_theoretic\_activity\-\textbackslash representation\_on\_polynomials.3\-\textbackslash end\-\textbackslash loop\_L.0.9.1\-\textbackslash draw\_matrix\-\textbackslash input\_csv\_file\textbackslash PGL.4.3\_rep.3\_\textbackslash L.csv\-\textbackslash box\_width.40\-\textbackslash bit\_depth.24\-\textbackslash partition.3.20.20\-\textbackslash end\-\textbackslash 626}
representation_tetrahedral_group_on_polynomials_of_degree_3:

$\text{(ORBITER_PATH)}\text{orbiter.out\,-v\,-4:\}$

$\text{define}\,G\text{-linear}\,\text{group}\,-\text{GL}\,3\,3\,\text{-subgroup}\,\text{by}\,\text{generators}\,\text{"tetra"\,"12"\,-2:\}$

"0,1,0,0,1,1,0,0,0,0,1,2,0,0,0,2,0"$

$\text{-end\,}$

$\text{with}\,G\text{-do\,}$

$\text{-group_theoretic_activity\,}$

$\text{-representation\,on\,polynomials\,of\,degree\,3:\}$

$\text{-end\,}$

$\text{draw\,matrix\,}$

$\text{input\,csv\,file\,GL\,3\,3\,Subgroup\,tetra\,12\,rep\,3\,\%L.csv\,}$

$\text{box\,width\,40\,-bit\,depth\,24\,-partition\,3\,10\,10\,-end\,-end\,loop\,}$

$\text{open\,GL\,3\,3\,Subgroup\,tetra\,12\,rep\,3\,0\,draw\,bmp\,}$

$\text{open\,GL\,3\,3\,Subgroup\,tetra\,12\,rep\,3\,1\,draw\,bmp\,}$

$\text{write\,GL\,3\,3\,Subgroup\,tetra\,12\,rep\,3\,0\,csv\,}$

$\text{SECTION\_CRYPTOGRAPHY:}\,$

$\text{EC\_add\,}$

$\text{EC\_cyclic\,subgroup\,}$

$\text{EC\,points\,13\,}$
EC_points.199:

```bash
$(ORBITER_PATH)orbiter.out -v.2:
-define F-finite_field-q.199-end
-with F-do
-finite_field_activity
-EC_points:"EC_5.7_q199".5.7-end
$(ORBITER_PATH)orbiter.out -v.2:
draw_matrix-input_csv_file:EC_5.7_q199_points_xy.csv
-box_width.20-bit_depth.24
-partition.2:"1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1".end
```

EC_Koblitz_encoding:

```bash
$(ORBITER_PATH)orbiter.out -v.6:
-define F-finite_field-q.199-end
-with F-do
-finite_field_activity
-EC_Koblitz_encoding:"EC_5.7_q199".5.7.67."147,164"."DEADBEEF".
-end
```

EC_bsgs:

```bash
$(ORBITER_PATH)orbiter.out -v.2:
-define F-finite_field-q.199-end
-with F-do
-finite_field_activity
-EC_bsgs:"EC_5.7_q199".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".
-end
```

EC_bsgs_decode:

```bash
$(ORBITER_PATH)orbiter.out -v.2:
-define F-finite_field-q.199-end
-with F-do
-finite_field_activity
-EC_bsgs_decode:"EC_bsgs".5.7.129,176.212.
```
NTRU
N=7
P=3
Q=41
D=2
NTRUE
XN1="-1,0,0,0,0,0,1,"
#D+1\cdot plus\cdot ones\cdot and\cdot D\cdot minus\cdot ones
ALICE_PRIVATE_F="-1,0,1,1,-1,0,1"
#D+1\cdot plus\cdot ones\cdot and\cdot D\cdot minus\cdot ones
ALICE_PRIVATE_G="0,-1,-1,0,1,0,1"

ALICE1:
$$(\text{ORBITER PATH})\text{orbiter.out} -v 2$$
-define F\cdot finite\cdot field\cdot q\cdot (NTRU)\cdot -end\$
-finite\cdot field\cdot activity\$
-extended\cdot gcd\cdot for\cdot polynomials\$
-define \$(NTRUE_XN1)\cdot $(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:
$$(\text{ORBITER PATH})\text{orbiter.out} -v 2$$
-define F\cdot finite\cdot field\cdot q\cdot (NTRU)\cdot -end\$
-finite\cdot field\cdot activity\$
-extended\cdot gcd\cdot for\cdot polynomials\$
-define \$(NTRUE_XN1)\cdot $(ALICE_PRIVATE_F)\$
-end

#F_p(x)=X^6+2X^5+X^3+X^2+X+1
ALICE_PRIVATE_FP="1,1,1,0,2,1"

ALICE\cdot public\cdot key:
$$(\text{ORBITER PATH})\text{orbiter.out} -v 2$$
-define F\cdot finite\cdot field\cdot q\cdot (NTRU)\cdot -end\$
\begin{verbatim}
9221  ▷▷ -with F-do\n9222  ▷▷ -finite_field_activity\n9223  ▷▷ -polynomial_mult_mod$(ALICE\_PRIVATE\_F)$\n9224  ▷▷ ▷ $\cdot $(ALICE\_PRIVATE\_G)$\cdot$(NTRUE\_XN1)$\n9225  ▷▷ ▷ -end
9226 9227  #C(X)=20X^{6}+40X^{5}+2X^{4}+38X^{3}+8X^{2}+26X+30
9228 ALICE\_PUBLIC\_KEY="30,26,8,38,2,40,20"
9229
9230 BOB\_MESSAGE="1,-1,1,0,-1"
9231 BOB\_ONE\_TIME\_KEY="-1,1,0,0,0,-1,1"
9232
9233 NTRU\_encrypt:
9234  ▷ $(ORBITER\_PATH)\_orbiter\_out-v.2\n9235  ▷ ▷ -define F-finite_field-q$\cdot$(NTRUE\_Q)$\cdot-end\n9236  ▷ ▷ -with F-do\n9237
9238 ▷ ▷ -finite_field_activity\n9239  ▷ ▷ -NTRU\_encrypt$\cdot$(NTRUE\_N)$\cdot$(NTRUE\_P)$\cdot$(ALICE\_PUBLIC\_KEY)$\n9240  ▷ ▷ ▷ $\cdot$(BOB\_ONE\_TIME\_KEY)$\cdot$(BOB\_MESSAGE)$\cdot-end
9241
9242  #E(X)=-31X^{6}+19X^{5}+4X^{4}+2X^{3}+40X^{2}+3X+25
9243 BOB\_ENCRYPT="25,3,40,2,4,19,31"
9244
9245 NTRU\_decrypt1:
9246  $(ORBITER\_PATH)\_orbiter\_out-v.2\n9247  ▷ ▷ -define F-finite_field-q$\cdot$(NTRUE\_Q)$\cdot-end\n9248  ▷ ▷ -with F-do\n9249
9250  ▷ ▷ -finite_field_activity\n9251  ▷ ▷ -polynomial_mult_mod$(ALICE\_PRIVATE\_F)$\n9252  ▷ ▷ ▷ $\cdot$(BOB\_ENCRYPT)$\cdot$(NTRUE\_XN1)$\n9253  ▷ ▷ ▷ -end
9254
9255  #C(X)=X^{6}-10X^{5}+33X^{4}+40X^{3}+40X^{2}+X+40
9256 ALICE\_C1="40,1,40,40,33,10,1"
9257
9258 NTRU\_decrypt2:
9259  $(ORBITER\_PATH)\_orbiter\_out-v.2\n9260  ▷ ▷ -define F-finite_field-q$\cdot$(NTRUE\_Q)$\cdot-end\n9261  ▷ ▷ -with F-do\n9262
9263  ▷ ▷ -finite_field_activity\n9264  ▷ ▷ -polynomial_center_lift$(ALICE\_C1)$\end{verbatim}
```plaintext
NTRU_decrypt3:
$(ORBITER_PATH)orbiter.out-v.2\n
$define F=finite_field-q$(NTRU_P)-end$
$define F=finite_field$
$finite_field-q$(NTRU_P)-end$
$finite_field_mod-p$$(ALICE_C2)-end$

$(ALICE_C2)$

#A(X)=X^6+X^5+X^4+2X^3+2X^2+X+2

ALICE_C3="2,1,2,1,1,1"

NTRU_decrypt4:
$(ORBITER_PATH)orbiter.out-v.2\n
$define F=finite_field-q$(NTRU_Q)-end$
$define F=finite_field$
$finite_field-q$(NTRU_Q)-end$
$finite_field_mod-p$$(ALICE PRIVATE_FP)-end$
$finite_field_mod-p$$(ALICE PRIVATE_FP)-do$

$(ALICE PRIVATE_FP)$

#A(X)=X^5+X^4+X^2+2X+1

ALICE_C4="1,2,1,0,1,2"

NTRU_decrypt5:
$(ORBITER_PATH)orbiter.out-v.2\n
$define F=finite_field-q$(NTRU_P)-end$
$define F=finite_field$
$finite_field-q$(NTRU_P)-end$
$finite_field_mod-p$$(ALICE_C4)-end$

$(ALICE_C4)$

#A(X)=-X^5+X^3+X^2+X^2+X+1

#plaintext.BO blindness

inv_59_mod:
$(ORBITER_PATH)orbiter.out-v.2-inverse_mod-59.10200

#the inverse of 59 mod 10200 is 2939

RSA_e:
$$(ORBITER_PATH)orbiter.out-v.2\n```
9315  ▷ ▷ -RSA-59·10403·2·"1921,1605,1804,2116,0518"
9316
9317
9318 RSA_d:
9319  ▷ $(ORBITER_PATH)orbiter.out-v.2\n9320  ▷ -RSA-2939·10403·2·"902,3509,9833,3548,5181"
9321
9322
9323 im1:
9324  ▷ $(ORBITER_PATH)orbiter.out-v.2-inverse_mod-869·1843488
9325
9326 #the inverse of 869·1843488 is 386093
9327
9328 #·FUNFACTOR::>
9329 RSA_e1:
9330  ▷ $(ORBITER_PATH)orbiter.out-v.2\n9331  ▷ -RSA-386093·1846303·3·"62114,60103,201518"
9332
9333 RSA_d1:
9334  ▷ $(ORBITER_PATH)orbiter.out-v.2\n9335  ▷ -RSA-869·1846303·3·"1248407,345776,317846"
9336
9337
9338
9339
9340
9341  #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9342  #·5503·4603·=·25330309
9343  #·5502·4602·=·25320204
9344
9345
9346 im1061:
9347  ▷ $(ORBITER_PATH)orbiter.out-v.2\n9348  ▷ -inverse_mod-1061·25320204
9349  ▷
9350 #·the inverse of 1061·25320204 is 2076209
9351
9352
9353
9354 RSA_e2:
9355  ▷ $(ORBITER_PATH)orbiter.out-v.2\n9356  ▷ -RSA_encrypt_text-2076209·25330309·3·creamcheese
9357
9358 #·-RSA_encrypt_text-386093·1846303·creamcheese
9359  #·408918,1735142,239809,654636
9360
9361

632
9362 RSA_d2:
9363  
9364  
9365  
9366  
9367  
9368  
9369  
9370 im3:
9371  
9372  
9373  
9374  
9375  
9376  
9377  
9378 RSA_e3:
9379  
9380  
9381 im4:
9382  
9383  
9384  
9385  
9386  
9387  
9388  
9389  
9390  
9391  
9392  
9393  
9394  
9395  
9396  
9397  
9398  
9399  
9400 RSA_e4:
9401  
9402  
9403  
9404  
9405  
9406  
9407 RSA_d4:
9408  

9362 RSA_d2:
9363  $(ORBITER_PATH)orbiter.out-v.2\ 
9364  -RSA-1061-25330309-3:"19019931,1619805,740498,2671344"
9365  
9366  
9367  
9368  
9369  
9370  
9371 im3:
9372  $(ORBITER_PATH)orbiter.out-v.2\ 
9373  -inverse_mod-2909-59248840
9374  
9375  
9376  #the inverse of 2909 mod 59248840 is 4358629
9377  
9378 RSA_e3:
9379  $(ORBITER_PATH)orbiter.out-v.2\ 
9380  -RSA_encrypt_text-2909-59264263-3:encrypted
9381  
9382 RSA_d3:
9383  $(ORBITER_PATH)orbiter.out-v.2\ 
9384  -RSA-4358629-59264263-3:"35270141,9642524,49091707"
9385  
9386  
9387  
9388  
9389  
9390  
9391  
9392  
9393  
9394  
9395 im4:
9396  $(ORBITER_PATH)orbiter.out-v.2--inverse_mod-583-62236200
9397  
9398  
9399  
9400 RSA_e4:
9401  $(ORBITER_PATH)orbiter.out-v.2\ 
9402  -RSA_encrypt_text-583-62251979-3:venividivici
9403  
9404  
9405  
9406  
9407 RSA_d4:
9408  $(ORBITER_PATH)orbiter.out-v.2\
\[
-\text{RSA} \cdot 32559247 \cdot 62251979 - 40513610, 53979973, 56449676, 35068535
\]

\[
-\text{RSA} \cdot 62251979 \cdot 38543669 - 52518863
\]

\[
-\text{RSA} \cdot 38543669 \cdot 52518863 \cdot 3 - \text{fascinating}
\]

\[
-\text{RSA} \cdot 52504368 \cdot 31526751, 8962078, 51045732, 51894467
\]

\[
-\text{RSA} \cdot 173 \cdot 52518863 \cdot 31526751, 8962078, 51045732, 51894467
\]

\[
-\text{sift_smooth} \cdot 100000 \cdot 100 - 2, 3, 5, 7, 11, 13, 17, 19
\]

\[
-\text{RSA} \cdot 47177497 \cdot 55040413 - 28702119, 48926559
\]

\[
-\text{RSA} \cdot 47177497 \cdot 55040413 - 28702119, 48926559
\]
9456 RSA_e7:
9457 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .2\ \$
9458 $\ -\text{RSA\ encrypt\_text}\cdot10048553\cdot15806093\cdot3\cdot\text{beachandfun}$
9459 $\ -$
9460 $\ #$
9461 7.853\cdot7.673=60256069
9462 7.852\cdot7.672=60240544
9463
9464 im8:
9465 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .2\ -\text{inverse\_mod}\cdot9017\cdot60240544$
9466 $\ -$
9467 $\ #$
9468 $\ #$
9469 $\ #$
9470 sqrt_big:
9471 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .2\ -\text{square\_root}\cdot1002001$
9472 $\ -$
9473 $\ -$
9474 $\ -$
9475 $\ -$
9476 $\ -$
9477 $\ -$
9478 sqrt_mod_33_41:
9479 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .2\ -\text{square\_root\_mod}\cdot33\cdot41$
9480 $\ -$
9481 $\ -$
9482 quadratic_sieve:
9483 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .5\ -\text{quadratic\_sieve}\cdot31\cdot500\cdot1$
9484 $\ -$
9485 $\ -$
9486 $\ -$
9487 $\ -$
9488 $\ -$
9489 pseudoprime3:
9490 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .5\$
9491 $\ -$
9492 $\ -$
9493 $\ -$
9494 $\ -$
9495 $\ -$
9496 pseudoprime10:
9497 $(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ .5\$
9498 $\ -$
9499 $\ -$
9500 $\ -$
9501 $\ -$
9502 4460190157
pseudoprime11:
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -power_mod 1293:2230095078:4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -power_mod 19645:2230095078:4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -power_mod 974586571:2230095078:4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -power_mod 974586571:1486730052:4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -power_mod 974586571:15222492:4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -power_mod 974586571:284796:4460190157

pseudoprime20:
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -primitive_root 4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -primitive_root 4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5

pseudoprime50:
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -primitive_root 4460190157
  $(\text{ORBITER PATH})\text{orbiter.out} -v 5 -primitive_root 4460190157

# mistake! long integer overflow
pseudoprime51:
▷ $(\text{ORBITER PATH})$ orbiter.out -v 5
▷ ▷ -seed 2531011 -find pseudoprime 51 5 5 5
▷ pdflatex pseudoprime_51.tex
▷ open pseudoprime_51.pdf

#75460072774683447021408970249004944659715367045417

pseudoprime30:
▷ $(\text{ORBITER PATH})$ orbiter.out -v 5
▷ ▷ -seed 2531011 -find pseudoprime 30 5 5 5
▷ pdflatex pseudoprime_30.tex
▷ open pseudoprime_30.pdf

#286525565474504516914595596387

pseudoprime31:
▷ $(\text{ORBITER PATH})$ orbiter.out -v 5
▷ ▷ -seed 2531011 -find pseudoprime 31 5 5 5
▷ pdflatex pseudoprime_31.tex
▷ open pseudoprime_31.pdf

#8777266765422645523724129853331

pseudoprime33:
▷ $(\text{ORBITER PATH})$ orbiter.out -v 5
▷ ▷ -seed 2531011 -find pseudoprime 33 5 5 5
▷ pdflatex pseudoprime_33.tex
▷ open pseudoprime_33.pdf

#37167419949829534554336304459891

pseudoprime34:
▷ $(\text{ORBITER PATH})$ orbiter.out -v 5
▷ ▷ -seed 2531011 -find pseudoprime 34 5 5 5
▷ pdflatex pseudoprime_34.tex
pseudoprime35:

$\text{(ORBITER PATH)} orbiter.out \ -v \ -5$

$\text{-seed:2531011-} \text{find_pseudoprime-35-5-5-}$

pdflatex pseudoprime_35.tex

open-pseudoprime_35.pdf

---

pseudoprime36:

$\text{(ORBITER PATH)} orbiter.out \ -v \ -5$

$\text{-seed:2531011-} \text{find_pseudoprime-36-5-5-}$

pdflatex pseudoprime_36.tex

open-pseudoprime_36.pdf

---

MATH360 hw2:

$\text{(ORBITER PATH)} orbiter.out \ -v \ -3$

$\text{-define F-} \text{finite_field-} \text{q-16-} \text{-end}$

$\text{-with F-} \text{do-} \text{finite_field_activity}$

$\text{-parse_and_evaluate-} \text{test"} \text{a*b} \text{a=8,b=14}$

$\text{(ORBITER PATH)} orbiter.out \ -v \ -3$

$\text{-define F-} \text{finite_field-} \text{q-16-} \text{-end}$

$\text{-with F-} \text{do-} \text{finite_field_activity}$

$\text{-parse_and_evaluate-} \text{test"} \text{a*b} \text{a=9,b=13}$

$\text{(ORBITER PATH)} orbiter.out \ -v \ -3$

$\text{-define F-} \text{finite_field-} \text{q-16-} \text{-end}$

$\text{-with F-} \text{do-} \text{finite_field_activity}$

$\text{-parse_and_evaluate-} \text{test"} \text{a*a*a*a*a} \text{a=9}$

$\text{(ORBITER PATH)} orbiter.out \ -v \ -3$

$\text{-define F-} \text{finite_field-} \text{q-16-} \text{-end}$

$\text{-with F-} \text{do-} \text{finite_field_activity}$

$\text{-parse_and_evaluate-} \text{test"} \text{(a+b)*(a+b)} \text{a=5,b=7}$

$\text{(ORBITER PATH)} orbiter.out \ -v \ -3$


```bash
9643  ▷ ▷ -define F=finite_field=q 16 -end\n9644  ▷ ▷ -with F=finite_field_activity\n9645  ▷ ▷ -parse_and_evaluate "test": "a*a+b*b": "a=5\n9646          b=7" -end
9647
9648 F_256_Rijndahl:
9649  ▷ $(ORBITER_PATH) orbiter.out -v 3\n9650  ▷ ▷ -define F=finite_field=q 256 -override polynomial 283 -end\n9651  ▷ ▷ -with F=do_finite_field_activity -cheat_sheet GF -end
9652
9653
9654
9655
9656
9657
9658
9659 all_square Roots_mod_n_1549411:
9660  ▷ $(ORBITER_PATH) orbiter.out -v 3 -all_square Roots_mod_n_1075922 1549411
9661  ▷
9662
9663
9664 power_mod_211:
9665  ▷ $(ORBITER_PATH) orbiter.out -v 3 -power_mod_n_2 211
9666  ▷ $(ORBITER_PATH) orbiter.out -v 3\n9667  ▷ ▷ -plot function power_mod_n_a2_n211.csv
9668  ▷ $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix\n9669  ▷ ▷ -input_csv_file power_mod_n_a2_n211_graph.csv\n9670  ▷ ▷ -box_width 10 -bit_depth 8 -partition 3 211 211 -end
9671
9672 power_mod_2_31:
9673  ▷ $(ORBITER_PATH) orbiter.out -v 3 -power_mod_n_2 31
9674  ▷ $(ORBITER_PATH) orbiter.out -v 3\n9675  ▷ ▷ -plot function power_mod_n_a2_n31.csv
9676  ▷ $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix\n9677  ▷ ▷ -input_csv_file power_mod_n_a2_n31_graph.csv\n9678  ▷ ▷ -box_width 10 -bit_depth 8 -partition 3 31 31 -end
9679
9680 power_mod_3_31:
9681  ▷ $(ORBITER_PATH) orbiter.out -v 3 -power_mod_n_3 31
9682  ▷ $(ORBITER_PATH) orbiter.out -v 3\n9683  ▷ ▷ -plot function power_mod_n_a3_n31.csv
9684  ▷ $(ORBITER_PATH) orbiter.out -v 2 -draw_matrix\n9685  ▷ ▷ -input_csv_file power_mod_n_a3_n31_graph.csv\n9686  ▷ ▷ -box_width 10 -bit_depth 8 -partition 3 31 31 -end
9687
9688
9689
```

639
SECTION CODING_THEORY_INTRODUCTION:

Hamming space distance matrix:

$$\text{$(ORBITER\ PATH)orbiter.out\ -Hamming\ space\ distance\ matrix\ 4\cdot2$}$$

- Draw matrix

- Input csv file

- Box width

- Bit depth

- Partition

- Open Hamming n4 q2 draw.bmp

- Code diagram

- Metric balls

- Do graph theoretic activity

- Export csv

- Export graphviz

- Save

- Code diagram

- Do graph theoretic activity

- Export

- Graph theoretic activity

- Save

- Do graph theoretic activity

- Export

- Export

- Save

- Hamming n4 q2 diagram

- Box width

- Bit depth

- Partition

- Hamming 5 2 graph:

- Define G - graph

- With G - do

- Graph theoretic activity - export

- Graph theoretic activity - export graphviz

- With G - do

- Graph theoretic activity - save

- Do graph theoretic activity

- Export

- Export

- Save

- Hamming 5 2 with 5 2 3 code:
(ORBITER_PATH)orbiter.out -v.2
-define G -graph -Hamming.5.2
-subset: "_code_5\_2\_3". "\_code\_5\_2\_3". "\_code\_5\_2\_3"
$(CODE_5\_2\_3\_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.5_2_code_5_2_3_report.tex
open Hamming.5_2_code_5_2_3_report.pdf

#group has order 32

# linear code with generator matrix

# Section 9.2: Hamming codes

SECTION CODING THEORY HAMMING CODES:

641
Hamming generator:

```bash
$ (ORBITER_PATH)orbiter.out -v.2\n  -define F=finite_field=q2-end\n  -define v=vector=field F=format 3\n  -dense $(SIMPLEX_CODE_GENERATOR)\n  -end\n  -with F=do\n  -finite_field_activity\n  -nullspace v\n  -end
```

Hamming code words:

```bash
$ (ORBITER_PATH)orbiter.out -v.2\n  -define v=vector=dense $(HAMMING_CODE_ROWS_IN_BINARY_RANKS)-end\n  -linear_code_through_basis 7 v
  -end\n```

Hamming weight enumerator:

```bash
$ (ORBITER_PATH)orbiter.out -v.2\n  -define F=finite_field=q2-end\n  -define v=vector=field F=format 4\n  -dense $(HAMMING_CODE_GENERATOR)\n  -end\n  -with F=do\n  -finite_field_activity\n  -weight_enum v-end
```

Hamming code diagram:

```bash
-$(ORBITER_PATH)orbiter.out-v.2-code_diagram "Hamming_7_4".\n  $(HAMMING_CODE_CODEWORDS)7-metric_balls 1\n  $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n  -input_csv_file Hamming_7_4_diagram_01_7_16.csv\n  -box_width 25-bit_depth 24-partition 4 16 8-end
```
\$\{ORBITER\_PATH\}orbiter.out-v.2.-draw_matrix\$
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_0\".\"0\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_1\".\"67\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_2\".\"37\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_3\".\"102\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_4\".\"22\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_5\".\"85\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_6\".\"51\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_7\".\"112\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_8\".\"15\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_9\".\"76\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_10\".\"42\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_11\".\"105\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_12\".\"25\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_13\".\"90\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_14\".\"60\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-code_diagram\"Hamming\_7\_4_word\_15\".\"127\".7.-metric\_balls:1
\$\{ORBITER\_PATH\}orbiter.out-v.2.-loop-L.0\_16\_1.-draw_matrix\$
\$\{ORBITER\_PATH\}orbiter.out-v.2.-input_csv_file-Hamming\_7\_4_word\_\%L_diagram\_7\_1.csv$
\$\{ORBITER\_PATH\}orbiter.out-v.2.-box_width-25.-bit_depth-14.-partition-4\_16\_8.-end$
\$\{ORBITER\_PATH\}orbiter.out-v.2.-end_loop$

code_Hamming_systematic:
\$\{ORBITER\_PATH\}orbiter.out-v.2$
\$\{ORBITER\_PATH\}orbiter.out-v.2.-define-v.-vector.-dense\$\{HAMMING\_CODE\_ROWS\_IN\_BINARY\_RANKS\}$.-end$
\$\{ORBITER\_PATH\}orbiter.out-v.2.-linear_code_through_basis.7.-v$

643
Hamming RREF:

Hamming nullspace:
# check equations of the Hamming code:

# a4+a5+a6+a7=1+0+1+0=0 mod 2 OK.
# a2+a3+a6+a7=0+1+1+0=0 mod 2 OK.
# a1+a3+a5+a7=1+1+0+0=0 mod 2 OK.

a2+a3+a6+a7 = 0+1+1+0 = 0 mod 2 OK.

a1+a3+a5+a7 = 1+1+0+0 = 0 mod 2 OK.

Hamming long:

$$(\text{ORBITER PATH})\text{orbiter.out}-v.2-long\_code.7.4$$

$$(\text{ORBITER PATH})\text{orbiter.out}-v.2-loop L.0.16.1-draw\_matrix$$

$$(\text{ORBITER PATH})\text{orbiter.out}-v.2-loop L.0.16.1-draw\_matrix$$

long_code_genma_n7_k4_codeword_0.csv
long_code_genma_n7_k4_codeword_15.csv
Weight distribution: (0,3^7,4^7,7)

Hamming code macwilliams:

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

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

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

Hamming singer:

$$(\text{ORBITER PATH})\text{orbiter.out}-v.3$$

$$(\text{ORBITER PATH})\text{orbiter.out}-v.3$$

$$(\text{ORBITER PATH})\text{orbiter.out}-v.3$$

$$(\text{ORBITER PATH})\text{orbiter.out}-v.3$$
Hamming cyclic generator:

```bash
$ (ORBITER_PATH) orbiter.out -v 2 -v -define F -finite_field -q 2 -end -define v -vector -format 3 -field F -dense $(SIMPLEX_CODE_GENMA_CYCLIC) -end -define -finite_field_activity -end
```

```
nullspace v
```
10003 $(\text{ORBITER\_PATH})\text{orbiter.out}\ \ -v.2\$
10004 \quad \triangleright \quad \text{define F finite field q 2 end}\$
10005 \quad \triangleright \quad \text{define v vector format 3 field F}\$
10006 \quad \triangleright \quad \text{-dense $(\text{SIMPLEX\_CODE\_GENMA\_CYCLIC})$}\$
10007 \quad \triangleright \quad \text{-end}\$
10008 \quad \triangleright \quad \text{-define F finite field q 2 end}\$
10009 \quad \triangleright \quad \text{-with F do finite field activity}\$
10010 \quad \triangleright \quad \text{-nullspace v}\$
10011 \quad \triangleright \quad \text{-end}\$
10012 \quad \text{pdflatex nullspace 4 7.tex}\$
10013 \quad \text{open nullspace 4 7.pdf}\$
10014
10015
10016 \text{Hamming cyclic clean:}\$
10017 \quad \triangleright \quad $(\text{ORBITER\_PATH})\text{orbiter.out}\ \ -v.2\$
10018 \quad \triangleright \quad \text{-define v vector dense 88,44,22,11 end}\$
10019 \quad \triangleright \quad \text{linear code through basis 7 v}\$
10020 \quad \triangleright \quad $(\text{ORBITER\_PATH})\text{orbiter.out}\ \ -v.2\ \ -draw\_matrix}\$
10021 \quad \triangleright \quad \text{-input csv file code matrix 16 8.csv}\$
10022 \quad \triangleright \quad \text{-box width 25 bit depth 8 partition 2 16 8 end}\$
10023 \quad \text{open code matrix 16 8 draw.bmp}\$
10024 \quad \text{pdflatex code 7 16.tex}\$
10025 \quad \text{open code 7 16.pdf}\$
10026
10027
10028 \text{Hamming cyclic clean long:}\$
10029 \quad \triangleright \quad $(\text{ORBITER\_PATH})\text{orbiter.out}\ \ -v.2\ \ -long\_code 7 4\$
10030 \quad \triangleright \quad "0,2,3"\$
10031 \quad \triangleright \quad "1,3,4"\$
10032 \quad \triangleright \quad "2,4,5"\$
10033 \quad \triangleright \quad "3,5,6"\$
10034 \quad \triangleright \quad $(\text{ORBITER\_PATH})\text{orbiter.out}\ \ -v.2\$
10035 \quad \triangleright \quad \text{-loop L 0 16 1 -draw\_matrix}\$
10036 \quad \triangleright \quad \text{-input csv file}\$
10037 \quad \triangleright \quad \text{long code genma n7 k4 codeword %L.csv}\$
10038 \quad \triangleright \quad \text{-box width 25 bit depth 8}\$
10039 \quad \triangleright \quad \text{-partition 3 4 2 end}\$
10040 \quad \text{-end loop}\$
10041
10042
10043
10044
10045
10046 #11111111 = 255
10047 #01010101 = 170
10048 #00110011 = 204
10049 #00001111 = 240

647
Golay23 code words:
\$\text{(ORBITER\ PATH)orbiter.out\ -v.2}\$
\$\text{-define\ v\ -vector\ -dense\ $(\text{GOLAY\ 23\ COLUMN\ RANKS\ PROJECTIVELY})\ -end}\$
\$\text{-linear\ code\ through\ columns\ of\ parity\ check\ projectively\ 12\ v}\$
\$\text{pdflatex\ code\ n23.k12.q2.tex}\$
\$\text{open-code\ n23.k12.q2.pdf}\$

Golay23 code diagram:
\$\text{(ORBITER\ PATH)orbiter.out\ -v.2}\$
\$\text{-code\ diagram\ from\ file\ "Golay\ 23\"}\$
\$\text{codewords\ n23.k12.q2.csv\ -enhance\ 4}\$

Golay23 code diagram draw:
\$\text{(ORBITER\ PATH)orbiter.out\ -v.2}\$
\$\text{-draw\ matrix}\$
\$\text{-input\ csv\ file\ Golay\ 23\ diagram\ 01.23.4096.csv}\$
\$\text{-box\ width\ 4\ -bit\ depth\ 8}\$
\$\text{-partition\ 20.4096.2048}\$

CRC codes:
encode_text_5bits:
  encode_text_5bits_check:
  encode_ constitution_field_induction:
  encode_text_5bits_1error:
encode_text_zero_5bits_1error:
$\$(ORBITER\_PATH)orbiter.out-\!v\cdot2\$
-defineF\cdot\text{finite Field}-q\cdot2\cdot\text{-end}\$
-withF\cdot\text{-do}\$
-finite_field_activity\$
-polynomial\_division\_from\_file\_all\_k\_bit\_error\_patterns\$
-defineF\cdot\text{finite field}-q\cdot2\cdot\text{-end}\$
-withF\cdot\text{-do}\$
-finite_field_activity\$
-polynomial\_division\_from\_file\_all\_k\_bit\_error\_patterns\_text\_zero.csv\$
13\cdot1\cdot\text{-end}\$
-pdflatex\_polynomial\_division\_file\_all\_1\_error\_patterns\_13.tex
-open-polynomial\_division\_file\_all\_1\_error\_patterns\_13.pdf
10146
10147 encode_text_zero_5bits_2error:
$\$(ORBITER\_PATH)orbiter.out-\!v\cdot2\$
-defineF\cdot\text{finite Field}-q\cdot2\cdot\text{-end}\$
-withF\cdot\text{-do}\$
-finite_field_activity\$
-polynomial\_division\_from\_file\_all\_k\_bit\_error\_patterns\_text\_zero.csv\$
13\cdot2\cdot\text{-end}\$
-pdflatex\_polynomial\_division\_file\_all\_2\_error\_patterns\_13.tex
-open-polynomial\_division\_file\_all\_2\_error\_patterns\_13.pdf
10155
10156 encode_text_zero_5bits_2error:
$\$(ORBITER\_PATH)orbiter.out-\!v\cdot2\$
-defineF\cdot\text{finite Field}-q\cdot2\cdot\text{-end}\$
-withF\cdot\text{-do}\$
-finite_field_activity\$
-polynomial\_division\_from\_file\_all\_k\_bit\_error\_patterns\_text\_zero.csv\$
13\cdot2\cdot\text{-end}\$
-pdflatex\_polynomial\_division\_file\_all\_2\_error\_patterns\_13.tex
-open-polynomial\_division\_file\_all\_2\_error\_patterns\_13.pdf
10164
10165 encode_text_5bits_2error:
$\$(ORBITER\_PATH)orbiter.out-\!encode_text_5bits\"Hithere\"\_text.csv\$
$\$(ORBITER\_PATH)orbiter.out-\!v\cdot2\$
-defineF\cdot\text{finite Field}-q\cdot2\cdot\text{-end}\$
-withF\cdot\text{-do}\$
-finite_field_activity\$
-polynomial\_division\_from\_file\_all\_k\_bit\_error\_patterns\_text.csv\$
13\cdot2\cdot\text{-end}\$
-pdflatex\_polynomial\_division\_file\_all\_2\_error\_patterns\_13.tex
-open-polynomial\_division\_file\_all\_2\_error\_patterns\_13.pdf
10174
10175 encode_text_5bits_error_0_7:
$\$(ORBITER\_PATH)orbiter.out-\!v\cdot2\$
-defineF\cdot\text{finite Field}-q\cdot2\cdot\text{-end}\$
-withF\cdot\text{-do}\$
-finite_field_activity\$
-polynomial\_division\_from\_file\_text\_error\_0_7.csv\$
13\cdot\text{-end}\$
-pdflatex\_polynomial\_division\_file\_13.tex
10180
open-polynomial_division_file_13.pdf

```
#algebra_global::find_CRC_polynomials.info=4.check=3.nb_sol=3
#0:
· 1101
#1:
· 1011
#2::1111

CRC_2.4.3:
$(ORBITER_PATH)orbiter.out -v -define F -finite_field -q 2 -end \ 
-define F -do -finite_field_activity \ 
-find_CRC_polynomials 2 4 3 -end

#algebra_global::find_CRC_polynomials.info=4.check=3.nb_sol=3
0:
· 1101
1:
· 1011
2::1111

CRC_2.4.4:
$(ORBITER_PATH)orbiter.out -define F -finite_field -q 2 -end \ 
-define F -do -finite_field_activity \ 
-find_CRC_polynomials 2 4 4 -end

CRC_2.5.4:
$(ORBITER_PATH)orbiter.out -define F -finite_field -q 2 -end \ 
-define F -do -finite_field_activity \ 
-find_CRC_polynomials 2 5 4 -end

CRC_2.6.4:
$(ORBITER_PATH)orbiter.out -define F -finite_field -q 2 -end \ 
-define F -do -finite_field_activity \ 
-find_CRC_polynomials 2 6 4 -end

CRC_2.7.4:
$(ORBITER_PATH)orbiter.out -define F -finite_field -q 2 -end \ 
-define F -do -finite_field_activity \ 
-find_CRC_polynomials 2 7 4 -end

CRC_2.8.4:
$(ORBITER_PATH)orbiter.out -v -2 -define F -finite_field -q 2 -end \ 
-define F -do -finite_field_activity \ 
-find_CRC_polynomials 2 8 4 -end
```
10275
10276 CRC_3_128_10:
10277 $\text{(ORBITER PATH)orbiter.out-v.1:\$
10278 \text{ }$-define F-finite field-q.2-end\$)
10279 \text{ }$-with F-do-finite field activity$
10280 \text{ }$-find CRC polynomials:3-128-10$
10281 \text{ }$-end$
10282
10283
10284
10285 CRC_2_256_8:
10286 $\text{(ORBITER PATH)orbiter.out-define F-finite field-q.2-end\$)
10287 \text{ }$-with F-do-finite field activity$
10288 \text{ }$-find CRC polynomials:2-256-8$
10289 \text{ }$-end$
10290
10291 # no solution
10292
10293 CRC_2_256_10:
10294 $\text{(ORBITER PATH)orbiter.out-define F-finite field-q.2-end\$)
10295 \text{ }$-with F-do-finite field activity$
10296 \text{ }$-find CRC polynomials:2-256-10$
10297 \text{ }$-end$
10298
10299
10300 # algebra_global::find CRC polynomials info=256 check=10 nb sol=368
10301 # 0:13 on Mac
10302
10303
10304 CRC_2_256_12:
10305 $\text{(ORBITER PATH)orbiter.out-v.1-define F-finite field-q.2-end\$)
10306 \text{ }$-with F-do-finite field activity$
10307 \text{ }$-find CRC polynomials:2-256-12$
10308 \text{ }$-end$
10309
10310
10311 # algebra_global::find CRC polynomials info=256 check=12 nb sol=2654
10312 # User time: 2:0
10313
10314 CRC_3_256_10:
10315 $\text{(ORBITER PATH)orbiter.out-define F-finite field-q.2-end\$)
10316 \text{ }$-with F-do-finite field activity$
10317 \text{ }$-find CRC polynomials:3-256-10$
10318 \text{ }$-end$
10319
10320
10321 # algebra_global::find CRC polynomials info=256 check=10 nb sol=140

653
# Orbiter·session·finished.
# User·time:·8:7

CRC_3_256_16:
$\$(ORBITER_PATH)orbiter.out-v:1-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-v:1-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-v:1-definеF--finite_field-q:2-end\$

CRC_3_8_4:
$\$(ORBITER_PATH)orbiter.out-v:2-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-v:2-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-v:2-definеF--finite_field-q:2-end\$

CRC_3_64_8:
$\$(ORBITER_PATH)orbiter.out-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-definеF--finite_field-q:2-end\$

CRC_poly_3_128_10:
$\$(ORBITER_PATH)orbiter.out-v:1-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-v:1-definеF--finite_field-q:2-end\$
$\$(ORBITER_PATH)orbiter.out-v:1-definеF--finite_field-q:2-end\$

# 12/26/2020: 243·polynomials·in·0:56·minutes·on·Mac

# Section·9.5: Coding·Theory·--·Reed·Muller·codes

SECTION_CODING_THEORY_REED_MULLER_CODES:
RM_3_1_code_words:

```bash
$ (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)

RM_3_1_Hamming_space_diagram:

```bash
$ (ORBITER_PATH) orbiter.out -v:2 -code_diagram"RM_3_1"-
  > $(REED_MULLER_3_1_CODEWORDS)-8-
  > -metric_balls-1
  > $(ORBITER_PATH) orbiter.out -v:2 -draw_matrix-
  > -input_csv_file RM_3_1_diagram_01_8_16.csv-
  > -box_width-25 -bit_depth-8 -partition-4:16:16-end
  > $(ORBITER_PATH) orbiter.out -v:2 -draw_matrix-
  > -input_csv_file RM_3_1_diagram_8_16.csv-
  > -box_width-25 -bit_depth-8 -partition-4:16:16-end
  > open-RM_3_1_diagram_8_16_draw.bmp
```

RM_3_1_hole0:

```bash
$ (ORBITER_PATH) orbiter.out -v:3-
```

RM_3_1_split:

```bash
$ (ORBITER_PATH) orbiter.out -split_by_values RM_3_1_holes_8_16.csv-
```

RM_3_1_holes_draw:

```bash
$ (ORBITER_PATH) orbiter.out -v:2-
  > -loop-L:0-3:1-
  > -draw_matrix-
  > -input_csv_file RM_3_1_holes_8_16.value%L.csv-
  > -box_width-25 -bit_depth-8 -partition-5:16:16-
  > -end-
  > -end_loop
```

RM_3_1_hole0:
10416  |  -define-F-finited-field-q2-end-
10417  |  -with-F-do-finited-field_activity-
10418  |  -algebraic_normal_form-RM_3_1_holes_8_16_value0.csv-v-8-end
10419
10420  #E_0+E_1+E_2+E_3+E_4
10421
10422
10423  RM_3_1_hole1:
10424  |  $(ORBITER_PATH)orborter.out-v-3-
10425  |  -define-F-finited-field-q2-end-
10426  |  -with-F-do-finited-field_activity-
10427  |  -algebraic_normal_form-RM_3_1_holes_8_16_value1.csv-v-8-end
10428
10429  #E_1=+X_0X_8^7:+X_1X_8^7:+X_2X_8^7:+X_3X_8^7:+X_4X_8^7:+X_5X_8^7:+X_6X_8^7
  +X_7X_8^7
10430
10431  RM_3_1_hole2:
10432  |  $(ORBITER_PATH)orborter.out-v-3-
10433  |  -define-F-finited-field-q2-end-
10434  |  -with-F-do-finited-field_activity-
10435  |  -algebraic_normal_form-RM_3_1_holes_8_16_value2.csv-v-8-end
10436
10437  #X0*X1*X8^6:+X0*X2*X8^6:+X0*X3*X8^6:+X0*X4*X8^6:+X0*X5*X8^6:+X0*X6*X8^6:+X0
  *X7*X8^6:+X1*X2*X8^6:+X1*X3*X8^6:+X1*X4*X8^6:+X1*X5*X8^6:+X1*X6*X8^6:+X1*X7
  *X8^6:+X2*X3*X8^6:+X2*X4*X8^6:+X2*X5*X8^6:+X2*X6*X8^6:+X2*X7*X8^6:+X3*X4*X8^6
  +X3*X5*X8^6:+X3*X6*X8^6:+X3*X7*X8^6:+X4*X5*X8^6:+X4*X6*X8^6:+X4*X7*X8^6
  +X5*X6*X8^6:+X5*X7*X8^6:+X6*X7*X8^6:+X0*X1*X2*X8^5:+X0*X1*X3*X8^5:+X0*X1*X4*X8^5
  +X0*X1*X5*X8^5:+X0*X1*X6*X8^5:+X0*X1*X7*X8^5:+X1*X2*X3*X8^5:+X1*X2*X4*X8^5
  +X1*X2*X5*X8^5:+X1*X2*X6*X8^5:+X1*X2*X7*X8^5:+X1*X3*X4*X8^5:+X1*X3*X5*X8^5
  +X1*X3*X6*X8^5:+X1*X3*X7*X8^5:+X1*X4*X5*X8^5:+X1*X4*X6*X8^5:+X1*X4*X7*X8^5
  +X1*X5*X6*X8^5:+X1*X5*X7*X8^5:+X1*X6*X7*X8^5:+X2*X3*X4*X8^5:+X2*X3*X5*X8^5
  +X2*X3*X6*X8^5:+X2*X3*X7*X8^5:+X2*X4*X5*X8^5:+X2*X4*X6*X8^5:+X2*X4*X7*X8^5
  +X2*X5*X6*X8^5:+X2*X5*X7*X8^5:+X2*X6*X7*X8^5:+X3*X4*X5*X8^5:+X3*X4*X6*X8^5
  +X3*X4*X7*X8^5:+X3*X5*X6*X8^5:+X3*X5*X7*X8^5:+X3*X6*X7*X8^5:+X4*X5*X6*X8^5
  +X4*X5*X7*X8^5:+X4*X6*X7*X8^5:+X5*X6*X7*X8^5:+X0*X1*X2*X3*X8^4:+X0*X1*X2*X4*X8^4
  +X0*X1*X3*X4*X8^4:+X0*X1*X3*X5*X8^4:+X0*X1*X3*X6*X8^4:+X0*X1*X3*X7*X8^4
  +X0*X1*X4*X5*X8^4:+X0*X1*X4*X6*X8^4:+X0*X1*X4*X7*X8^4:+X0*X2*X3*X4*X8^4
  +X0*X2*X3*X5*X8^4:+X0*X2*X3*X6*X8^4:+X0*X2*X3*X7*X8^4:+X0*X2*X4*X5*X8^4
  +X0*X2*X4*X6*X8^4:+X0*X2*X4*X7*X8^4:+X0*X3*X4*X5*X8^4:+X0*X3*X4*X6*X8^4
  +X0*X3*X4*X7*X8^4:+X0*X3*X5*X6*X8^4:+X0*X3*X5*X7*X8^4:+X0*X3*X6*X7*X8^4
  +X0*X3*X7*X8^4:+X1*X2*X3*X4*X8^4:+X1*X2*X3*X5*X8^4:+X1*X2*X3*X6*X8^4
  +X1*X2*X3*X7*X8^4:+X1*X2*X4*X5*X8^4:+X1*X2*X4*X6*X8^4:+X1*X2*X4*X7*X8^4
  +X1*X3*X4*X5*X8^4:+X1*X3*X4*X6*X8^4:+X1*X3*X4*X7*X8^4:+X1*X3*X5*X6*X8^4
  +X1*X3*X5*X7*X8^4:+X1*X3*X6*X7*X8^4:+X1*X3*X7*X8^4:+X1*X4*X5*X6*X8^4
  +X1*X4*X5*X7*X8^4:+X1*X4*X6*X7*X8^4:+X1*X4*X7*X8^4:+X1*X5*X6*X7*X8^4
  +X1*X5*X7*X8^4:+X1*X6*X7*X8^4:+X1*X7*X8^4
656
4*X5*X6*X8^4 + X1*X4*X6*X7*X8^4 + X1*X5*X6*X7*X8^4 + X2*X3*X4*X6*X8^4 + X2*X3*X5*X6*X8^4 + X2*X3*X5*X7*X8^4 + X2*X4*X5*X6*X8^4 + X3*X4*X5*X7*X8^4 + X3*X4*X6*X7*X8^4 + X3*X5*X6*X7*X8^4

10438 # E_2+E_3+E_4
10440
10441
10442 RM_4_1:
10443 \>$\text{ORBITER\_PATH}/\text{orbiter.out}-v:2\$
10444 \>$\text{linear\_code\_through\_columns\_of\_parity\_check-5}$\$
10445 \>$\text{REED\_MULLER\_4\_1\_COLUMNS\_OF\_PARTITY\_CHECK}$\$
10446 \>$\text{pdflatex\_code\_n16\_k5\_q2.tex}$\$
10447 \>$\text{open\_code\_n16\_k5\_q2.pdf}$\$
10448
10449
10450 # codewords\_n16\_k5\_q2.csv
10451
10452
10453
10454 RM_4_1\_diagram:
10455 \>$\text{ORBITER\_PATH}/\text{orbiter.out}-v:2\$
10456 \>$\text{-code\_diagram\_from\_file:\"RM_4_1\"}$\$
10457 \>$\text{codewords\_n16\_k5\_q2.csv-16}$\$
10458 \>$\text{#-enhance-4}$\$
10459 \>$\text{#-metric\_balls-3}$\$
10460
10461 RM_4_1\_diagram\_draw:
10462 \>$\text{ORBITER\_PATH}/\text{orbiter.out}-v:2\text{-draw\_matrix}$\$
10463 \>$\text{-input\_csv\_file:RM_4_1\_diagram\_01\_16\_32.csv}$\$
10464 \>$\text{-box\_width-25\text{-bit\_depth-8\text{-partition-10\text{-256\\text{-256}}\text{-end}}}$\$
10465 \>$\text{open\_RM\_4\_1\_diagram\_01\_16\_32\_draw.bmp}$\$
10466
10467
10468 RM_4_1\_split:
10469 \>$\text{ORBITER\_PATH}/\text{orbiter.out}-\text{split\_by\_values\_RM_4_1\_holes\_16\_32.csv}$\$
10470
10471 RM_4_1\_diagram\_draw\_holes:
10472 \>$\text{ORBITER\_PATH}/\text{orbiter.out}-v:2\text{-draw\_matrix}$\$
10473 \>$\text{-input\_csv\_file:RM_4_1\_holes\_16\_32.csv}$\$
10474 \>$\text{-box\_width-25\text{-bit\_depth-8\text{-partition-10\text{-256\\text{-256}}\text{-end}}}$\$
10475 \>$\text{ORBITER\_PATH}/\text{orbiter.out}-v:2\text{-loop-L-0-7-1\text{-draw\_matrix}}$\$
10476 \>$\text{-input\_csv\_file:RM_4_1\_holes\_16\_32\_value\_L.csv}$\$
10477 \>$\text{-box\_width-25\text{-bit\_depth-8\text{-partition-10\text{-256\\text{-256}}\text{-end\_end\_loop}}}$\$
10478
10479
10480
657
10481 RM_4_1_diagram_metric_spheres:
10482 ➤ -$(ORBITER\_PATH)orbiner.out\ -v.2\ -code\_diagram\_from\_file:\ \"RM_4_1\"\ :\ \
10483 ➤ ▶ codewords_n16_k5_q2.csv\ 16\ -metric\_balls\ :\ 
10484 ➤ $(ORBITER\_PATH)orbiner.out\ -v.2\ -draw\_matrix\ :\ 
10485 ➤ ▶ -input\_csv\_file\ RM_4_1\_diagram_16_32.csv\ 
10486 ➤ ▶ -box\_width\ :\ 25\ -bit\_depth\ :\ 8\ -partition\ :\ 10.256.256\ -end
10487
10488
10489
10490 RM_4_1_hole0:
10491 ➤ $(ORBITER\_PATH)orbiner.out\ -v.3\ -define\_F\ -finite\_field\ -q.2\ -end\ :\ 
10492 ➤ ▶ -with\_F\ -do\ -finite\_field\_activity\ :\ 
10493 ➤ ▶ -algebraic\_normal\_form\ RM_4_1\_holes_16_32\_value0.csv\ 16\ -end
10494
10495
10496
10497
10498
10499
10500
10501 Reed\_Muller\_6:
10502 ➤ $(ORBITER\_PATH)orbiner.out\ -v.2\ -long\_code\_64.7\ :\ 
10503 ➤ ▶ -set\_builder\ -loop\ :\ 0.64.1\ -end\ :\ 
10504 ➤ ▶ -set\_builder\ -loop\ :\ 0.32.1\ -affine\_function\ :\ 2.1\ -end\ :\ 
10505 ➤ ▶ -set\_builder\ -loop\ :\ 0.16.1\ -affine\_function\ :\ 4.2\ -clone\_with\_affine\_function\ :\ 4.3\ -end\ :\ 
10506 ➤ ▶ -set\_builder\ -set\_builder\ -set\_builder\ -loop\ :\ 0.4.1\ -affine\_function\ :\ 1.4\ :\ 
10507 ➤ ▶ ▶ -clone\_with\_affine\_function\ :\ 1.12\ -end\ -clone\_with\_affine\_function\ :\ 1.16\ :\ 
10508 ➤ ▶ ▶ -end\ -clone\_with\_affine\_function\ :\ 1.32\ -end\ :\ 
10509 ➤ ▶ -set\_builder\ -set\_builder\ -loop\ :\ 0.8.1\ -affine\_function\ :\ 1.8\ :\ 
10510 ➤ ▶ ▶ -clone\_with\_affine\_function\ :\ 1.24\ -end\ -clone\_with\_affine\_function\ :\ 1.32\ -end\ :\ 
10511 ➤ ▶ ▶ -set\_builder\ -loop\ :\ 0.16.1\ -affine\_function\ :\ 1.16\ -clone\_with\_affine\_function\ :\ 1.48\ -end\ :\ 
10512 ➤ ▶ -set\_builder\ -loop\ :\ 0.32.1\ -affine\_function\ :\ 1.32\ -end\ :\ 
10513 ➤ $(ORBITER\_PATH)orbiner.out\ -v.2\ -draw\_matrix\ :\ 
10514 ➤ ▶ -input\_csv\_file\ long\_code\_genma_n64_k7.csv\ 
10515 ➤ ▶ -box\_width\ :\ 25\ -bit\_depth\ :\ 8\ -partition\ :\ 3.7.64\ -end
10516 ➤ open\_long\_code\_genma_n64_k7\_draw.bmp
10517
10518
10519
10520
10521
10522 RM_6:
10523 ➤ $(ORBITER\_PATH)orbiner.out\ -v.2\ -long\_code\_64.7\ :\ 
10524 ➤ ▶ $(RM_6\_GENERATOR_1)\ :\
10525  ▶  ▶  $(RM_6\_GENERATOR_2)\\
10526  ▶  ▶  $(RM_6\_GENERATOR_3)\\
10527  ▶  ▶  $(RM_6\_GENERATOR_4)\\
10528  ▶  ▶  $(RM_6\_GENERATOR_5)\\
10529  ▶  ▶  $(RM_6\_GENERATOR_6)\\
10530  ▶  ▶  $(RM_6\_GENERATOR_7)\\
10531  ▶  ▶  $(ORBITER\_PATH)orbiter.out-v.2-draw_matrix\\
10532  ▶  ▶  -input_csv_file-long_code_genma_n64_k7.csv\\
10533  ▶  ▶  -box_width-25-bit_depth-8-partition-3:7:64-end\\
10534  ▶  ▶  $(ORBITER\_PATH)orbiter.out-v.2-draw_matrix\\
10535  ▶  ▶  -input_csv_file-long_code_genma_n64_k7_codeword_0.csv\\
10536  ▶  ▶  -box_width-25-bit_depth-8-partition-3:8:8-end\\
10537  ▶  ▶  $(ORBITER\_PATH)orbiter.out-v.2-draw_matrix\\
10538  ▶  ▶  -input_csv_file-long_code_genma_n64_k7_codeword_1.csv\\
10539  ▶  ▶  -box_width-25-bit_depth-8-partition-3:8:8-end\\
10540  ▶  ▶  $(ORBITER\_PATH)orbiter.out-v.2-draw_matrix\\
10541  ▶  ▶  -input_csv_file-long_code_genma_n64_k7_codeword_2.csv\\
10542  ▶  ▶  -box_width-25-bit_depth-8-partition-3:8:8-end\\
10543
10544
10545
10546
10547  RM6words:
10548  ▶  -mkdir-RM6
10549  ▶  #$\$(ORBITER\_PATH)orbiter.out-v.2-draw_matrix-input_csv_file-long_code_genma_n64_k7_codeword_0.csv-box_width-25-bit_depth-8-partition-4:8:8-end
10550  ▶  $(ORBITER\_PATH)orbiter.out-v.2-loop-L0:128:1\\
10551  ▶  ▶  -draw_matrix-input_csv_file\\
10552  ▶  ▶  ▶  long_code_genma_n64_k7_codeword_\%L.csv\\
10553  ▶  ▶  ▶  -box_width-25-bit_depth-8-partition-4:8:8-end\\
10554  ▶  ▶  ▶  -mv-long_code_genma_n64_k7_codeword_\%L_draw.bmp\\
10555  ▶  ▶  ▶  RM6/RM_6_1_codeword_\%L.bmp\\
10556  ▶  ▶  -end_loop
10557
10558
10559
10560  RM6_convert:
10561  ▶  -mkdir-RM6_PNG
10562  ▶  ▶  convert-RM6/RM_6_1_codeword_0.bmp-frame8-RM6_PNG/000.png
10563  ▶  ▶  convert-RM6/RM_6_1_codeword_1.bmp-frame8-RM6_PNG/001.png
10564  ▶  ▶  convert-RM6/RM_6_1_codeword_2.bmp-frame8-RM6_PNG/002.png
10565  ▶  ▶  convert-RM6/RM_6_1_codeword_3.bmp-frame8-RM6_PNG/003.png
10566  ▶  ▶  convert-RM6/RM_6_1_codeword_4.bmp-frame8-RM6_PNG/004.png
10567  ▶  ▶  convert-RM6/RM_6_1_codeword_5.bmp-frame8-RM6_PNG/005.png
10568  ▶  ▶  convert-RM6/RM_6_1_codeword_6.bmp-frame8-RM6_PNG/006.png
10569  ▶  ▶  convert-RM6/RM_6_1_codeword_7.bmp-frame8-RM6_PNG/007.png
10570  ▶  ▶  convert-RM6/RM_6_1_codeword_8.bmp-frame8-RM6_PNG/008.png

659
RM_6_poster:


Section 9.6: Coding Theory -- BCH Codes

draw_cyclotomic_mod_21_q8:
$($ORBITER_PATH)orbiter.out\-v.2\$
$-\text{draw}_\text{options}\$
$-\text{radius}\cdot100\$
$-\text{line}\_\text{width}\cdot1.0\cdot\text{embedded}\$
$-\text{end}\$
$-\text{draw}_\text{mod}_{n-n.21}\text{--file}\text{mod}_{21}\_\text{cyclotomic}\$
$-\text{cyclotomic}\_\text{sets}\cdot8\cdot1,2,4,5,7,10,13\text{--end}\$
$\text{pdflatex}\_\text{mod}_{21}\_\text{cyclotomic}\_\text{draw}.\text{tex}$
$\text{open}\_\text{mod}_{21}\_\text{cyclotomic}\_\text{draw}.\text{pdf}$

F.8_BCH_code_d3:
$($ORBITER_PATH)orbiter.out\-v.3\$
$-\text{define}\_\text{finite}_\text{field}\cdot-q.8\cdot\text{override}_\text{polynomial}\cdot11\text{--end}\$
$-\text{with}\_\text{do}\_\text{finite}_\text{field}_\text{activity}\text{--make}\_\text{BCH}\_\text{code}_{21,3}\text{--end}$
$\text{pdflatex}\_\text{BCH}\_\text{codes}_{q.8}_{n21}_{d3}.\text{tex}$
10735 \triangleright open-BCH_codes_q8_n21_d3.pdf
10736
10737 \#generator-polynomial.is.X^{-\{4\}}+4X^{-\{3\}}+4X^{-\{2\}}+\cdot \cdot \cdot +1
10738
10739 F_8.BCH_code_d4:
10740 \triangleright $(\text{ORBITER}\_\text{PATH})\text{orbiter.out} -v.3:\$
10741 \triangleright \triangleright -\text{define\text{-}}\text{finite\text{-}field} -q.8 -\text{override\text{-}polynomial.11\text{-}end}\$
10742 \triangleright \triangleright -\text{with\text{-}F\text{-}do\text{-}finite\text{-}field\_activity} -\text{make\_BCH\_code.21\_4\text{-}end}\$
10743
10744 \#generator-polynomial.is.X^{-\{5\}}+6X^{-\{4\}}+\cdot \cdot \cdot +\cdot \cdot \cdot +1
10745
10746
10747 F_8.BCH_code_d5:
10748 \triangleright $(\text{ORBITER}\_\text{PATH})\text{orbiter.out} -v.3:\$
10749 \triangleright \triangleright -\text{define\text{-}finite\text{-}field} -q.8 -\text{override\text{-}polynomial.11\text{-}end}\$
10750 \triangleright \triangleright -\text{with\text{-}F\text{-}do\text{-}finite\text{-}field\_activity} -\text{make\_BCH\_code.21\_5\text{-}end}\$
10751 \triangleright pdflatex -BCH_codes_q8_n21_d5.tex
10752 \triangleright open-BCH_codes_q8_n21_d5.pdf
10753
10754 \#generator-polynomial.is.X^{-\{7\}}+3X^{-\{6\}}+\cdot \cdot \cdot +2X^{-\{2\}}+\cdot \cdot \cdot +1
2
10755
10756 F_8.BCH_code_d6:
10757 \triangleright $(\text{ORBITER}\_\text{PATH})\text{orbiter.out} -v.3:\$
10758 \triangleright \triangleright -\text{define\text{-}finite\text{-}field} -q.8 -\text{override\text{-}polynomial.11\text{-}end}\$
10759 \triangleright \triangleright -\text{with\text{-}F\text{-}do\text{-}finite\text{-}field\_activity} -\text{make\_BCH\_code.21\_6\text{-}end}\$
10760
10761
10762 \#generator-polynomial.is.X^{-\{9\}}+5X^{-\{8\}}+\cdot \cdot \cdot +4X^{-\{3\}}+\cdot \cdot \cdot +1
10763
10764 F_8.BCH_code_d7:
10765 \triangleright $(\text{ORBITER}\_\text{PATH})\text{orbiter.out} -v.3:\$
10766 \triangleright \triangleright -\text{define\text{-}finite\text{-}field} -q.8 -\text{override\text{-}polynomial.11\text{-}end}\$
10767 \triangleright \triangleright -\text{with\text{-}F\text{-}do\text{-}finite\text{-}field\_activity} -\text{make\_BCH\_code.21\_7\text{-}end}\$
10768 \triangleright pdflatex -BCH_codes_q8_n21_d7.tex
10769 \triangleright open-BCH_codes_q8_n21_d7.pdf
10770
10771 \#generator-polynomial.is.X^{-\{10\}}+X^{-\{9\}}+\cdot \cdot \cdot +2X^{-\{8\}}+\cdot \cdot \cdot +6X^{-\{3\}}+\cdot \cdot \cdot +1
2
10772
10773 F_8.BCH_code_d8:
10774 \triangleright $(\text{ORBITER}\_\text{PATH})\text{orbiter.out} -v.3:\$
10775 \triangleright \triangleright -\text{define\text{-}finite\text{-}field} -q.8 -\text{override\text{-}polynomial.11\text{-}end}\$
10776 \triangleright \triangleright -\text{with\text{-}F\text{-}do\text{-}finite\text{-}field\_activity} -\text{make\_BCH\_code.21\_8\text{-}end}\$
10777
10778 \#generator-polynomial.is.X^{-\{12\}}+2X^{-\{10\}}+\cdot \cdot \cdot +6X^{-\{9\}}+\cdot \cdot \cdot +7X^{-\{7\}}+\cdot \cdot \cdot +6X^{-\{6\}}+\cdot \cdot \cdot +2
X^{-\{5\}}+\cdot \cdot \cdot +5X^{-\{3\}}+\cdot \cdot \cdot +1
0
#k=9

# after reduction: \([63,27]_2 \cdot r=36\)

10779

10780

10781

10782

10783

10784

10785 F2_BCH_code_n21:

10786 \$(\text{ORBITER_PATH})\text{orbiter.out}-v.3\$

10787 \$\text{-define-F:--finite_field-q.2-end}\$

10788 \$\text{-with-F:--do--finite_field_activity}\$

10789 \$\text{-make_BCH_code:21.3}\$

10790 \$\text{-end}\$

10791

10792

10793 F7_RS_code_n6:

10794 \$(\text{ORBITER_PATH})\text{orbiter.out}-v.30\$

10795 \$\text{-define-F:--finite_field-q.7-end}\$

10796 \$\text{-with-F:--do--finite_field_activity}\$

10797 \$\text{-make_BCH_code:6.3}\$

10798 \$\text{-end}\$

10799

10800

10801 F_64_again:

10802 \$(\text{ORBITER_PATH})\text{orbiter.out}-v.3\$

10803 \$\text{-define-F:--finite_field-q.64-end}\$

10804 \$\text{-with-F:--do--finite_field_activity}\$

10805 \$\text{-cheat_sheet_GF}\$

10806 \$\text{-end}\$

10807 \$\text{pdflatex-GF_64.tex}\$

10808 \$\text{open-GF_64.pdf}\$

10809

10810

10811 BCH_255_5_evaluate_elementary_symmetric_functions_1:

10812 \$(\text{ORBITER_PATH})\text{orbiter.out}-v.3\text{-define-F:--finite_field-q.256-end}\$

10813 \$\text{-define-e1--formula\"e1\"\"e1\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_1)\$}\$

10814 \$\text{-define-e2--formula\"e2\"\"e2\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_2)\$}\$

10815 \$\text{-define-e3--formula\"e3\"\"e3\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_3)\$}\$

10816 \$\text{-define-e4--formula\"e4\"\"e4\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_4)\$}\$

10817 \$\text{-define-e5--formula\"e5\"\"e5\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_5)\$}\$

10818 \$\text{-define-e6--formula\"e6\"\"e6\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_6)\$}\$

10819 \$\text{-define-e7--formula\"e7\"\"e7\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_7)\$}\$

10820 \$\text{-define-e8--formula\"e8\"\"e8\"\"$(\text{ELEMENTARY_SYMME}\text{TRIC}_8_8)\$}\$

10821 \$\text{-define-E8--collection\"e1,e2,e3,e4,e5,e6,e7,e8\"}\$

10822 \$\text{-with-F:--do--finite_field_activity}\$

10823 \$\text{-evaluate-E8\"x0=2,x1=4,x2=16,x3=29,x4=76,x5=157,x6=95,x7=133\"-end}\$

10824

10825 BCH_255_5_evaluate_elementary_symmetric_functions_2:
$(ORBITER \ PATH)orbiter.out -v.3 -define F=finite_field -q.256 -end \n-define e1= formula "e1"."e1"."$(ELEMENTARY_SYMMETRIC_8.1)\n-define e2= formula "e2"."e2"."$(ELEMENTARY_SYMMETRIC_8.2)\n-define e3= formula "e3"."e3"."$(ELEMENTARY_SYMMETRIC_8.3)\n-define e4= formula "e4"."e4"."$(ELEMENTARY_SYMMETRIC_8.4)\n-define e5= formula "e5"."e5"."$(ELEMENTARY_SYMMETRIC_8.5)\n-define e6= formula "e6"."e6"."$(ELEMENTARY_SYMMETRIC_8.6)\n-define e7= formula "e7"."e7"."$(ELEMENTARY_SYMMETRIC_8.7)\n-define e8= formula "e8"."e8"."$(ELEMENTARY_SYMMETRIC_8.8)\n-with F= do=- finite_field_activity \n-evaluate E8: "x0=8, x1=64, x2=205, x3=143, x4=70, x5=217, x6=130, x7=23" -end

BCH15:
#$(ORBITER \ PATH) orbiter.out - BCH 15 2 3
#$(ORBITER \ PATH) orbiter.out - BCH 15 2 5
#$(ORBITER \ PATH) orbiter.out - BCH 15 2 7
#$(ORBITER \ PATH) orbiter.out - BCH 15 2 9

BCH11:
$(ORBITER \ PATH) orbiter.out - BCH 11 2 3
$(ORBITER \ PATH) orbiter.out - BCH 11 2 5
$(ORBITER \ PATH) orbiter.out - BCH 11 2 7

BCH13:
$(ORBITER \ PATH) orbiter.out - BCH 13 2 3
$(ORBITER \ PATH) orbiter.out - BCH 13 2 5
$(ORBITER \ PATH) orbiter.out - BCH 13 2 7

BCH7:
$(ORBITER \ PATH) orbiter.out - BCH 7 2 3
$(ORBITER \ PATH) orbiter.out - BCH 7 2 5

BCH21:
$(ORBITER \ PATH) orbiter.out - BCH 21 2 3
$(ORBITER \ PATH) orbiter.out - BCH 21 2 5
$(ORBITER \ PATH) orbiter.out - BCH 21 2 7
$(ORBITER \ PATH) orbiter.out - BCH dual 21 2 7

BCH93:
$(ORBITER \ PATH) orbiter.out - BCH 93 2 3

BCH255:
$(ORBITER_PATH)orbiter.out -BCH 255\cdot2\cdot4
$(ORBITER_PATH)orbiter.out --v.2 -draw_matrix -input_csv_file=BCH_255_4.csv
$ -box_width=40 -bit_depth=24
$ -partition=10 "239" "255" -end

#BCH_255_4.csv

BCH273:
$(ORBITER_PATH)orbiter.out -BCH 273\cdot2\cdot4

draw_mod_31:
$(ORBITER_PATH)orbiter.out --v.2
$ -draw_options -embedded -end
$ -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
$ -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
$ -draw_mod_n 251 -mod_251
$ 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
\input{draw.tex}
\input{mod255_cyclotomic_1_draw.tex}
\input{mod255_cyclotomic_3_draw.tex}
\input{mod63_4_cyclotomic_3_6_draw.tex}
\input{GF64.tex}
\input{bch_elementary_symmetric_functions_3.tex}
evaluate elementary symmetric functions 1:

```plaintext
define F finite field q 64 end

define e1 formula "e1" end
define e2 formula "e2" end
define e3 formula "e3" end

define E3 collection "e1,e2,e3"

with F do finite field activity

evaluate E3 "x0=8,x1=62,x2=15" end
```

# The values of the formulae are:

0: 57
1: 0
2: 1

# poly: 1,0,2,1

BCH_63_4_evaluate_elementary_symmetric_functions_1:

```plaintext
define F finite field q 64 end

define e1 formula "e1" end
define e2 formula "e2" end
define e3 formula "e3" end

define E3 collection "e1,e2,e3"

with F do finite field activity

evaluate E3 "x0=33,x1=45,x2=52" end
```

# The values of the formulae are:

0: 56
1: 0
2: 1

# poly: 1,0,3,1

BCH_63_4_evaluate_elementary_symmetric_functions_2:

```plaintext
BCH_21_poly_mult_mod_F4:

define F finite field q 4 end

polynomial mult mod "1,0,2,1" "1,0,3,1"
```

# C(X)=X^6 + X^5 + X^4 + X^2 + 1
poly·1,0,1,0,1,1,1

BCH\_21\_poly\_division\_a:

1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1".\-

BCH\_21\_poly\_division\_b:

1,0,2,1".\-

BCH\_21\_poly\_division\_ab:

1,0,1,0,1,1,1".\-

BCH\_21\_generator\_matrix:

1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1".\-

BCH\_21\_15\_weight\_enumerator:

"1,0,1,0,1,1,1".\-

$(ORBITER\_PATH)orbiter.out\-v.2\-

-define\-F\-finite\_field\-q.4\-end\-

-define\-F\-do\-

-finite\_field\_activity\-

-polynomial\_division\-

-define\-v\-vector\-field\F\-format.15\-

-dense\$(BCH\_21\_15\_GENERATOR\_MATRIX).\-
11061 \triangleright \triangleright \ -end
11062 \triangleright \triangleright \ -with-F.-do-
11063 \triangleright \triangleright \ -finite_field_activity.-weight Enumerator.v.-end
11064
11065 \# too slow!
11066
11067 BCH_21_15_dual:
11068 \$(\texttt{ORBITER_PATH})\texttt{orbiter.out} \texttt{-v.2} \texttt{-} \\
11069 \triangleright \triangleright \ -define-F.-finite_field-q.4.-end-
11070 \triangleright \triangleright \ -define-v.-vector.-field.F.-format.15-
11071 \triangleright \triangleright \ -do \triangleright -dense-$(\texttt{BCH_21_15 GENERATOR MATRIX})\texttt{-} -end-
11072 \triangleright \triangleright \ -with-F.-do.-finite_field_activity-
11073 \triangleright \triangleright \ -nullspace-v-
11074 \triangleright \triangleright \ -normalize_from_the_right-
11075 \triangleright \triangleright \ -end
11076
11077
11078 BCH_21_6_weight Enumerator:
11079 \$(\texttt{ORBITER_PATH})\texttt{orbiter.out} \texttt{-v.2} \texttt{-} \\
11080 \triangleright \triangleright \ -define-F.-finite_field-q.4.-end-
11081 \triangleright \triangleright \ -define-v.-vector.-format.6.-field.F.-
11082 \triangleright \triangleright \ -do \triangleright -dense-$(\texttt{BCH_21_6 GENERATOR MATRIX})\texttt{-} -end-
11083 \triangleright \triangleright \ -with-F.-do-
11084 \triangleright \triangleright \ -finite_field_activity.-weight Enumerator.v.-end
11086
11087 \# $1y^{(21)}+63x^8y^{(13)}+294x^{(12)}y^9+756x^{(14)}y^7+1890x^{(16)}y^5+1092x^{(18)}y^3$
11088
11089 \# $(1,0,0,0,0,0,0,0,0,63,0,0,0,0,294,0,0,0,756,0,0,1890,0,0,1092,0,0,0)$
11090
11091
11092
11093 BCH_21_6_4_macwilliams:
11094 \$(\texttt{ORBITER_PATH})\texttt{orbiter.out} \texttt{-v.2} \texttt{-make_macwilliams_system} \texttt{21} \texttt{6} \texttt{4}
11095 \texttt{pdf to latex: \texttt{MacWilliams_n21_k6_q4.tex}}
11096 \texttt{open: \texttt{MacWilliams_n21_k6_q4.pdf}}
11097
11098
11099
11100 \# \texttt{ww := [1,0,0,0,84,252,1575,10080,58032,319662,141224,5133744,15282792,}\\
11101 \cdot 37951620,79336530,135622080,190615824,213273081,188911548,125744304,59721}\\
11102 \cdot 732,17767512,2580255]$
11103
11104
11105
11106
11107
11108
11109
11110
11111
671
11105 \> \> -define-F=finite_field\=-q4\=end\;
11106 \> \> -with-F=do\;
11107 \> \> -finite_field_activity\;
11108 \> \> -field_reduction:\"BCH\_21\_15\_4\".2\:15\:21\:$(BCH\_21\_15)$\:\end 
11109 \> \> $(ORBITER\_PATH)\\biter.\out=--v2\:\;
11110 \> \> -draw_matrix=--input_csv_file=\bCH\_21\_15\_4.csv\;
11111 \> \> -box_width=--bit_depth=24\;
11112 \> \> -partition=4\:"30\:"42\"=end 
11113 \> pdflatex\\field_reduction\_Q4\_q2\_15\_21.tex 
11114 \> open-BCH\_21\_15\_4\_draw.bmp 
11115 \> \#open-field_reduction\_Q4\_q2\_15\_21.pdf 
11116
11117 \#poly\=of\=degree=12:1,0,1,0,1,0,0,1,0,0,0,1 
11118
11119 BCH\_21\_poly\\_division\_c: 
11120 \> $(ORBITER\_PATH)\\biter.\out=--v2\:\;
11121 \> \> -define-F=finite_field\=-q2\=end\;
11122 \> \> -with-F=do\;
11123 \> \> -finite_field_activity\;
11124 \> \> -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 
11125 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\".\"1,0,1,0,1,0,0,1,1,0,0,0,1\"=end 
11126
11127 F16\\_roots\_5: 
11128 \> $(ORBITER\_PATH)\\biter.\out=--v3\:\;
11129 \> \> -define-F=finite_field\=-q2\=end\;
11130 \> \> -with-F=do\=finite_field_activity\=nth\roots\_5\=end 
11131
11132
11133
11134 F64\\_roots\_21: 
11135 \> $(ORBITER\_PATH)\\biter.\out=--v3\:\;
11136 \> \> -define-F=finite_field\=-q2\=end\;
11137 \> \> -with-F=do\=finite_field_activity\=nth\roots\_21\=end 
11138
11139
11140
11141 F256\\_roots\_771: 
11142 \> $(ORBITER\_PATH)\\biter.\out=--v3\:\;
11143 \> \> -define-F=finite_field\=-q256\=end\;
11144 \> \> -with-F=do\=finite_field_activity\=nth\roots\_771\=end 
11145
11146
11147
11148 F256\_BCH\_code\_d16: 
11149 \> $(ORBITER\_PATH)\\biter.\out=--v3\:\;
11150 \> \> -define-F=finite_field\=-q256\=end\;

672
#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_CODE

```
$ORBITER_PATH$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\~
-define F=finite_field Activity\~
-polynomial_division A B -end\~

F256_CODE

```

F256_CODE

```
$ORBITER_PATH$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\~
-define F=finite_field Activity\~
-define F=finite_field Activity\~
-write code for division check q 256 n 771 r 30.cpp A B -end

```

#Section 9.7: Coding Theory - Reed-Solomon codes

673
11194 SECTION_CODING_THEORY_REED_SOLOMON_CODES:
11195
11196
11197 #ToDo:
11198 ⊿
11199 F_7_BCH_code_n6:
11200 ⊿ $(ORBITER_PATH)orbiter.out \-v\-3:\n11201 ⊿ ⊿ -define F\-finite_field\-q\-7\-end:\n11202 ⊿ ⊿ -with F\-do\-finite_field_activity\-make_BCH_code\-7\-3\-end
11203
11204
11205
11206 RREF_RS_6_4_7_weight_enumerator:
11207 ⊿ $(ORBITER_PATH)orbiter.out \-v\-2:\n11208 ⊿ ⊿ -define F\-finite_field\-q\-7\-end:\n11209 ⊿ ⊿ -define v\-vector\-format\-4\-field F\-end:
11210 ⊿ ⊿ ⊿ -compact $(CODE_RS_6_4_7)\-end:\n11211 ⊿ ⊿ ⊿ -with F\-do\-finite_field_activity\-end\n11212 ⊿ ⊿ ⊿ -weight Enumerator v\-end:
11213 ⊿ ⊿ ⊿ -end
11214
11215
11216
11217
11218
11219
11220
11221
11222
11223 Code_RS_11:
11224 ⊿ $(ORBITER_PATH)orbiter.out \-v\-2:\n11225 ⊿ ⊿ -define F\-finite_field\-q\-11\-end:\n11226 ⊿ ⊿ -define v\-vector\-format\-8\-field F\-end:
11227 ⊿ ⊿ ⊿ -compact $(CODE_RS_10_8_11)\-end:\n11228 ⊿ ⊿ ⊿ -end:\n11229 ⊿ ⊿ ⊿ -with F\-do\-finite_field_activity\-end\n11230 ⊿ ⊿ ⊿ -weight Enumerator RREF v\-end
11231 ⊿ pdfLaTeX RREF_example_q11_8_10.tex
11232 ⊿ #gs \-sDEVICE=png16 \-dFIXEDMEDIA \-dDEVICEWIDTHPOINTS=500 \-dDEVICEHEIGHTPOINTS=450:\n11233 ⊿ ⊿ \# -r240 \-o RREF_example_q11_8_10_page%02d.png\n11234 ⊿ ⊿ # RREF_example_q11_8_10.pdf
11235 ⊿ ⊿ open RREF_example_q11_8_10.pdf
11236
11237
11238 Code_RS_11_weight_enumerator:
11239 ⊿ $(ORBITER_PATH)orbiter.out \-v\-2:\n
```
-define F -finite_field -q 11 -end \\
-define v -vector -format 8 -field F.\ 
-compact $(CODE_RS_11_RREF)\ 
-end\ 
-with F -do\ 
-finite_field_activity\ 
-weight Enumerator -v\ 
-end

# the group cannot be computed

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

RS 8 field Reduction:
$(ORBITER_PATH) orbiter.out -v 2 \ 
-define F -finite_field -q 8 -end \ 
-with F -do\ 
-finite_field_activity\ 
-field Reduction "RS 8 red 2" \ 
-2.5.7$(CODE_RS 8)\ 
-end

$ (ORBITER_PATH) orbiter.out -v 2 \ 
-dr...
```bash
-define v-vector-format 15-field F\n-compact $(RS_8_reduced)\n-end\n-with F-do\n-finite_field_activity\n-weight Enumerator-v\n-end

CODE_21_15_4 store:
$(ORBITER_PATH)orbiter.out -v 2-
-store_as_csv_file "code_21_15_4.csv"-
15 21 $(CODE_21_15_4)\n$(ORBITER_PATH)orbiter.out -v 2 -draw_matrix-
-input_csv_file "code_21_15_4.csv"-
-box_width 40 -bit_depth 24-
-partition 4 "15" "21"-
-end

CODE_21_15_4 weight Enumerator:
$(ORBITER_PATH)orbiter.out -v 2-
-define F-finite_field -q 2 -end-
-define v-vector-format 15-field F-
compact $(CODE_21_15_4)\n-end\n-with F-do\n-finite_field_activity\n-weight Enumerator-v\n-end

Reed solomon F8 work:
$(ORBITER_PATH)orbiter.out -v 3 -define F-finite_field -q 8 -end-
-with F-do-finite_field_activity-
-parse_and_evaluate "test" "(t-a)*(t-a*a)" "a=2" -end

SECTION CODING THEORY BOUNDS:

bounds for d given n6 k4 q7:
$(ORBITER_PATH)orbiter.out -v 2-
```

676
-make_bounds_for_d_given_n_and_k_and_q:6.4.7

bounds_for_d_given_n15_k6_q2:

$(ORBITER_PATH)orbiter.out-v.2:

-define F:finite_field-q:2:-end

-define P:projective_space:8:F:-end

-with P:-do

-projective_space_activity:

-make_gilbert_varshamov_code:15.5:

-end

 GV_n15_k6_d5:

$(ORBITER_PATH)orbiter.out-v.2:

 GV_n15_k6_d5_weight Enumerator:

$(ORBITER_PATH)orbiter.out-v.2:

-definition F:finite_field-q:2:-end:

-definition v:vector:format:6:field:F:
code_n15_k6_d6_a.we:

```
do

\end
```

# surprise: d = 6

```
do

\end
```

```
\begin{verbatim}
11399 11400 11401 11402 11403 11404 11405 11406 11407 11408 11409 11410 11411 11412 11413 11414 11415 11416 11417 11418 11419 11420 11421 11422 11423 11424 11425 11426 ...
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```
Section 9.9: Coding Theory -- Classification

# code-classification:
codes_8_4_4:
  $(ORBITER_PATH)orbiter.out-\v-6-
  -orbiter_path-$(ORBITER_PATH)-
  -define-G-
  -linear_group-PGL-2-end-
  -with-G-do-
  -group_theoretic_activity-
  -poset_classification_control-
  -problem_label-codes_8_4_4-
  -drawposet-
  -draw_options-embedded-radius-250-
  -line_width-1.0-spanning_tree-end-
  -report-end-
  -linear_codes-3-8-
  -end
codes_8_4_4_draw:
  $(ORBITER_PATH)orbiter.out-\v-3-
  -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
codes_8_4_4_poset:
  pdflatex-codes_8_4_4_poset.tex
open-codes_8_4_4_poset.pdf
dfdflatex codes_8_4_4_poset.tex
deropen-codes_8_4_4_poset.pdf
codes_8_4_4:
  $(ORBITER_PATH)orbiter.out-\v-
  -define-G-
  -linear_group-PGL-
  -with-G-do-
  -group_theoretic_activity-
  -poset_classification_control-
  -problem_label-codes_8_4_4-
  -drawposet-
  -draw_options-embedded-radius-250-
  -line_width-1.0-spanning_tree-end-
  -report-end-
  -linear_codes-3-8-
  -end
codes_8_4_4_draw:
  $(ORBITER_PATH)orbiter.out-\v-3-
  -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
codes_8_4_4_poset:
  pdflatex-codes_8_4_4_poset.tex
11521 ▷ ▷ -end
11522 ▷ ▷
11523
11524 #5/31/2020: 28-min-22-sec: on Mac
11525
11526 #0: 1-orbits
11527 #1: 1-orbits
11528 #2: 1-orbits
11529 #3: 1-orbits
11530 #4: 1-orbits
11531 #5: 1-orbits
11532 #6: 1-orbits
11533 #7: 1-orbits
11534 #8: 1-orbits
11535 #9: 2-orbits
11536 #10: 3-orbits
11537 #11: 4-orbits
11538 #12: 5-orbits
11539 #13: 5-orbits
11540 #14: 4-orbits
11541 #15: 3-orbits
11542 #16: 1-orbits
11543 #total: 36
11544
11545
codes_d4:
11547 ▷ $(ORBITER_PATH)orbiter.out- v.3\ 
11548 ▷ ▷ -define G -linear_group -PGL-4\2-end\ 
11549 ▷ ▷ -with G- do\ 
11550 ▷ ▷ -group_theoretic_activity\ 
11551 ▷ ▷ -poset_classification_control -W\ 
11552 ▷ ▷ -problem_label codes_r4_d4 - draw_poset\ 
11553 ▷ ▷ -embedded -end-linear codes_4:100\ 
11554 ▷ ▷ -end\ 
11555 ▷ ▷ -end
11556
11557
codes_24_12_8:
11559 ▷ $(ORBITER_PATH)orbiter.out- v.6\ 
11560 ▷ ▷ -orbiter_path $(ORBITER_PATH)\ 
11561 ▷ ▷ -define G\ 
11562 ▷ ▷ -linear_group -PGL-12\2-end\ 
11563 ▷ ▷ -with G- do\ 
11564 ▷ ▷ -group_theoretic_activity\ 
11565 ▷ ▷ -poset_classification_control\ 
11566 ▷ ▷ ▷ -problem_label codes_24_12_8\ 
11567 ▷ ▷ ▷ -draw_poset\ 

681
codes_24_12_8_draw:
$\$(ORBITER\_PATH)orbiter.out\,-v\,3\$

draw\_layered\_graph:

codes_24_12_8_poset_lvl_24.layered_graph:

radius-100-spanning_tree-embedded

line_width-0.5-x\_stretch-1.4

scale-0.25-nodes\_empty

definiG

with-G-do

group\_theoretic\_activity

poset\_classification\_control

problem\_label-glynn\_arc

draw\_options:-embedded:-radius-250

line_width-1.0-spanning\_tree-embedded

draw\_poset

report-\end

linear\_codes-6:10

end

glynn\_arc

codes_24_12_8_poset_lvl_24.layered_graph

codes_24_12_8_draw:

five\_points\_in\_general:
Chapter 10 - Combinatorics

Section 10.1: Combinatorics

Codes $q_{13}$

- Defining $G$
- Linear group $\text{PGL}(4,2)$
- With $G$ do group-theoretic activity
- Problem label: five points in general
- Draw options: embedded, radius 250
- Line width 1.0, spanning tree
- Report
- Draw poset
- Linear codes $6,12$

Report

---

Chapter 10 - Combinatorics

Section 10.1: Combinatorics

Codes $q_{13}$
SECTION_COMBINATORICS:

Char_Sym.4:

Char_Sym.5:

Char_Sym.6:

all_subsets.10.3:

rank_k_subsets.test:

Walsh_matrix.4:

Walsh_matrix.4-end

Dedekind.10.10:

Dedekind.30.2:

Dedekind.100.2:
elementary_symmetric_functions_4:
$(\text{ORBITER\_PATH})\text{orbiter.out} -\text{make\_elementary\_symmetric\_functions}\_4\_4$

elementary_symmetric_functions_8:
$(\text{ORBITER\_PATH})\text{orbiter.out} -\text{make\_elementary\_symmetric\_functions}\_8\_8$

# 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"
$(0, 5) (1, 8, 12, 9, 4, 11, 7) (2, 10, 6)$

LARGE_SET_S5="10, 11, 0, 7, 12, 2, 3, 1, 4, 5, 8, 6, 9"
file S:

echo -ROW,C0

$\{(LARGE_SET_S0)\}
$\{(LARGE_SET_S1)\}
$\{(LARGE_SET_S2)\}
$\{(LARGE_SET_S3)\}
$\{(LARGE_SET_S4)\}
$\{(LARGE_SET_S5)\}
$\{(LARGE_SET_S6)\}
$\{(LARGE_SET_S7)\}
$\{(LARGE_SET_S8)\}
$\{(LARGE_SET_S9)\}
$\{(LARGE_SET_S10)\}
$\{END\}

> S.csv

Large set H5:

$\{(ORBITER_PATH)orbiter.out\} -v -o 10

-define G-permutation_group -symmetric_group -13

-subgroup_by_generators H5,5,1:$(GENERATORS_H5)-end

-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:

$\{(ORBITER_PATH)orbiter.out\} -v -o 10

-define G-permutation_group -symmetric_group -13

-subgroup_by_generators C13,13,1:$(GENERATORS_C13)-end
### the following lines were created using -export_orbiter:  
Generators Perm13_Subgroup_C13_13 =  
"11,0,10,12,5,3,7,4,2,8,6,9,1"  
Perm13_Subgroup_C13_13:  
\$(ORBITER_PATH)orbiter.out -v 2\  
-define G permutation_group symmetric_group_13\  
-subgroup_by_generators Perm13_Subgroup_C13_13 13·1\  
\$(GENERATORS_Perm13_Subgroup_C13_13)\  
-end\  
Large set mult C13xS:  
\$(ORBITER_PATH)orbiter.out -v 10\  
-define G permutation_group symmetric_group_13 -end\  
-end\  
-multiply_elements_csv_column_major_ordering\  
C13_elts.csv.S.C13xS.csv\  
-end\  
Large set mult C13xSxH5:  
\$(ORBITER_PATH)orbiter.out -v 10\  
-define G permutation_group symmetric_group_13 -end\  
-end\  
-multiply_elements_csv_column_major_ordering\  
C13xS.H5_elts.csv.C13xSxH5.csv
Large set mult_C13xSxH5 apply:

```bash
$ (ORBITER_PATH)orbiter.out -v 4\n
define A -vector -dense "10,9,8,7,6,5,4,3,2,1" -end\n
define D -diophant\n
label part10\n
coefficient_matrix A\n
RHS "10,10,1"\n
x_min_global 0 -x_max_global 10\n
end\n
with D -do\n
diophant_activity -solve mckay\n
diophant -end\n```

# Finds 42 solutions with 67 backtrack steps

```bash
oc tic monomials:\n
$ (ORBITER_PATH)orbiter.out -v 4\n
define A -vector -dense "1,1,1,1" -end\n
define D -diophant\n
diophant -label octic monomials\n
coefficient_matrix A\n```
11894 \(-\text{RHS}\{}8,8,1\)\n11895 \(-x\text{\_min\_global}\text{\_0} - x\text{\_max\_global}\text{\_8}\)\n11896 \(-\text{end}\)\n11897 \(-\text{with\_D\_do}\)\n11898 \(-\text{diophant\_activity}\text{\_solve\_mckay}\)\n11899 \(-\text{end}\)\n11900 sort \(-r\text{\_octic\_monomials}\text{\_sol}\text{\_octic\_monomials\_sorted.txt}\)\n11901 #Found 165 solutions with 210 backtrack steps\n11902 #165=\text{binomial}(11,3)\n11903 test system:\n11904 test system:
11906 test system:
11907 \$(\text{ORBITER\_PATH})\text{orbiter.out\_v\_4}\)
11908 \(-\text{define\_A\_vector\_format\_7\_dense}\$(\text{TEST\_SYSTEM})\text{\_end}\)
11909 \(-\text{define\_D\_diophant}\)
11911 \(-\text{diophant\_activity}\text{\_solve\_mckay}\)
11914 \(-\text{end}\)
11916 McKay test:
11917 McKay test:
11918 \$(\text{ORBITER\_PATH})\text{orbiter.out\_v\_4}\)
11919 \(-\text{define\_A\_vector\_format\_7\_dense}\$(\text{TEST\_SYSTEM})\text{\_end}\)
11921 \(-\text{define\_D\_diophant}\)
11922 \(-\text{diophant\_activity}\text{\_solve\_mckay}\)
11928 \(-\text{end}\)
11930 DLX test:
11931 \$(\text{ORBITER\_PATH})\text{orbiter.out\_v\_4}\)
11932 \(-\text{define\_A\_vector\_format\_7\_dense}\$(\text{TEST\_SYSTEM})\text{\_end}\)
11933 \(-\text{define\_D}\)
11934 \(-\text{diophant\_label\_test\_system}\)
11935 \(-\text{diophant\_activity}\text{\_solve\_mckay}\)
11940 \(-\text{diophant\_activity}\text{\_solve\_DLX}\)
Section 10.3: Combinatorial Linear Spaces

SECTION COMBINATORIAL LINEAR SPACES:

linsp6:

define A -vector -format 1 -dense "15,10,6,3,1" -end

define D -diophant -label linsp6

coefficient_matrix A

RHS "15,15,1"

x_min_global 0

x_max_global 15

end

with D -do

diophant_activity -solve mckay

end

# Found 15 solutions with 22 backtrack steps

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
11988  linsp30_pt_types:
11989  $(ORBITER_PATH)orbiter.out.-v.4\  
11990  -define A -vector -format.1 -dense."6,4,3" -end\  
11991  -define D -diophant\  
11992  -label linsp30_pt_types\  
11993  -coefficient_matrix A\  
11994  -RHS."29,29,1" -x.bounds."0,1,0,27,0,24"\  
11995  -end\  
11996  -with D -do\  
11997  -diophant_activity -solve_mckay\  
11998  -end\  
11999  linsp30_pt_distribution:
12000  $(ORBITER_PATH)orbiter.out.-v.4\  
12001  -define A -vector -format.6 -dense\  
12002  "1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21"\  
12003  -end\  
12004  -define D -diophant\  
12005  -label linsp30_pt_distribution\  
12006  -coefficient_matrix A\  
12007  -RHS."30,30,1,7,7,1,135,135,1,96,96,1,0,351,2,0,276,2"\  
12008  -x_min_global.0 -x_max_global.30\  
12009  -end\  
12010  -with D -do\  
12011  -diophant_activity -solve_mckay\  
12012  -end\  
12013  -with D -do\  
12014  -diophant_activity -draw_as_bitmap 20.8\  
12015  -end\  
12016  
12017  
12018  
12019  
12020  
12021  
12022  
12023  
12024  
12025  
12026  
12027  
12028  
12029  
12030  
12031  
12032  
12033  
12034  

691
geo_paszch:
$\texttt{(ORBITER\_PATH)orbiter.out-v.8}\$
define Test_lines -set -loop 1:7:1 -end
define Geo -geometry_builder
$\texttt{-V.6-B.4-TDO.2-fuse.1}\$
fname GEO-paszch
-test Test_lines
-end

geo_petersen:
$\texttt{(ORBITER\_PATH)orbiter.out-v.8}\$
define Test_lines -set -loop 3:11:1 -end
define Geo -geometry_builder
$\texttt{-V.10-B.15-TDO.3-fuse.1}\$
fname GEO-petersen -girth 5
-search tree
-test Test_lines
-end

geo_7_3:
$\texttt{(ORBITER\_PATH)orbiter.out-v.2}\$
define Test_lines -set -loop 3:8:1 -end
define Geo -geometry_builder
$\texttt{-V.7-B.7-TDO.3}\$
-fuse 1 -fname GEO_7_3
-test Test_lines
-end

geo_7_3_no_square_test:
$\texttt{(ORBITER\_PATH)orbiter.out-v.2}\$
define Test_lines -set -loop 3:8:1 -end
define Geo -geometry_builder
$\texttt{-V.7-B.7-TDO.3}\$
fuse 1 -fname GEO_7_3
-test Test_lines
-no_square_test
-end

geo_7_3_no_square_test_draw:
$\texttt{(ORBITER\_PATH)orbiter.out-v.10}\$
draw incidence_structure_description
-width 60 -with 10:6 -end
define C -combinatorial_objects
-file_of incidence geometries 7_3_nst.inc 7_7:21
-end
-with C -do
geo 7.3 orderly:

geo 7.3 orderly draw:

geo 7.3 orderly mem debug:

geo 8.3:
define-Geo-geometry_builder:\
V.8.-B.8.-TDO.3:\
fuse.1.-fname_GEO.8.3:\
test-Test_lines:\
end

#-print_at_line.4
#1:geo:0.11.18.29.30.38.44.54.
#ago=48

g9.3:
$\{\text{ORBITER\_PATH}\}\text{orbiter.out}-v.2\$
\text{-define-Test_lines}-set-loop.3:10:1:-end\$
\text{-define-Geo-geometry_builder}\$
\text{-V.9.-B.9.-TDO.3}\$
\text{-fuse.1.-fname_GEO.9.3}\$
\text{-test-Test_lines}\$
\text{-end}
g10.3:
$\{\text{ORBITER\_PATH}\}\text{orbiter.out}-v.20\$
\text{-define-Test_lines}-set-loop.4:11:1:-end\$
\text{-define-Geo-geometry_builder}\$
\text{-V.10.-B.10.-TDO.3.-fuse.1}\$
\text{-fname_GEO.10.3}\$
\text{-test-Test_lines}\$
\text{-end}

g10.3-orderly:
$\{\text{ORBITER\_PATH}\}\text{orbiter.out}-v.20\$
\text{-define-Test_lines}-set-loop.4:11:1:-end\$
\text{-define-Geo-geometry_builder}\$
\text{-V.10.-B.10.-TDO.3.-fuse.1}\$
\text{-fname_GEO.10.3}\$
\text{-test-Test_lines}\$
\text{-orderly}\$

#10:geos
#8/26/2021:0:sec.on:Mac

g10.3-orderly:
geo_10_3.orderly_mem_debug:
$\$(\text{ORBITER\_PATH})\text{orbiter.out-v.2}\$

-define Test_lines-set-loop 4 11 1-end
-define Geo-geometry_builder-
-V.10-B.10-TDO.3-fuse 1-
-fname GEO-10.3-
test Test_lines-
-orderly-
-end

geo_10_3.tree:
$\$(\text{ORBITER\_PATH})\text{orbiter.out-v.20}\$

-define Test_lines-set-loop 0 11 1-end
-define GEO-geometry_builder-
-V.10-B.10-TDO.3-fuse 1-
-fname GEO-10.3-
-search_tree-
test Test_lines-
-end
$\$(\text{ORBITER\_PATH})\text{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-
-tree_draw-
-restrict 2-
-file 10_3_tree.txt-
-select_path "0,0,15,26,46,56,72,80,93,106,119"-
-end
dpdflatex 10_3_tree_draw.tex
don\textit{10_3_tree_draw.pdf}

-nodes_empty-
-sideways-

Desargues_path_lex_least:
\texttt{\$\$(\text{DESARGUES\_PATH\_LEX\_LEAST})\text{\textgreater}\text{Desargues_path_lex_least.inc}}
\texttt{\$\$(\text{ORBITER\_PATH})\text{orbiter.out-v.10}\$

-draw incidence structure description-
-width 60 -with 10.6 -end \end{enumerate}

\end{document}
geo_12_3:

\$\text{(ORBITER\_PATH)\textcolor{red}{\textbackslash orbiter.out-\textbackslash v.2}}\$

-define Test_lines -set -loop 4\_13\_1 -end

-define Geo -geometry_builder

-V\_12 -B\_12 -TDO\_3

-fuse\_1 -fname GEO\_12\_3

test Test_lines

-end

geo_12_3 orderly:

\$\text{(ORBITER\_PATH)\textcolor{red}{\textbackslash orbiter.out-\textbackslash v.2}}\$

-define Test_lines -set -loop 4\_13\_1 -end

-define Geo -geometry_builder

-V\_12 -B\_12 -TDO\_3

-fuse\_1 -fname GEO\_12\_3

test Test_lines

-f\_orderly

-end

geo_13_3:

\$\text{(ORBITER\_PATH)\textcolor{red}{\textbackslash orbiter.out-\textbackslash 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

-end

geo_13_3 orderly:

\$\text{(ORBITER\_PATH)\textcolor{red}{\textbackslash orbiter.out-\textbackslash 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
```
12312  ▷  ▷  ▷  -test-Test_lines\n12313  ▷  ▷  ▷  -f_orderly\n12314  ▷  ▷  -end
12315
12316
12317
12318  geo_14_3:
12319  ▷  $(ORBITER_PATH)orbiter.out-v.2\n12320  ▷  ▷  -define-Test_lines-set-loop-4-15-1-end\n12321  ▷  ▷  -define-Geo-geometry_builder\n12322  ▷  ▷  ▷  -V-14-B-14-TDO-3\n12323  ▷  ▷  ▷  -fuse-1-fname_GEO-14_3\n12324  ▷  ▷  ▷  -test-Test_lines\n12325  ▷  ▷  -end
12326
12327  #21399-geos,-56448,-128,-24^2,-16^3,-14^3,-12^7-8^15,-7,6^12,4^91,-1^20328
12328  #User-time:0:55
12329  #nb_calls_to_densenauty=2089344
12330
12331
12332  geo_14_3_orderly:
12333  ▷  $(ORBITER_PATH)orbiter.out-v.2\n12334  ▷  ▷  -define-Test_lines-set-loop-4-15-1-end\n12335  ▷  ▷  -define-Geo-geometry_builder\n12336  ▷  ▷  ▷  -V-14-B-14-TDO-3\n12337  ▷  ▷  ▷  -fuse-1-fname_GEO-14_3\n12338  ▷  ▷  ▷  -test-Test_lines\n12339  ▷  ▷  ▷  -f_orderly\n12340  ▷  ▷  ▷  -end
12341
12342  #User-time:0:50
12343
12344
12345  15_3.inc:
12346  ▷  $(ORBITER_PATH)orbiter.out-v.2\n12347  ▷  ▷  -define-Test_lines-set-loop-4-16-1-end\n12348  ▷  ▷  -define-Geo-geometry_builder\n12349  ▷  ▷  ▷  -V-15-B-15-TDO-3\n12350  ▷  ▷  ▷  -fuse-1-fname_GEO-15_3\n12351  ▷  ▷  ▷  -test-Test_lines\n12352  ▷  ▷  ▷  -end
12353
12354  #245342-geos,-8064,-720,-192^2,-128,-72,-48^6,-32,-30^2,-24^2,-20^2,-18,-16^10,-
12355  15^2,12^11,10^3,-8^34,-6^59,-5^5,-4^180,-3^69,2^94,-2^3709,-1^241240
12356  #-8-min-11-sec-on-Mac
12357  #running-out-of-memory
```
The unique Cremona-Richmond configuration with group of order 720

User time: 0 of a second, dt=0 tps=100

nb_calls_to_densenauty=23

1

geo_17_3_g4.orderly:

1 memory_debug 2

-define Test_lines -set -loop 4 18 1 -end

-define Geo -geometry_builder -V 17 -B 17 -TDO 3

-fuse 1 -fname GEO 17 3 g4

girth 4

-test Test_lines

-ordelay

-end

#1.sol

geo_18_3_g4:

$ (ORBITER_PATH) orbiter.out -v 2

$ (ORBITER_PATH) orbiter.out -v 2

-memory debug 2

-define Test_lines -set -loop 4 19 1 -end

-define Geo -geometry_builder -V 17 -B 17 -TDO 3

-fuse 1 -fname GEO 17 3 g4

girth 4

-test Test_lines

-end

#1.sol

geo_19_3_g4:
12404 ▶ ▶ ▶ -V.18-B.18-TDO.3\n12405 ▶ ▶ ▶ -fuse.1-fname_GEO.18.3_g4\n12406 ▶ ▶ ▶ -girth.4\n12407 ▶ ▶ ▶ -search_tree\n12408 ▶ ▶ ▶ -test(Test_lines)\n12409 ▶ ▶ -end
12410
12411 #.sol,1·sec
12412
12413 geo.19.3_g4:
12414 ▶ $(ORBITER_PATH)orbiter.out-v.2\n12415 ▶ -define(Test_lines-set-loop.4.20.1-end)\n12416 ▶ -define-Geo-geometry_builder\n12417 ▶ ▶ -V.19-B.19-TDO.3\n12418 ▶ ▶ -fuse.1-fname_GEO.19.3_g4\n12419 ▶ ▶ -girth.4\n12420 ▶ ▶ -test(Test_lines)\n12421 ▶ ▶ -end
12422
12423 #.14.sol,10·sec-on Mac
12424
12425 geo.20.3_g4:
12426 ▶ $(ORBITER_PATH)orbiter.out-v.2\n12427 ▶ -define(Test_lines-set-loop.4.21.1-end)\n12428 ▶ -define-Geo-geometry_builder\n12429 ▶ ▶ -V.20-B.20-TDO.3\n12430 ▶ ▶ -fuse.1-fname_GEO.20.3_g4\n12431 ▶ ▶ -girth.4\n12432 ▶ ▶ -test(Test_lines)\n12433 ▶ ▶ -end
12434
12435 #.162.sol,User-time:2:5·on Mac
12436
12437 geo.21.3_g4:
12438 ▶ $(ORBITER_PATH)orbiter.out-v.2\n12439 ▶ -define(Test_lines-set-loop.4.22.1-end)\n12440 ▶ -define-Geo-geometry_builder\n12441 ▶ ▶ -V.21-B.21-TDO.3\n12442 ▶ ▶ -fuse.1-fname_GEO.21.3_g4\n12443 ▶ ▶ -girth.4\n12444 ▶ ▶ -test(Test_lines)\n12445 ▶ ▶ -end
12446
12447
12448 geo.15.4:
12449 ▶ $(ORBITER_PATH)orbiter.out-v.2\n12450
12451 ▶ ▶ -define Test_lines: set: -loop 4: 16: 1: -end\n12452 ▶ ▶ -define Geo: geometry_builder\n12453 ▶ ▶ V 15: -B 15: -TDO 4\n12454 ▶ ▶ fuse 1: -fname GEO 15 4\n12455 ▶ ▶ search_tree\n12456 ▶ ▶ test Test_lines\n12457 ▶ ▶ -end\n12458 ▶ $(ORBITER_PATH) orbiter.out -v 2\n12459 ▶ ▶ draw_options: -embedded -radius 50\n12460 ▶ ▶ ▶ nodes_empty\n12461 ▶ ▶ ▶ scale 0.5: -line_width 0.3: -end\n12462 ▶ ▶ ▶ tree_draw 15 4 tree.txt\n12463 ▶ pdflatex 15 4 tree draw tex\n12464 ▶ open 15 4 tree draw pdf\n12465
12466 # 4 objects\n12467 # ago=360, 30, 24, 15\n12468 # User time: 0.15 of a second, dt=15 tps=1.00\n12469 # nb calls to densenauty=6767\n12470
12471
12472 12473 geo 16 4 g4:\n12474 ▶ $(ORBITER_PATH) orbiter.out -v 2\n12475 ▶ ▶ -define Test_lines: set: -loop 4: 17: 1: -end\n12476 ▶ ▶ -define Geo: geometry_builder\n12477 ▶ ▶ ▶ V 16: -B 16: -TDO 4\n12478 ▶ ▶ ▶ fuse 1: -fname GEO 16 4 g4\n12479 ▶ ▶ ▶ girth 4\n12480 ▶ ▶ ▶ test Test_lines\n12481 ▶ ▶ ▶ -end\n12482
12483 # none\n12484
12485 12486 40 4 g4 inc:\n12487 ▶ $(ORBITER_PATH) orbiter.out -v 2\n12488 ▶ ▶ -define Test_lines: set: -loop 0: 41: 1: -end\n12489 ▶ ▶ -define Geo: geometry_builder\n12490 ▶ ▶ ▶ V 40: -B 40: -TDO 4\n12491 ▶ ▶ ▶ fuse 1: -fname GEO 40 4 g4\n12492 ▶ ▶ ▶ girth 4\n12493 ▶ ▶ ▶ search_tree\n12494 ▶ ▶ ▶ test Test_lines\n12495 ▶ ▶ ▶ -end\n12496 ▶ $(ORBITER_PATH) orbiter.out -v 2\n12497 ▶ ▶ -draw_options: -embedded -radius 50\n701
12498  ▶  ▶  ▶  -xin:10000.-yin:10000\,
12499  ▶  ▶  ▶  -xout:100000.-yout:100000\,
12500  ▶  ▶  ▶  -nodes_empty\,
12501  ▶  ▶  ▶  -scale:0.5.-line_width:0.3.-end\,
12502  ▶  ▶  ▶  -tree_draw-40_4_g4_tree.txt
12503  ▶  ▶  ▶  pdflatex-40_4_g4_tree_draw.tex
12504  ▶  ▶  ▶  open-40_4_g4_tree_draw.pdf
12505
12506
12507  #=2:geos.-ago=51840^2
12508  #User-time:.0.18-of-a-second,-dt=18-tps=.100
12509  #nb_calls_to_densenauty=1065
12510
12511
12512  geo_63_3_g6:
12513  ▶  $(ORBITER_PATH)orbiter.out-v.2\,
12514  ▶  ▶  -define-Test_lines-set-loop-4.64:1.-end\,
12515  ▶  ▶  -define-Geo-geometry_builder\,
12516  ▶  ▶  ▶  -V-63.-B-63.-TD0:3\,
12517  ▶  ▶  ▶  -fuse:1.-fname_GEO-63_3_g6\,
12518  ▶  ▶  ▶  -girth:6\,
12519  ▶  ▶  ▶  -test-Test_lines\,
12520  ▶  ▶  ▶  -end
12521
12522
12523
12524  geo_LSQ6:
12525  ▶  $(ORBITER_PATH)orbiter.out-v.2\,
12526  ▶  ▶  -define-Test_lines-set-loop-1.19:1.-end\,
12527  ▶  ▶  -define-Geo-geometry_builder\,
12528  ▶  ▶  ▶  -V-6,6,6.-B-1,1,1,36.-TD0\,
12529  ▶  ▶  ▶  "1,0,0,6,-0,1,0,6,0,0,1,6"\,
12530  ▶  ▶  ▶  -fuse:3.-fname_GEO-LSQ6\,
12531  ▶  ▶  ▶  -test-Test_lines\,
12532  ▶  ▶  ▶  -end
12533
12534
12535
12536
12537  ####################################################################################################
12538  #Section-10.5:-Design-Theory
12539
12540  SECTION DESIGN THEORY:
12541
12542
12543  design_PG_2_3:
12544  ▶  $(ORBITER_PATH)orbiter.out-v.8\,
design_PG_2.4:
$(ORBITER_PATH)orbiter.out -v.8:\
$-create_design-q.4-family-PG_2.q-end

design_PG_2.3_table_create:
$(ORBITER_PATH)orbiter.out -v.2:\
$-define-D-design-q.3-family-PG_2.q-end\n$-define-Sym13-permutation_group-symmetric_group-13-end\n$-define-T-design_table-D."PG_2.13"-Sym13-end.

# written file PG_2.13_design_table.csv
# 1108800 designs
# User time: 7:30

design_PG_2.3_group_5:
$(ORBITER_PATH)orbiter.out -v.2:\
$-define-D-design-q.3-family-PG_2.q-end\n$-define-T-design_table-D."PG_2.13"-Sym13-end\n$-define-LSW-large_set_with_symmetry_assumption-T.\n$-H."5"$(GENERATORS_H5)\n$-N."1200"$(GENERATORS_N5)\n$-prefix:"H5"\n$-selected_orbit_length-5\n$-end\n$-with-LSW-do\n$-large_set_with_symmetry_assumption_activity\n$-normalizer_on_orbits_of_a_given_length-5\n$-end
design_PG_2.3_group_5_sol_0:
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$
$\$(ORBITER\_PATH)\$orbiter.out\$-v.2:\$


```
12639  \>
12640  \>
12641  \>
12642
12643
12644
12645 #to-create-simple-7-designs-on-33-points-with-block-size-8-and-lambda=10-invariant-under-PGGL(2,32):
12646
12647  KM_PGGL_2_32:
12648  \>
12649  \>
12650  \>
12651  \>
12652  \>
12653  \>
12654  \>
12655  \>
12656  \>
12657  \>
12658  \>
12659  \>
12660  \>
12661  \>
12662  \>
12663  \>
12664  \>
12665  \>
12666  \>
12667  \>
12668  \>
12669  \>
12670  \>
12671  \>
12672  \>
12673  \>
12674  \>
12675  \>
12676  \>
12677  \>
12678  \>
12679  \>
12680  \>
12681
12682
12683  KM_PSL_3_5:
12684  \>
```

705
design large set rank k subsets a:
   $(ORBITER\ PATH)orbiter.out\ -v\ -2\ -rank\ k\ subset\ 13\ 4\ $(PLANE\ 1)

design large set rank k subsets b:
   $(ORBITER\ PATH)orbiter.out\ -v\ -2\ -rank\ k\ subset\ 13\ 4\ $(PLANE\ 2)

design large set rank k subsets c:
   $(ORBITER\ PATH)orbiter.out\ -v\ -2\ -rank\ k\ subset\ 13\ 4\ $(PLANE\ 3)

design large set rank k subsets d:
   $(ORBITER\ PATH)orbiter.out\ -v\ -2\ -rank\ k\ subset\ 13\ 4\ $(PLANE\ 4)

# Section 10.6: Design Theory - Large Sets

SECTION DESIGN THEORY LARGE SETS:

AG_2\_3.inc:
   $(ORBITER\ PATH)orbiter.out\ -v\ 2\ 
   -define Geo-geometry_builder 
   -V-9-B-12\ 
   -TDO-4-fuse-1\ 
   -fname GEO AG_2\_3\ 
   -test 3,4,5,6,7,8,9\ 
   -end

# 9-12-3
-1-1
# 432

LS_AG_2_3_design_table_create:

adox $(ORBITER_PATH)orbiter.out -v 20

define D design list of blocks

9 3 $(AG_2_3_BLOCKS) end

define Sym 9 permutation group symmetric group 9 end

define T design table D "AG_2_3" Sym9

# creates AG_2_3_design_table.csv

# and AG_2_3.makefile

# 0, 0, 13, 22, 27, 35, 41, 47, 53, 55, 59, 71, 76

# 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:

adox $(ORBITER_PATH)orbiter.out -v 20

define gens vector file AG_2_3 gens csv end

define G permutation group

bsgs AG_2_3 "AG_2_3" 840 362880 "0,1,2,3,4,5,6,7" 8 gens end

with G do

- group theoretic activity

- orbits on points

- stabilizer of orbit rep 0

- end

# Written file AG_2_3_stab_orb_0.makefile of size 239

# the stabilizer of the first design:

AG_2_3_stab_orb_0:

adox $(ORBITER_PATH)orbiter.out -v 20

define gens vector file AG_2_3_stab_orb_0 gens csv end

define G permutation group

bsgs AG_2_3 "AG_2_3" stab orb 0 "AG_2_3_stab_orb_0"

840 432 "0,1,2,3,4,5,6,7,8" 5 gens
12817 \end{verbatim}
12818 \end{verbatim}
12819 \end{verbatim}
12820 \end{verbatim}
12821 \end{verbatim}
12822 \end{verbatim}
12823 \end{verbatim}
12824 \end{verbatim}
12825 \end{verbatim}
12826 \begin{verbatim}
12827 AG_2_3_stab_orb_0_Perm840_res192:
12828 $(ORBITER_PATH)orbiter.out-v.2-
12829 \end{verbatim}
12830 \end{verbatim}
12831 \end{verbatim}
12832 \end{verbatim}
12833 \end{verbatim}
12834 \end{verbatim}
12835 \end{verbatim}
12836 \end{verbatim}
12837 \end{verbatim}
12838 \end{verbatim}
12839 \end{verbatim}
12840 \end{verbatim}
12841 \end{verbatim}
12842 \begin{verbatim}
12843 LS_AG_2_3_disjoint_sets_graph_and_cliques:
12844 $(ORBITER_PATH)orbiter.out-v.2-
12845 \end{verbatim}
12846 \end{verbatim}
12847 \end{verbatim}
12848 \end{verbatim}
12849 \end{verbatim}
12850 \end{verbatim}
12851 \end{verbatim}
12852 \end{verbatim}
12853 \end{verbatim}
12854 \end{verbatim}
12855 \end{verbatim}
12856 \end{verbatim}
12857 \end{verbatim}
12858 \end{verbatim}
12859 \end{verbatim}
12860 \end{verbatim}
12861 \end{verbatim}
12862 \end{verbatim}
12863 \end{verbatim}
LS_AG_2_3.disjoint_sets_split:
$(ORBITER_PATH) orbiter.out -v 4 \
> > -define-Gamma -graph -load  
> > AG_2_3_design_table_disjoint_sets.colored_graph  
> > -end  
> > -with-Gamma -do  
> > graph_theoretic_activity  
> > -split_by_clique -0 -0  
> > -end  

#AG_2_3_design_table_disjoint_sets_0.graph

#AG_2_3_design_table_disjoint_sets_0_subset.txt

LS_AG_2_3_export_solutions:
$(ORBITER_PATH) orbiter.out -v 20  
> > -define-D -design -list_of_blocks 9 3  
> > $ (AG_2_3_BLOCKS) -end  
> > -define-Sym9 -permutation_group -symmetric_group 9 -end  
> > -define-T -design_table -D: AG_2_3 -Sym9  
> > -with-D -do  
> > -design_activity  
> > -extract_solutions_by_index -AG_2_3 -Sym9  
> > AG_2_3_design_table_disjoint_sets_sol.csv  
> > solutions.csv  
> > "  
> > -end

#User.time: 0.39 of a second, dt=39 tps=100

#solutions.csv

SECTION DESIGN THEORY DELANDTSHOEER DOYEN:

DD_PP4:
12911▶ $(ORBITER\_PATH)\$\text{orbiter.out}\$-v.6\$
12912▶ ▶ -\text{DelandtsheerDoyen}\$(PP4)\$(PP4\_GROUP1)$(PP4\_MASK1)\$
12913▶ ▶ ▶ -\text{end}\$
12914
12915\text{DD\_PP4\_system:}
12916▶ $(ORBITER\_PATH)\$\text{orbiter.out}\$-v.4\$
12917▶ ▶ -\text{define}\text{-diophant}-\text{label}\text{-PP4}\$
12918▶ ▶ ▶ -\text{problem}\text{-of}\text{Steiner}\text{-type}\text{-10}\text{-PP4}\text{_pair}\text{-covering}.\text{csv}\$
12919▶ ▶ ▶ ▶ -\text{has}\text{-sum}\text{-1}\$
12920▶ ▶ ▶ ▶ -\text{end}\$
12921▶ ▶ ▶ ▶ -\text{with}\text{-D}\text{-do}\$
12922▶ ▶ ▶ ▶ ▶ -\text{diophant}\text{-activity}\text{-solve}\text{mckay}\$
12923▶ ▶ ▶ ▶ -\text{end}\$
12924
12925
12926
12927
12928
12929\text{DD\_CC:}
12930▶ $(ORBITER\_PATH)\$\text{orbiter.out}\$-v.6\$
12931▶ ▶ -\text{DelandtsheerDoyen}\text{-search}\text{wrt}\text{subgroup}\$
12932▶ ▶ ▶ -$(\text{DELANDTSHEER}\text{-DOYEN}\text{-PROBLEM}\text{-COLBOURN}\text{-COLBOURN}\text{-7}\text{-13})\$
12933▶ ▶ ▶ ▶ -$(\text{DELANDTSHEER}\text{-DOYEN}\text{-PROBLEM}\text{-COLBOURN}\text{-COLBOURN}\text{-7}\text{-13}\text{-GROUP1})\$
12934▶ ▶ ▶ ▶ ▶ -$(\text{DELANDTSHEER}\text{-DOYEN}\text{-PROBLEM}\text{-COLBOURN}\text{-COLBOURN}\text{-7}\text{-13}\text{-MASK1})\$
12935▶ ▶ ▶ ▶ ▶ -\text{end}\$
12936
12937#\text{target}\text{-level:}\text{-6}
12938#k2:15
12939#number\text{-of}\text{-k-orbits}\text{-at}\text{-target}\text{-level:}:1774964
12940
12941#\text{creates}\text{-DD\_CC\_7\_13}\text{-pair}\text{-covering}.\text{csv}
12942
12943
12944\text{DD\_CC\_system:}
12945▶ $(ORBITER\_PATH)\$\text{orbiter.out}\$-v.4\$
12946▶ ▶ -\text{define}\text{-diophant}-\text{label}\text{-DD\_CC\_7\_13}\$
12947▶ ▶ -\text{problem}\text{-of}\text{Steiner}\text{-type}\text{-45}\text{-DD\_CC\_7\_13}\text{-pair}\text{-covering}.\text{csv}\$
12948▶ ▶ -\text{has}\text{-sum}\text{-3}\$
12949▶ ▶ -\text{end}\$
12950▶ ▶ -\text{with}\text{-D}\text{-do}\$
12951▶ ▶ ▶ -\text{diophant}\text{-activity}\text{-solve}\text{mckay}\$
12952▶ ▶ ▶ -\text{end}\$
12953▶
12954
12955
12956#\text{no}\text{-solution}
12957
```
12958
12959
12960 # 18603.:27.*.53.*.13
12961
12962 DD_M1_G1:
12963 \# $(ORBITER_PATH) orbiter.out -v.4 \n12964 \# -Delandtsheer_Doyen \n12965 \# $(DELANDTSHEER_DOYEN_PROBLEM_27_53) \n12966 \# $(DELANDTSHEER_DOYEN_PROBLEM_27_53_GROUP1) \n12967 \# $(DELANDTSHEER_DOYEN_PROBLEM_27_53_MASK1) \n12968 \# -end \n12969
12970 DD_M1_G1_S:
12971 \# $(ORBITER_PATH) orbiter.out -v.4 \n12972 \# -Delandtsheer_Doyen \n12973 \# $(DELANDTSHEER_DOYEN_PROBLEM_27_53) \n12974 \# $(DELANDTSHEER_DOYEN_PROBLEM_27_53_GROUP1) \n12975 \# $(DELANDTSHEER_DOYEN_PROBLEM_27_53_MASK1) \n12976 \# -singleton \n12977 \# -end \n12978
12979
12980 DD_PG_2_4_M1_G1:
12981 \# $(ORBITER_PATH) orbiter.out -v.4 \n12982 \# -Delandtsheer_Doyen \n12983 \# $(DELANDTSHEER_DOYEN_PROBLEM_3_7) \n12984 \# $(DELANDTSHEER_DOYEN_PROBLEM_3_7_GROUP1) \n12985 \# $(DELANDTSHEER_DOYEN_PROBLEM_3_7_MASK1) \n12986 \# -end \n12987
12988 PG_2_27_special:
12989 \# $(ORBITER_PATH) orbiter.out -v.2 \n12990 \# -define F -finite_field -q 27 -override_polynomial 46 -end \n12991 \# -define P -projective_space 2 F -end \n12992 \# -with P -do -projective_space_activity \n12993 \# -cheat_sheet \n12994 \# -end \n12995 \# pdflatex PG_2_27.tex \n12996 \# open PG_2_27.pdf
12997
12998
12999
13000
13001
13002
13003
13004
```
SECTION_TACTICAL_DECOMPOSITIONS:

max arc 16 4_start:
$(ORBITER_PATH)orbiter.out-v.4-maximal_arc_parameters.16.4

max arc 16 4_convert_stack.tdo:
$(ORBITER_PATH)orbiter.out-v.4-convert_stack_to_tdo-max_arc_q16_r4.stack

max arc 16 4_refine:
$(ORBITER_PATH)orbiter.out-v.4-tdo_refinement-
\input_file_max_arc_q16_r4.tdo-dual_is_linear_space-end

max arc 16 4r_print:
$(ORBITER_PATH)orbiter.out-v.4-tdo_print-max_arc_q16_r4r.tdo.

SECTION_SPREADS:

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

desarguesian_spread_in_PG_5_2:
$(ORBITER_PATH)orbiter.out-v.3-
\define_FQ-finite_field-q.8-end-
\define_Fq-finite_field-q.2-end-
\with_FQ-and_Fq-do-finite_field_activity-

713
desarguesian spread in PG $3\times 4$:
\begin{itemize}
  \item $(\text{ORBITER\_PATH})$orbiter.out \(-v.3\)
  \item \texttt{-define FQ\textunderscore finite\_field\textunderscore q\textunderscore 16\textunderscore end}
  \item \texttt{-define F\textunderscore q\textunderscore 16\textunderscore end}
  \item \texttt{-with FQ\textunderscore and F\textunderscore do\textunderscore finite\_field\_activity}
  \item \texttt{-cheat\textunderscore sheet\textunderscore desarguesian\textunderscore spread\textunderscore 2\textunderscore end}
\end{itemize}

\begin{itemize}
  \item \texttt{pdflatex\textunderscore Desarguesian\textunderscore Spread\textunderscore 3\textunderscore 4.tex}
  \item \texttt{open\textunderscore Desarguesian\textunderscore Spread\textunderscore 3\textunderscore 4.pdf}
\end{itemize}

\begin{itemize}
  \item \texttt{desarguesian\textunderscore spread\textunderscore in\textunderscore PG\textunderscore 3\textunderscore 5:}
  \item $(\text{ORBITER\_PATH})$orbiter.out \(-v.3\)
  \item \texttt{-define FQ\textunderscore finite\_field\textunderscore q\textunderscore 25\textunderscore end}
  \item \texttt{-define F\textunderscore q\textunderscore 25\textunderscore end}
  \item \texttt{-with FQ\textunderscore and F\textunderscore do\textunderscore finite\_field\_activity}
  \item \texttt{-cheat\textunderscore sheet\textunderscore desarguesian\textunderscore spread\textunderscore 2\textunderscore end}
\end{itemize}

\begin{itemize}
  \item \texttt{pdflatex\textunderscore Desarguesian\textunderscore Spread\textunderscore 3\textunderscore 5.tex}
  \item \texttt{open\textunderscore Desarguesian\textunderscore Spread\textunderscore 3\textunderscore 5.pdf}
\end{itemize}

\begin{itemize}
  \item \texttt{spreads4:}
  \item $(\text{ORBITER\_PATH})$orbiter.out \(-v.3\)
  \item \texttt{-define F\textunderscore q\textunderscore 2\textunderscore end}
  \item \texttt{-define P\textunderscore projective\_space\textunderscore 3\textunderscore F\textunderscore end}
  \item \texttt{-with P\textunderscore do}
  \item \texttt{-projective\_space\_activity}
  \item \texttt{-spread\textunderscore classify\textunderscore 2}
  \item \texttt{-problem\textunderscore label\textunderscore spreads\textunderscore 2\textunderscore 2\textunderscore depth\textunderscore 5}
  \item \texttt{-draw\textunderscore poset}
  \item \texttt{-end}
\end{itemize}

\begin{itemize}
  \item \texttt{spreads16\textunderscore 4:}
  \item $(\text{ORBITER\_PATH})$orbiter.out \(-v.6\)
  \item $\texttt{-orbiter\_path\textunderscore (\text{ORBITER\_PATH})}$
  \item \texttt{-define F\textunderscore q\textunderscore 4\textunderscore end}
  \item \texttt{-define P\textunderscore projective\_space\textunderscore 3\textunderscore F\textunderscore end}
  \item \texttt{-with P\textunderscore do}
  \item \texttt{-projective\_space\_activity}
  \item \texttt{-spread\textunderscore classify\textunderscore 2}
  \item \texttt{-problem\textunderscore label\textunderscore spreads\textunderscore 4\textunderscore 2}
  \item \texttt{-W\textunderscore depth\textunderscore 17\textunderscore -draw\textunderscore poset}
\end{itemize}
#17

          aaaaaaaaaaaaaaaaaaagaaaaaaaaadaaaaaQaaaaaaafaaaaaaaaaba
          aaaaaaaaaaaaaaaaaaaaaaaaaadamejgpaacaajlfmdabfncjmbneaa
          haaaaaaaaaaaaaaaaaaaaaaaaaaaaafaaaaaaaaaaafaaaaaaab
          aaaaaaaaaaaaaaaaaaaaaaaaaadamejgpaacaajlfmdabfncjmbneaa
          aaaaaaaaaaaaaaaaaaaaaaaaaaaadamejgpaacaajlfmdabfncjmbneaa
          aaaaaaaaaaaaaaaaaaaaaaaaaaaadamejgpaacaajlfmdabfncjmbneaa

#-1-3-1126-in-0-of-a-second, dt=0-tps=-100
#(81600,-1200,-576)-average-is-27792+0/:-3

#17-::-3-orbits
#total::-1129
#(81600,-1200,-576)-average-is-27792+0/:-3

#Section-11.2:-Translation-planes

SECTION_TRANSATION_PLANES:

TP_9_0:

$(ORBITER_PATH)orbi	ter.out-v.3\$

#-end
13139  > $(ORBITER_PATH)orbiter.out.-v.2.-draw_matrix.
13140  > -input_csv_file:TP9-0_incma.csv
13141  > -box_width:6.-bit_depth:8.-partition:6:91:91.-end
13142  > #open:TP9-0_incma_draw.bmp
13143
13144  TP_9_1:
13145  > $(ORBITER_PATH)orbiter.out.-v.3.
13146  > -define:F.-finite_field:-q:3.-end.
13148  > -define:PGL5.-linear_group:-PGL.5:F.-end.
13149  > -with:PGL4.-and:PGL5.-do.
13150  > -group_theoretic_activity.
13151  >   > -Andre_Bruck_Bose_construction:0."TP9-1".
13152  >   > -end
13153  > $(ORBITER_PATH)orbiter.out.-v.2.-draw_matrix.
13154  > -input_csv_file:TP9-1_incma.csv
13155  > -box_width:6.-bit_depth:8.-partition:6:91:91.-end
13156  > open:TP9-1_incma_draw.bmp
13157  > pdflatex:TP9-1_report.tex
13158  > open:TP9-1_report.pdf
13159
13160
13161  # ToDo:
13162
13163  TP_16_4:
13164  > $(ORBITER_PATH)orbiter.out.-v.3.
13165  > -define:F.-finite_field:-q:4.-end.
13167  > -define:PGL5.-linear_group:-PGL.5:F.-end.
13168  > -with:PGL4.-and:PGL5.-do.
13169  > -group_theoretic_activity.
13170  >   > -Andre_Bruck_Bose_construction:0."TP16-4-HALL".
13171  >   > -end
13172  > $(ORBITER_PATH)orbiter.out.-v.2.-draw_matrix.
13173  > -input_csv_file:TP16-4-HALL_incma.csv
13176  >   > -end
13177  > open:TP16-4-HALL_incma_draw.bmp
13178  > pdflatex:TP16-4-HALL_report.tex
13179  > open:TP16-4-HALL_report.pdf
13180
13181
13182  #"1200", // Hall spread
13183  #"81600", // Desarguesian spread
13184  #"576", // Semifield spread
13185
716
13186
13187
13188 TP_16_2:
13189 $ (ORBITER_PATH) orbiter.out -v.3 \\
13190 >define-F-finite_field-q.2-end \\
13191 >define-PGL8-linear_group-PGL.8.F-end \\
13192 >define-PGL9-linear_group-PGL.9.F-end \\
13193 >with-PGL8-and-PGL9-do \\
13194 >group_theoretic_activity \\
13195 >Andre_Bruck_Bose_construction-0."TP16_0_1008." \\
13196 >end \\
13197
13198
13199 #"1008", 
13200 #"1008", 
13201 #"1728", 
13202 #"216", 
13203 #"360", 
13204 #"288", 
13205 #"3600", 
13206 #"244800", 
13207 
13208 
13209 
13210 #================================================================================================================================
13211 #Section 11.3: Packings
13212
13213
13214 SECTION_PACKINGS:
13215
13216 spread_table_PG_3_4:
13217 >mkdir SPREAD_TABLES_4 
13218 $(ORBITER_PATH) orbiter.out -v.6 \\
13219 >define-F-finite_field-q.4-end \\
13220 >define-P-projective_space-3-F-end \\
13221 >define-T-spread_table-P.2:"0,1,2"."SPREAD_TABLES_4/". \\
13222 
13223 
13224 # 5096448 spreads 
13225 # 1020 self-dual spreads 
13226 # User time: 56:38 on Mac 
13227
13228 spread_table_PG_3_5_regular:
13229 >mkdir SPREAD_TABLES_5_REG 
13230 $(ORBITER_PATH) orbiter.out -v.6 \\
13231 >define-F-finite_field-q.5-end \\
13232 >define-P-projective_space-3-F-end \

717
13233 >>> -define-T-spread_table-P-2:12."SPREAD_TABLES_5_REG/\n13234 >>> -print_symbols
13235
13236
13237 #12/9/2020: 34·sec·on·Mac
13238 #12·is·the·index·of·the·regular·spread·in·the·classification·of·spreads
13239 #155000·spreads
13240
13241
13242 PG_3_5_desarguesian_spread:
13243 >>> $(ORBITER_PATH)orbiter.out-v.3\n13244 >>> -define-FQ-finite_field-q.25-end\n13245 >>> -define-Fq-finite_field-q.5-end\n13246 >>> -with-FQ-and-Fq-do\n13247 >>> -finite_field_activity\n13248 >>> -cheat_sheet_desarguesian_spread-2\n13249 >>> -end
13250 >>> pdflatex·Desarguesian_Spread_3_5.tex
13251 >>> open·Desarguesian_Spread_3_5.pdf
13252
13253 #Spread·elements·by·rank:·(0,·0,05,·36,·108,·72,·144,·581,·509,·686,·415,·639,·758
13254 ·,·285,·722,·332,·343,·202,·592,·473,·238,·675,·379,·166,·545,·249,·451:)
13255
13256 PG_3_5_element_of_order_31:
13257 >>> $(ORBITER_PATH)orbiter.out-v.6·-define-G\n13258 >>> -linear_group-GL.3.5-end\n13259 >>> -with-G-do\n13260 >>> -group_theoretic_activity\n13261 >>> -raise_to_the_power:"0,1,0,0,1,3,0,4".31\n13262 >>> -end
13263 >>> pdflatex·GL_3_5_power.tex
13264 >>> open·GL_3_5_power.pdf
13265
13266 PG_3_5_element_of_order_31_normalizer:
13267 >>> $(ORBITER_PATH)orbiter.out-v.6·-define-G\n13268 >>> -linear_group-PGL.4.5-end\n13269 >>> -with-G-do\n13270 >>> -group_theoretic_activity\n13271 >>> -normalizer_of_cyclic_subgroup"31":\n13272 >>> "2,0,0,0,·0,0,1,0,0,0,0,1,0,3,0,4":\n13273 >>> -end
13274 >>> mv·normalizer_of_31_in_PGL_4_5.tex·normalizer_of_31_AB_in_PGL_4_5.tex
13275 >>> pdflatex·normalizer_of_31_AB_in_PGL_4_5.tex
13276 >>> open·normalizer_of_31_AB_in_PGL_4_5.pdf
13277
13278
13279 PG_3.5_element_of_order_31_GL_normalizer:
13280 \>$\$(ORBITER\_PATH)\$orbiter.out-v-6.-define-G:\$
13281 \>$\$-linear_group-GL.4.5.-end:\$
13282 \>$\$-with-G-do:\$
13283 \>$\$-group_theoretic_activity:\$
13284 \>$\$-normalizer_of_cyclic_subgroup."124".\$
13285 \>$\$"2,0,0,0,0,0,1,0,0,0,1,0,3,0,4".\$
13286 \>$\$-end\$
13287 \>$\$#mv-normalizer_of_31_in_PGL.4.5.tex-normalizer_of_31_A.B_in_PGL.4.5.tex\$
13288 \>$\$pdflatex-normalizer_of_124_in_GL.4.5.tex\$
13289 \>$\$open-normalizer_of_124_in_GL.4.5.pdf\$
13290 PG_3.5_element_of_order_31_ME_normalizer:
13291 PG_3.5_assume_31_graph:
13292 \>$\$(ORBITER\_PATH)\$orbiter.out-v-5:\$
13293 \>$\$-define-F.-finite_field-q.5.-end:\$
13294 \>$\$-define-P.-projective_space-3.F.-end:\$
13295 \>$\$-define-T.-spread_table-P.2."12"."SPREAD\_TABLES.5\_REG/".\$
13296 \>$\$-define-PW.-packing_with_symmetry_assumption-T.\$
13297 \>$\$-H."H31"\$:\$(PGL.4.5\_SUBGROUP.31\_ME).-end:\$
13298 \>$\$-N."N31"\$:\$(PGL.4.5\_SUBGROUP.31\_ME\_NORMALIZER).-end\$
13299 \>$\$-end\$
13300 \>$\$-end\$
13301 \>$\$#mv-normalizer_of_31_in_PGL.4.5.tex-normalizer_of_31_ME_in_PGL.4.5.tex\$
13302 \>$\$pdflatex-normalizer_of_31_ME_in_PGL.4.5.tex\$
13303 \>$\$open-normalizer_of_31_ME_in_PGL.4.5.pdf\$
13304 #normalizer_has_order.1488=4*372=4*4*3*31
13305 PG_3.5_assume_31_graph:
13326  open-H31_reduced_spread_orbits_orbits_report.pdf
13327  pdflatex-H31_line_orbits_orbits_report.tex
13328  open-H31_line_orbits_orbits_report.pdf
13329  pdflatex-H31_line_orbits_orbits_report.tex
13330  open-H31_line_orbits_orbits_report.pdf
13331  pdflatex-N31_line_orbits_orbits_report.tex
13332  open-N31_line_orbits_orbits_report.pdf
13333  pdflatex-H31_point_orbits_orbits_report.tex
13334  open-H31_point_orbits_orbits_report.pdf
13335  pdflatex-N31_point_orbits_orbits_report.tex
13336  open-N31_point_orbits_orbits_report.pdf
13337  #pdflatex-H31_spread_orbits_orbits_report.tex
13338  #open-H31_spread_orbits_orbits_report.pdf
13339  #H31_spread_orbits_orbits.bin
13340  #H31_line_orbits_orbits_report.tex
13341  #H31_spread_orbits_orbit_types_report.tex
13342  #H31_spread_orbits_orbits.bin
13343  #H31_good_orbits
13344  #H31_spread_types_reduced_orbit_types_report.tex
13345  #H31_reduced_spread_orbits_orbits.bin
13346  #H31_fpc0_lo.graph
13347  #H31_fpc0_lo.sol.txt
13348  #H31_fpc0_lo_sol.csv
13349
13350  PG_3.5_assume_31_fpc0_lo_cliques:
13351  $(ORBITER_PATH)orbiter.out-v.2:\
13352  -define-G-graph-load-H31_fpc0_lo.graph-end\
13353  -with-G-do\
13354  -graph-theoretic_activity\
13355  -find_cliques-target_size:1-end-end\
13356  -print_symbols
13357  #H31_fpc0_lo_sol.txt
13358  #H31_fpc0_lo_sol.csv
13359
13360
13361
13362
13363  #ToDo: problem when computing the orbits of the normalizer:
13364
13365  PG_3.5_assume_31_read:
13366  $(ORBITER_PATH)orbiter.out-v.5:\
13367  -define-F-finite_field-q.5-end\
13368  -define-P-projective_space-3-F-end\
13369  -define-T-spread_table-P.2."12"."SPREAD_TABLES.5_REG/".\n13370  -define-PW-packing_with_symmetry_assumption-T:\
13371  -H."H31"$(PGL_4.5_SUBGROUP_31_ME)-end\
13372  -N."H31"$(PGL_4.5_SUBGROUP_31_ME)-end\n
720
PG_3_5_packing_0 dualize:

PG_3_5_assume_3:

PG_3_5_packing_0 dualize:

PG_3_5_assume_3:
13416 ▶ ▶ -print_symbols
13417
13418 PG_3.5_3B_create_graph_on_long_orbits:
13419 ▶ $(ORBITER_PATH)orbiter.out-v.5\n13420 ▶ ▶ -define-F-finite_field-q.5-end\n13421 ▶ ▶ -define-P-projective_space-3-F-end\n13422 ▶ ▶ -define-T-spread_table-P.2:"12"."SPREAD_TABLES.5_REG/"\n13423 ▶ ▶ -define-PW-packing_with_symmetry_assumption-T\n13424 ▶ ▶ ▶ ▶ -H."H3B_ME"$(PGL_4.5_SUBGROUP_3B_ME)-end\n13425 ▶ ▶ ▶ ▶ -N."N3B_ME"$(PGL_4.5_SUBGROUP_3B_ME_NORMALIZER)-end\n13426 ▶ ▶ ▶ ▶ -end\n13427 ▶ ▶ -define-PWF-packing_choose_fixed_points-PW\n13428 ▶ ▶ ▶ 1-W-problem_label-N3B_ME_fixp_cliques-end\n13429 ▶ ▶ -define-L-packing_long_orbits-PWF\n13430 ▶ ▶ ▶ -orbit_length.3\n13431 ▶ ▶ ▶ -clique_size.10\n13432 ▶ ▶ ▶ -list_of_cases_from_file-N3B_ME_fixp_cliques.csv\n13433 ▶ ▶ ▶ -create_graphs\n13434 ▶ ▶ ▶ -end\n13435 ▶ ▶ -print_symbols
13436
13437 #16120-vertices
13438 #creates-H3B_ME_fpc0_lo.graph
13439
13440 PG_3.5_assume_3B_fpc0_lo_cliques:
13441 ▶ $(ORBITER_PATH)orbiter.out-v.2\n13442 ▶ ▶ -define-G-graph-load-H3B_ME_fpc0_lo.graph-end\n13443 ▶ ▶ -with-G-do\n13444 ▶ ▶ ▶ -graph_theoretic_activity-find_cliques\n13445 ▶ ▶ ▶ -target_size.10-end-end\n13446 ▶ ▶ ▶ -print_symbols
13447
13448 #768-solutions
13449 #User-time:.8:16
13450
13451 PG_3.5_assume_3B_long_read:
13452 ▶ $(ORBITER_PATH)orbiter.out-v.5\n13453 ▶ ▶ -define-F-finite_field-q.5-end\n13454 ▶ ▶ -define-P-projective_space-3-F-end\n13455 ▶ ▶ -define-T-spread_table-P.2:"12"."SPREAD_TABLES.5_REG/"\n13456 ▶ ▶ -define-PW-packing_with_symmetry_assumption-T\n13457 ▶ ▶ ▶ ▶ -H."H3B_ME"$(PGL_4.5_SUBGROUP_3B_ME)-end\n13458 ▶ ▶ ▶ ▶ -N."N3B_ME"$(PGL_4.5_SUBGROUP_3B_ME_NORMALIZER)-end\n13459 ▶ ▶ ▶ ▶ -end\n13460 ▶ ▶ -define-PWF-packing_choose_fixed_points-PW\n13461 ▶ ▶ ▶ 1-W-problem_label-N3B_ME_fixp_cliques-end\n13462 ▶ ▶ -define-L-packing_long_orbits-PWF\n
722
13463 · · · -orbit_length.3.-clique_size.10.\ 
13464 · · · -list_of_cases_from_file.N3B_ME_fixp_cliques.csv.\ 
13465 · · · -read_solutions.\ 
13466 · · · -end.\ 
13467 · · · -print_symbols
13468
13469 13470 #total_number_of_packings.=.768
13471 13472 #written_file.N3B_ME_fixp_cliques_count.csv.of_size.38
13473 #packing_long.orbits: list_of_cases_from_file.before_save_packings_by_case
13474 #written_file.H3B_ME_packings.csv.of_size.150540
13475
13476
13477
13478
13479 PG.3.5_packing0_print:
13480 · · · $(ORBITER_PATH)orbiter.out.-v.5.\ 
13481 · · · -define.F.finite_field.-q.5.-end.\ 
13482 · · · -define.P.projective_space.3.F.-end.\ 
13483 · · · -define.T.spread_table.P.2."12"."SPREAD_TABLES.5_REG\/".\ 
13484 · · · -define.P.W.packing.with_symmetry_assumption.T.\ 
13485 · · · · · · -H."H31_ME".(PGL.4.5_SUBGROUP_31_ME).-end.\ 
13486 · · · · · · -N."N31_ME".(PGL.4.5_SUBGROUP_31_ME_NORMALIZER).-end.\ 
13487 · · · · · · -end.\ 
13488 · · · -define.P.WF.packing.choose_fixed_points.P.W.\ 
13489 · · · · · · 0.-W.-problem_label.N3B_ME_fixp_cliques.-end.\ 
13490 · · · · · · -with.P.WF.-do.-packing.fixed_points_activity.\ 
13491 · · · · · · -print_packing.(PG.3.5_Packing.0_WITH_AGO3_FIXP444).\ 
13492 · · · -end
13493
13494
13495
13496
13497 #H-in-the-action-on-point.has.6-orbits.of.length.1.and.50-orbits.of.length.3
13498 #the-fixed-points.make-up.a.line.It.is.the.first.of.the-two-special-lines.
13499 #line-orbits.2.and.3.make-up.the-second.of.the-special-line.
13500
13501 #H-in-the-action-on-lines.has.270-orbits.
13502 #There.are.2-orbits.of.length.1.and.268-orbits.of.length.3
13503 #the-two-orbits.of-length-one.are-the-special-lines.
13504 #the-first-line.is.fixed-pointwise.
13505
13506
13507 #spread.orbits.of.length.1.\"\ 
13508 # Orbit.0:
13509 $$

723
13510 #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)
13511 13512 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"
13513 13514 #> ▶ -define L- packing_long_orbits: PWF- ▶ -orbit_length: 31- ▶ -clique_size: ▶ -create_graphs- ▶ -end- ▶ \ 13515 13516 13517 13518 PG_3_5_H3_orbit_of_length_1:
13519 ▶ $(ORBITER_PATH)orbiter.out- ▶ -v- ▶ 5- ▶ \ 13520 ▶ ▶ ▶ -define F finite_field- ▶ -q- ▶ 5- ▶ -end- ▶ \ 13521 ▶ ▶ ▶ -define P projective_space- ▶ 3- ▶ -F- ▶ -end- ▶ \ 13522 ▶ ▶ ▶ -define T spread_table- ▶ P- ▶ 2- ▶ "12- ▶ "SPREAD_TABLES_5_REG"- ▶ \ 13523 ▶ ▶ ▶ ▶ ▶ -with T- ▶ -do- ▶ \ 13524 ▶ ▶ ▶ ▶ ▶ ▶ -spread_table_activity- ▶ \ 13525 ▶ ▶ ▶ ▶ ▶ ▶ ▶ -find_spread_and_dualize $(ORBITE OF_LENGTH_1)- ▶ \ 13526 ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ ▶ -end- ▶ \ 13527 13528 #The given spread has index 0 in the spread table
13529 #The dual spread has index 1 in the spread table
13530 13531 13532 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"
13533 13534 13535 13536 #H31_packings.csv
13537 #H3B_ME_packings.csv
13538 13539 13540 PG_3_5_packings_compare:
13541 ▶ $(ORBITER_PATH)orbiter.out- ▶ -v- ▶ 5- ▶ \ 13542 ▶ ▶ -define F finite_field- ▶ -q- ▶ 5- ▶ -end- ▶ \ 13543 ▶ ▶ -define P projective_space- ▶ 3- ▶ -F- ▶ -end- ▶ \ 13544 ▶ ▶ -define T spread_table- ▶ P- ▶ 2- ▶ "12- ▶ "SPREAD_TABLES_5_REG"- ▶ \ 13545 ▶ ▶ -define PWF packing_with_symmetry_assumption T- ▶ \ 13546 ▶ ▶ ▶ -with T- ▶ -H- ▶ "H31_ME" $(PGL_4_5_SUBGROUP_31_ME)- ▶ -end- ▶ \ 13547 ▶ ▶ ▶ -with T- ▶ -N- ▶ "N31_ME" $(PGL_4_5_SUBGROUP_31_ME_NORMALIZER)- ▶ -end- ▶ \ 13548 ▶ ▶ ▶ -end- ▶ \ 13549 ▶ ▶ -define PWF packing_choose_fixed_points PW- ▶ \ 13550 ▶ ▶ ▶ -with PWF- ▶ -maximum L- ▶ -W- ▶ -end- ▶ \ 13551 724
13551 \> \> with-PWF-do-packing_fixed_points_activity\ 
13552 \> \> \> compare_files_of_packings-H31.packings.csv\H3B\ME\packings.csv\ 
13553 \> \> end 
13554 
13555 PG_3.5.packings_sort_each_row: 
13556 \> $(ORBITER\PATH)\orbiter.out-v.5-csv_file_sort_each_row-H31.packings.csv 
13557 \> $(ORBITER\PATH)\orbiter.out-v.5-csv_file_sort_each_row-H3B\ME\packings.csv 
13558 
13559 \#H31.packings_sorted.csv 
13560 
13561 
13562 
13563 
13564 PG_3.5.assume_31.classify: 
13565 \> $(ORBITER\PATH)\orbiter.out-v.2\ 
13566 \> \> -define-C-combinatorial_objects\ 
13567 \> \> \> -file_of_packings_through_spread_table\ 
13568 \> \> \> \> H31.packings.csv\ 
13569 \> \> \> \> SPREAD\TABLES\5\REG/spread_25_spreads.csv\ 
13570 \> \> \> \> 5\ 
13571 \> \> -end\ 
13572 \> \> \> -define-F-finite_field-q.5\-end\ 
13573 \> \> \> -define-P-projective_space\3\F\-end\ 
13574 \> \> \> -with-C-do\ 
13575 \> \> \> -combinatorial_object_activity\ 
13576 \> \> \> \> -canonical_form\PG\P\ 
13577 \> \> \> \> -classification_prefix\H31\packings\ 
13578 \> \> \> -end\ 
13579 \> \> \> -report\ 
13580 \> \> \> -end 
13581 \> pdflatex\H31\packings\classification.tex 
13582 \> open\H31\packings\classification.pdf 
13583 
13584 \#ToDo: 
13585 
13586 PG_3.5.assume_3B.classify: 
13587 \> $(ORBITER\PATH)\orbiter.out-v.2\ 
13588 \> \> -define-F-finite_field-q.5\-end\ 
13589 \> \> -define-P-projective_space\3\F\-end\ 
13590 \> \> -with-P-do\ 
13591 \> \> -projective_space_activity\ 
13592 \> \> \> -canonical_form\PG\ 
13593 \> \> \> \> -input\-file_of_packings_through_spread_table\ 
13594 \> \> \> \> \> H3B\ME\packings.csv\ 
13595 \> \> \> \> \> \> SPREAD\TABLES\5\REG/spread_25_spreads.csv\ 
13596 \> \> \> \> \> \> -end\ 
13597 \> \> \> \> \> \> -classification_prefix\H3B\packings\
13615 # Section 11.4: BLT-sets
13618
13619 SECTION_BLT_SETS:
13620
13621 BLT_5_1:
13622 $(ORBITER\_PATH)\$ orbiter.out -v -2:
13623 \$ -define F:\_finite_field\_q 5 -end\$
13624 \$ -define 0:\_orthogonal\_space 0急剧 F: -end\$
13625 \$ -with 0:\_do:\_orthogonal\_space\_activity\$)
13626 \$ -create BLT\_set\_catalogue 1 -end\$
13627 \$ -end\$
13628 \$ pdflatex catalogue\_q5\_isol0.tex
13629 open catalogue\_q5\_isol0.pdf
13630
13631 BLT_5 Linear:
13632 $(ORBITER\_PATH)\$ orbiter.out -v -2:
13633 \$ -define F:\_finite_field\_q 5 -end\$
13634 \$ -define 0:\_orthogonal\_space 0急剧 F: -end\$
13635 \$ -with 0:\_do:\_orthogonal\_space\_activity\$)
13636 \$ -create BLT\_set\_family "Linear" -end\$
13637 \$ -end\$
13638 \$ pdflatex BLT Linear\_q5.pdf
13639
13640 BLT_9 K1:
13641 $(ORBITER\_PATH)\$ orbiter.out -v -2:
13642 \$ -define F:\_finite_field\_q 9 -end\$
13643 \$ -define 0:\_orthogonal\_space 0急剧 F: -end\$
13644 \$ -with 0:\_do:\_orthogonal\_space\_activity\$)
13645 \$ -create BLT\_set\_family "K1" -end\$
13646
13645 \> \> -end
13646 \> pdflatex BLT_K1_q9.tex
13647 \> open-BLT_K1_q9.pdf
13648
13649
13650
13651
13652 BLT_11_0:
13654 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,
13655 \> \> -define F:finite_field=q:11\,-end\,
13656 \> \> -define 0:-orthogonal_space=0.5:F\,-end\,
13657 \> \> \> -with 0:-do:-orthogonal_space\activity\,
13658 \> \> \> \> -create_BLT_set\,-catalogue=0:-end\,
13659 \> \> \> -end
13660 \> #pdflatex_0.1_6_2_report.tex
13661
13662
13663 BLT_11_Fisher:
13664 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,
13665 \> \> -define F:finite_field=q:11\,-end\,
13666 \> \> -define 0:-orthogonal_space=0.5:F\,-end\,
13667 \> \> \> -with 0:-do:-orthogonal_space\activity\,
13668 \> \> \> \> -create_BLT_set\,-family="Fisher":-end\,
13669 \> \> \> -end
13670 \> pdflatex BLT_Fisher_q11.tex
13671 \> open-BLT_Fisher_q11.pdf
13672
13673 BLT_11_Mondello:
13674 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,
13675 \> \> -define F:finite_field=q:11\,-end\,
13676 \> \> -define 0:-orthogonal_space=0.5:F\,-end\,
13677 \> \> \> -with 0:-do:-orthogonal_space\activity\,
13678 \> \> \> \> -create_BLT_set\,-family="Mondello":-end\,
13679 \> \> \> -end
13680 \> pdflatex BLT_Mondello_q11.tex
13681 \> open-BLT_Mondello_q11.pdf
13682
13683
13684 BLT_13_FTWKB:
13685 \> $(\text{ORBITER\_PATH})\text{orbiter.out}\,-v.2\,
13686 \> \> -define F:finite_field=q:11\,-end\,
13687 \> \> -define 0:-orthogonal_space=0.5:F\,-end\,
13688 \> \> \> -with 0:-do:-orthogonal_space\activity\,
13689 \> \> \> \> -create_BLT_set\,-family="FTWKB":-end\,
13690 \> \> \> -end
13691 \> pdflatex BLT_FTWKB_q11.tex

727
for $K_2$, $q$ must be congruent to $2$ or $3 \mod 5$

BLT_{13,K2}:

- define $F$ - finite field $- q_{13} -$ end\n
- define $O$ - orthogonal space $0:5:F$ - end\n
- with $O$ - do - orthogonal space activity\n
- create BLT set - family "$Kantor2" -$end\n
- end

BLT set starter 5:

BLT set starter 14:

BLT set graphs:

BLT set cliques:
# Chapter 12: Graph Theory

## Section 12.1: Creating Graphs

SECTION_CREATING_GRAPHS:

Cycle_13:

```bash
$(ORBITER_PATH)orbiter.out-v.2
```

```bash
define Gamma:-graph
```

```bash
-cycle_13
```

```bash
-end
```

triangle_graph:

```bash
echo $(TRIANGLE_GRAPH) > triangle_graph.csv
```

```bash
$(ORBITER_PATH)orbiter.out-v.6
```

```bash
define G:-graph
```

```bash
-load_csv_no_border
```

```bash
triangle_graph.csv
```

```bash
-end
```

Chain_232:

```bash
$(ORBITER_PATH)orbiter.out-v.2
```

```bash
define P1:-vector-dense 2,3,2-end
```
13786 \> \> \> -define P2 \> vector \> dense 2,3,2 \> -end \>
13787 \> \> \> -define Gamma \> -graph \>
13788 \> \> \> -chain_graph P1 \> P2 \>
13789 \> \> \> -end \>
13790 \>
13791 \>
13792 \>
13793 \>
13794 Paley_13_graph: \>
13795 \> $(ORBITER_PATH) orbiter.out -v2 \>
13796 \> \> -define Gamma \> -graph Paley-13 \> -end \>
13797 \>
13798 \>
13799 \>
13800 \>
13801 \>
13802 trihedral_pair_graph: \>
13803 \> $(ORBITER_PATH) orbiter.out -v2 \>
13804 \> \> -define Gamma \>
13805 \> \> \> -graph trihedral_pair_disjointness_graph \>
13806 \> \> \> -end \>
13807 \>
13808 \>
13809 small_graph: \>
13810 \> $(ORBITER_PATH) orbiter.out -v2 \>
13811 \> \> -define G \> graph \> edges as pairs 5 "0,1,0,2,0,3,0,4,1,3,1,4,2,4" \> -end \>
13812 \>
13813 \>
13814 \>
13815 \>
13816 petersen: \>
13817 \> $(ORBITER_PATH) orbiter.out -v2 \>
13818 \> \> -define G \> graph \> Johnson 5 2 0 \> -end \>
13819 \>
13820 \>
13821 \>
13822 \>
13823 Johnson_6_2_0: \>
13824 \> $(ORBITER_PATH) orbiter.out -v2 \>
13825 \> \> -define G \> graph \> Johnson 6 2 0 \> -end \>
13826 \>
13827 Hamming_graph_3: \>
13828 \> $(ORBITER_PATH) orbiter.out -v2 \>
13829 \> \> -define G \> graph \> Hamming 3 2 \> -end \>
13830 \>
13831 \>
13832 Hamming_graph_7:
There is a unique distance-regular graph \( \Gamma \) with intersection array \( \{10, 8, 2; 1, 1, 4, 5\} \). It was constructed in Cohen (1981), and uniqueness (given the intersection array) was proved in Cohen & Tits (1985).

HJ graph:
```
$\$(ORBITER_PATH)orbiter.out\$ -v 2 \\
define G graph
load csv no border
halljanko315.csv
end
```

HJ315 orbital graph 3:
```
$\$(ORBITER_PATH)orbiter.out\$ -v 2 \\
define gens -vector -file
halljanko315.gens.csv
end
define G permutation_group
bsgs halljanko315."File\_halljanko315"
315 1209600 "0,1,2" 6 gens
end
```

HJ d2 graph:
```
$\$(ORBITER_PATH)orbiter.out\$ -v 2 \\
define G graph
load csv no border
halljanko315.csv
distance 2
end
```

Cayley \( Z_{11} \) mod3:
```
$\$(ORBITER_PATH)orbiter.out\$ -v 2 \\
define F finite_field -q 11
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" \
```
Cayley_Sym4_coxeter:
$\text{(ORBITER\_PATH)}\text{orbiter\_out\_v\_2}$
\begin{verbatim}
\define S vector dense "1,0,2,3,0,2,1,3,0,1,3,2" end
\define G permutation_group symmetric_group 4
\define Gamma graph
\end{verbatim}
\begin{verbatim}
\Cayley_graph G S
\end{verbatim}

Cayley_Sym4_star:
$\text{(ORBITER\_PATH)}\text{orbiter\_out\_v\_2}$
\begin{verbatim}
\define S vector dense "1,0,2,3,2,1,0,3,3,1,2,0" end
\define G permutation_group symmetric_group 4
\define Gamma graph
\end{verbatim}
\begin{verbatim}
\Cayley_graph G S
\end{verbatim}

SECTION GRAPH THEORETIC ACTIVITIES:

triangle_graph_properties:
\begin{verbatim}
\echo $(TRIANGLE\_GRAPH)>triangle_graph.csv
\define G graph
\define load_csv_no_border
\define triangle_graph.csv
\define -end
\define with G do
\define -graph_theoretic_activity_properties
\end{verbatim}
\begin{verbatim}
\end
\end{verbatim}
Cycle_13_draw:
$\langle ORBITER\_PATH\rangle orbiter.out -v 2: \n\% \% -define-Gamma -graph -cycle 13 -end \n\% -with-Gamma -do \n\% -graph_theoretic_activity -export_csv -end \n\% -with-Gamma -do \n\% -graph_theoretic_activity -export_graphviz -end \n$\langle ORBITER\_PATH\rangle orbiter.out -v 2 -draw_matrix \n\% -input_csv_file Cycle_13.csv \n\% -box_width 20 -bit_depth 8 -partition 4 13 13 -end \n\% dot -Tpng Cycle_13.gv > Cycle_13.png \n\% twopi -Tpng Cycle_13.gv > Cycle_13.png \n\% open_cycle_13_draw.bmp \n\% pdflatex Cycle_13_report.tex \n\% open_cycle_13_report.pdf

Cycle_9_eigenvalues:
$\langle ORBITER\_PATH\rangle orbiter.out -v 2: \n\% -define-Gamma -graph \n\% -cycle 9 -end \n\% -with-Gamma -do \n\% -graph_theoretic_activity -eigenvalues -end \n$\langle ORBITER\_PATH\rangle orbiter.out -v 2 -draw_matrix \n\% -input_csv_file Paley_13.csv \n\% -box_width 20 -bit_depth 8 -partition 4 13 13 -end \n\% dot -Tpng Paley_13.gv > Paley_13.png \n\% open_paley_13_draw.bmp \n\% pdflatex Paley_13_eigenvalues.tex \n\% open_paley_13_eigenvalues.pdf

Paley_13_draw:
$\langle ORBITER\_PATH\rangle orbiter.out -v 2: \n\% -define-Gamma -graph -Paley 13 -end \n\% -with-Gamma -do \n\% -graph_theoretic_activity -export_csv -end \n\% -with-Gamma -do \n\% -graph_theoretic_activity -export_graphviz -end \n$\langle ORBITER\_PATH\rangle orbiter.out -v 2 -draw_matrix \n\% -input_csv_file Paley_13.csv \n\% -box_width 20 -bit_depth 8 -partition 4 13 13 -end \n\% dot -Tpng Paley_13.gv > Paley_13.png \n\% open_paley_13_draw.bmp

Paley_13_eigenvalues:
$\langle ORBITER\_PATH\rangle orbiter.out -v 2: \n\% -define-Gamma -graph \n\% -Paley 13 -end \n\% -with-Gamma -do \n
Paley_13_eigenvalues:
Cayley_Z11_1mod3_draw:

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

-draw_options -xin 2000000 -yin 2000000 -embedded -radius 20000 -end
-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
-define Gamma -graph

Cayley_graph G S -end

with Gamma -do
-define S -vector -dense

"1,0,2,3,0,2,1,3,0,1,3,2" -end
-define G -permutation_group -symmetric_group 4
-define Gamma -graph
-define S -vector -dense

Cayley_graph G S -end

with Gamma -do
-define Gamma -graph
-define Gamma -draw

Cayley_Z11_1mod3_draw:

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

-define S -vector -dense

"1,0,2,3,0,2,1,3,0,1,3,2" -end
-define G -permutation_group -symmetric_group 4
-define Gamma -graph

Cayley_graph G S -end

with Gamma -do
-define Gamma -graph
-define Gamma -draw

Cayley_Z11_1mod3_draw:

$(\text{ORBITER PATH})\text{orbiter.out} -v.2$
\begin{verbatim}
14019 >> > -draw_options-xin:1000000-yin:1000000-
14020 >> > -embedded-radius:10000-nodes_empty:end-
14021 >> > -define-S-vector:dense
14022 >> > "1,0,2,3,4,0,2,1,3,4,0,1,3,2,4,0,1,2,4,3":end-
14023 >> > -define-G-permutation_group-symmetric_group:5-
14024 >> > -end-
14025 >> > -define-Gamma-graph
14026 >> > -Cayley_graph-G:S:
14027 >> > -end-
14028 >> > -with-Gamma-do-
14029 >> > -graph_theoretic_activity-draw-end
14030 >> pdflatex-Cayley_graph_Perm5_draw.tex
14031 >> open-Cayley_graph_Perm5_draw.pdf
14032
14033 Cayley_Sym4_star_eigenvalues_and_draw:
14034 >> $(ORBITER_PATH)orbiter.out-v:2-
14035 >> > -draw_options-xin:1000000-yin:1000000-embedded-end-
14036 >> > -define-S-vector:dense"1,0,2,3,2,1,0,3,3,1,2,0":end-
14037 >> > -define-G-permutation_group-symmetric_group:4-
14038 >> > -end-
14039 >> > -end-
14040 >> > -with-Gamma-do-
14041 >> > -graph_theoretic_activity-eigenvalues-end-
14042 >> > -with-Gamma-do-
14043 >> > -graph_theoretic_activity-draw-end
14044 >> pdflatex-Cayley_graph_Perm4_draw.tex
14045 >> open-Cayley_graph_Perm4_draw.pdf
14046 >> with-Gamma-do-
14047 >> -graph_theoretic_activity-draw-end
14048 >> pdflatex-Cayley_graph_Perm4_eigenvalues.tex
14049 >> open-Cayley_graph_Perm4_eigenvalues.pdf
14050 >> small_graph_draw:
14051 >> $(ORBITER_PATH)orbiter.out-v:2-
14052 >> > -define-G-graph-edges_as_pairs:5-
14053 >> > "0,1,0,2,0,3,0,4,1,3,1,4,2,4":end-
14054 >> > -end-
14055 >> > -with-G-do-
14056 >> > -graph_theoretic_activity-export_csv-end-
14057 >> > -with-G-do-
14058 >> > -graph_theoretic_activity-export_graphviz-end-
14059 >> > -graph_theoretic_activity-save-end
14060 >> $(ORBITER_PATH)orbiter.out-v:2-draw_matrix-
14061 >> > -input_csv_file-graph_v5_e7.csv-
\end{verbatim}
petersen_draw:

$(ORBITER_PATH)orbiter.out -v:2 -define G -graph -Johnson-5:2:0 -end -graph_theoretic_activity -export_csv -end -with G -do -graph_theoretic_activity -export_graphviz -end -with G -do -graph_theoretic_activity -save -end $(ORBITER_PATH)orbiter.out -v:2 -draw_matrix -input_csv_file:Johnson_5_2_0.csv -box_width:40 -bit_depth:24 -partition:4 "1,1,1,1" "1,1,1,1" -end dot -Tpng:Johnson_5_2_0.gv >Johnson_5_2_0.png

Johnson_6_2_0_draw:


Hamming_graph_3_draw:

Hamming_graph_7.draw:

$\text{(ORBITER\_PATH)\_orbiter.out\_\_v.2\_\_draw_matrix}\$

$\text{-input\_csv\_file\_Hamming\_3\_2.csv}\$

$\text{-box\_width\_40\_\_bit\_depth\_24}\$

$\text{-partition\_4\_"1,1,1,1,1,1,1,1"\_\_end}\$

$\text{dot\_\_Tpng\_\_Hamming\_3\_2.gv}\rightarrow\text{Hamming\_3\_2.png}\$

$\text{with\_\_Gamma\_\_do}\$

$\text{-graph\_theoretic\_activity\_\_export\_csv\_\_end}\$

$\text{-with\_\_G\_\_do}\$

$\text{-graph\_theoretic\_activity\_\_export\_graphviz\_\_end}\$

$\text{-with\_\_G\_\_do}\$

$\text{-graph\_theoretic\_activity\_\_save\_\_end}\$

$\text{$\text{(ORBITER\_PATH)\_orbiter.out\_\_v.2\_\_draw_matrix}$}\$

$\text{-input\_csv\_file\_Hamming\_7\_2.csv}\$

$\text{-box\_width\_8\_\_bit\_depth\_24\_\_partition\_4\_128\_128\_\_end}\$

$\text{dot\_\_Tpng\_\_Hamming\_7\_2.gv}\rightarrow\text{Hamming\_7\_2.png}\$

$\text{with\_\_Gamma\_\_do}\$

$\text{-graph\_theoretic\_activity\_\_export\_csv\_\_end}\$

$\text{-with\_\_G\_\_do}\$

$\text{-graph\_theoretic\_activity\_\_properties\_\_end}\$

$\text{Chain\_232\_properties:}\$

$\text{$\text{(ORBITER\_PATH)\_orbiter.out\_\_v.2}$}\$

$\text{-define\_\_P1\_\_vector\_\_dense\_2,3,2\_\_end}$

$\text{-define\_\_P2\_\_vector\_\_dense\_2,3,2\_\_end}$

$\text{-define\_\_Gamma\_\_graph}$

$\text{\_\_chain\_\_graph\_P1\_P2}\$

$\text{-end}\$

$\text{\_\_with\_\_Gamma\_\_do}\$

$\text{-graph\_theoretic\_activity\_\_properties\_\_end}\$

$\text{Chain\_232\_eigen:}\$

$\text{$\text{(ORBITER\_PATH)\_orbiter.out\_\_v.2}$}\$

$\text{-define\_\_P1\_\_vector\_\_dense\_2,3,2\_\_end}$

$\text{-define\_\_P2\_\_vector\_\_dense\_2,3,2\_\_end}$

$\text{-define\_\_Gamma\_\_graph}$

$\text{\_\_chain\_\_graph\_P1\_P2}\$
#need-the-file: halljanko315.csv

HJ_properties:

\$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ 6\$

-define \(-G\)-graph-

-\text{load\ csv\ no\ border} -hallenganko315.csv-

-end-

-with \(-G\)-do-

-graph_theoretic_activity\ -properties-

-end-

\text{graph\ theoretic\ activity\ -properties} -end-

#Degree\ type:\ (10^{315})-

HJ_d2_properties:

\$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\ 6\$

-define \(-G\)-graph-

-\text{load\ csv\ no\ border} -hallenganko315.csv-

-distance 2-

-end-

-with \(-G\)-do-

-graph_theoretic_activity-

-properties-

-end-

\text{graph\ theoretic\ activity\ -properties} -end-

#Degree\ type:\ (80^{315})-

PGO.5.2_collinearity_graph: O.5.2_incidence_matrix.csv-

-define \(-\text{inc\ -vector\ -file} 0.5.2\_incidence_matrix.csv\)-end-
triangular_pair_graph_draw:

```bash
define-Gamma-graph-collinearity_graph-\text{Inc}-end
```
14254 \#pdflatex-graphs_v4_poset_lvl_6.tex
14255 \#open-graphs_v4_poset_lvl_6.pdf
14256
tournament_classify_4:
14259 $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.2$
14260 \#define GC \#graph_classification\n14261 \#n.4-tournament\n14262 \#define \#poset_classification_control\n14263 \#define \#problem_label.tournament_4-depth.6-
14264 \#define \#draw_options-radius.250-embedded-end\n14265 \#define \#draw_poset\n14266 \#draw_options-radius.250-embedded-end\n14267 \#draw_options-radius.250-embedded-end\n14268 \#draw_options-radius.250-embedded-end\n14269 \#draw_options-radius.250-embedded-end\n14270 \#draw_options-radius.250-embedded-end\n14271 \#draw_options-radius.250-embedded-end\n14272 \#draw_options-radius.250-embedded-end\n14273 \#draw_options-radius.250-embedded-end\n14274 \#draw_options-radius.250-embedded-end\n14275 \#draw_options-radius.250-embedded-end\n14276 \#draw_options-radius.250-embedded-end\n14277 \#draw_options-radius.250-embedded-end\n14278 \#draw_options-radius.250-embedded-end\n14279 \#draw_options-radius.250-embedded-end\n14280 \#draw_options-radius.250-embedded-end\n14281 \#draw_options-radius.250-embedded-end\n14282
graph_classify_8_r3:
14286 $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v.3$
14287 \#define GC \#graph_classification\n14288 \#n.8-regular.3\n14289 \#define \#poset_classification_control\n14290 \#define \#problem_label.graphs_v8_r3-depth.12-
14291 \#define \#draw_options-radius.250-
14292 \#define \#draw_options-radius.250-
14293 \#define \#draw_options-radius.250-
14294 \#define \#draw_options-radius.250-
14295 \#define \#draw_options-radius.250-
14296 \#define \#draw_options-radius.250-
14297 \#define \#draw_options-radius.250-
14298 \#define \#draw_options-radius.250-
14299 \#define \#draw_options-radius.250-
14300 \#define \#draw_options-radius.250-
Section 12.4: Graph Theory: Clique finding

small_graph_cliques:
$(ORBITER\ PATH)orbiter.out\ -v\ 10\$
$\>$ \> -define\ G\ -graph\ -load\ graph\ v5\ e7.colored_graph\ -end\$
$\>$ \> -with\ G\ -do\$
$\>$ \> -graph\ theoretic\ activity\$
$\>$ \> -find\ cliques\ -target\ size\ 3\$
$\>$ \> -end
$\>$ \> -nb\ sol\ =\ 3

small_graph_cliques_Sajeeb:
$(ORBITER\ PATH)orbiter.out\ -v\ 2\$
$\>$ \> -define\ G\ -graph\ -load\ graph\ v5\ e7.colored_graph\ -end\$
$\>$ \> -with\ G\ -do\$
$\>$ \> -graph\ theoretic\ activity\$
$\>$ \> -find\ cliques\ -Sajeeb\ -target\ size\ 3\$
$\>$ \> -end
$\>$ \> -nb\ sol\ =\ 3

Paley_13\ aut:
$(ORBITER\ PATH)orbiter.out\ -v\ 2\$
14348 ▷ ▷ -define-Gamma-graph-Paley.13-end\n14349 ▷ ▷ -with-Gamma-do\n14350 ▷ ▷ -graph_theoretic_activity\n14351 ▷ ▷ -automorphism_group\n14352 ▷ ▷ -end\n14353
14354 #writes-Paley.13_group.makefile
14355 #User.time:.0.of.a.second,dt=0.tps=.100
14356 #nb.calls_to_densenauty=1
14357
14358
14359
14360 Paley.13:
14361 ▷ $(ORBITER_PATH)orbiter.out-v.2\n14362 ▷ ▷ -define-gens-vector-file-Paley.13.gens.csv-end\n14363 ▷ ▷ -define-G-permutation_group\n14364 ▷ ▷ -bsgs-Paley.13:"Paley\_13".13.78."0,1".3.gens-end\n14365
14366
14367 Paley.13_cliques.classify:
14368 ▷ $(ORBITER_PATH)orbiter.out-v.4\n14369 ▷ ▷ -define-gens-vector-file-Paley.13.gens.csv-end\n14370 ▷ ▷ -define-G-permutation_group\n14371 ▷ ▷ -bsgs-Paley.13:"Paley\_13".13.78."0,1".3.gens-end\n14372 ▷ ▷ -define-Gamma-graph-Paley.13-end\n14373 ▷ ▷ -with-G-do\n14374 ▷ ▷ -group_theoretic_activity\n14375 ▷ ▷ ▷ -poset_classification_control\n14376 ▷ ▷ ▷ ▷ -W\n14377 ▷ ▷ ▷ ▷ -problem_label-Paley13_cliques\n14378 ▷ ▷ ▷ ▷ -clique_test-Gamma\n14379 ▷ ▷ ▷ ▷ -depth.5\n14380 ▷ ▷ ▷ ▷ -end\n14381 ▷ ▷ ▷ -orbits_on_subsets.5\n14382 ▷ ▷ ▷ -report\n14383 ▷ ▷ ▷ -end\n14384
14385 #User.time:.01.of.a.second,dt=1.tps=.100
14386
14387
14388 Paley.13_cliques:
14389 ▷ $(ORBITER_PATH)orbiter.out-v.10\n14390 ▷ ▷ -define-Gamma-graph-Paley.13-end\n14391 ▷ ▷ -with-Gamma-do\n14392 ▷ ▷ -graph_theoretic_activity\n14393 ▷ ▷ ▷ -find_cliques-target_size.3\n14394 ▷ ▷ ▷ -end
```
14395
14396
14397
14398
14399
14400
14401 PGO_5_2_cliques:0_5_2_incidence_matrix.csv-
14402 $\langle$ORBITER\PATH$\rangle$orbiter.out--v.3\$
14403 $\langle$define\-Inc\-vector\-file0_5_2_incidence_matrix.csv\-end$\$
14404 $\langle$define\-Gamma\-graph\-collinearity_graph\-Inc\-end$\$
14405 $\langle$with\-Gamma\-do$\$
14406 $\langle$graph\-theoretic\_activity$\$
14407 $\langle$find\_cliques\-target\_size.3\-end$\$
14408 $\langle$end
14409
14410
14411 HJ_d2_c5:
14412 $\langle$ORBITER\PATH$\rangle$orbiter.out--v.6\$
14413 $\langle$define\-G\-graph$\$
14414 $\langle$load\_csv\_no\_border$\$
14415 $\langle$halljanko315.csv$\$
14416 $\langle$distance.2$\$
14417 $\langle$end$\$
14418 $\langle$with\-G\-do$\$
14419 $\langle$graph\-theoretic\_activity$\$
14420 $\langle$find\_cliques\-target\_size.5\-end$\$
14421 $\langle$end
14422
14423
14424
14425 $\langle$graph\-theoretic\_activity\-perform\_activity\Gr\->label=halljanko315\nb\_sol=26208
0
14426
14427
14428 HJ64_cliques5:
14429 $\langle$ORBITER\PATH$\rangle$orbiter.out--v.6\$
14430 $\langle$define\-Gamma\-graph$\$
14431 $\langle$load$\$
14432 $\langle$Group\_Perm315\_Orbital\_3\_colored\_graph$\$
14433 $\langle$end$\$
14434 $\langle$with\-Gamma\-do$\$
14435 $\langle$graph\-theoretic\_activity$\$
14436 $\langle$find\_cliques\-target\_size.5\-end$\$
14437 $\langle$end
14438
14439 $\langle$graph\-theoretic\_activity\-perform\_activity\Gr\->label=Group\_Perm315\_Orbital\_3\_nb\_sol=1008
```
HJ64_cliques5_classify:

```bash
$ (ORBITER_PATH) orbiter.out -v 6
```

-define-Gamma=-graph

-define -load

Group_Perm315_Orbital_3.colored_graph

-end

-define-gens=-vector

-define -load

Group_Perm315_Orbital_3.colored_graph

-end

-define-G-permutation_group

-bsgs-halljanko315:File\halljanko315

-315-1209600:0,1,42,956 gens

-with-G-do

-group_theoretic_activity

-poset_classification_control

-\W

-problem_label-HJ64_cliques

-clique_test-Gamma

-depth 5

-orbits_on_subsets 5

-report

-end

HJ64_cliques_reps_lvl_5

#1-orbit

#ROW,REP,AGO,OL

#0,"0,8,31,110,283",1200,1008

#END
SECTION_COMBINATORIAL_OBJECTS:

Hirschfeld_q4_from_set:

hyperoval_16_create:

EC_read:

PG_3_5_assume_31_read:

LS_AG_2_3_read:
14534  \>  \>  \>  -file_of_designs\`
14535  \>  \>  \>  solutions.csv:9-84-3-12\`
14536  \>  \>  \>-end
14537
14538
14539
14540 geo_7.3_read:
14541  \>  \$\{(ORBITER\_PATH)\}orbiter.out\-v.10\`
14542  \>  \>  -draw_incidence_structure\_description\`
14543  \>  \>  \>  \>-width.60\-with.10\-6\-end\`
14544  \>  \>  \>  -define-C\-combinatorial\_objects\`
14545  \>  \>  \>  \>-file\_of\_incidence\_geometries\`
14546  \>  \>  \>  \>  7.3.\_inc:7.7.21\`
14547  \>  \>  \>-end
14548
14549
14550
14551 Desargues_path_lex_least_read:
14552  \>  \>  echo\$\{(DESARGUES\_PATH\_LEX\_LEAST)\}>Desargues_path_lex_least.\_inc
14553  \>  \$\{(ORBITER\_PATH)\}orbiter.out\-v.10\`
14554  \>  \>  -draw_incidence_structure\_description\`
14555  \>  \>  \>  \>-width.60\-with.10\-6\-end\`
14556  \>  \>  \>  -define-C\-combinatorial\_objects\`
14557  \>  \>  \>  \>  -file\_of\_incidence\_geometries\_by\_row\_ranks\`
14558  \>  \>  \>  \>  \>  Desargues_path_lex_least.\_inc:10:10\-3\`
14559  \>  \>  \>  \>-end
14560
14561
14562
14563 #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14564 #\-Section.11.1:\-File-Formats
14565
14566 geo_pasch_read:
14567  \>  \$\{(ORBITER\_PATH)\}orbiter.out\-v.10\`
14568  \>  \>  -define-C\-combinatorial\_objects\`
14569  \>  \>  \>  -file\_of\_incidence\_geometries\`
14570  \>  \>  \>  \>  pasch.\_inc:6\-4\-12\`
14571  \>  \>  \>-end
14572
14573 geo_pasch_given:
14574  \>  \$\{(ORBITER\_PATH)\}orbiter.out\-v.10\`
14575  \>  \>  -define-C\-combinatorial\_objects\`
14576  \>  \>  \>  -incidence\_geometry\`
14577  \>  \>  \>  \>  "0,1,4,6,8,11,13,14,17,19,22,23"\`
14578  \>  \>  \>  \>  \>  6\-4\-12\`
14579  \>  \>  \>-end
14580
Chapter 14 - Canonical Forms with Nauty

Section 14.1: Overview of Canonical Forms

Section 14.2: Objects in Projective Space

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


```
14628 \> \> \> \> \> -export_group;\\
14629 \> \> \> \> \> -end;\\
14630 \> \> \> \> -end
14631 > pdflatex EC_classification.tex
14632 > open-EC_classification.pdf
14633 > $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix;
14634 > > -input_csv_file-EC_object0_TDA_flag_orbits.csv;
14635 > > -secondary_input_csv_file-EC_object0_TDA.csv;
14636 > > -box_width 20 -bit_depth 24;
14637 > > -end
14638 > open-EC_object0_TDA_flag_orbits_draw.bmp
14639
14640
14641
14642
14643 Hirschfeld_q4_c::Hirschfeld_surface_q4.txt
14644 > $(ORBITER_PATH)orbiter.out -v 6;
14645 > > -define C -combinatorial_objects;\\
14646 > > > -file_of_points-Hirschfeld_surface_q4.txt;\\
14647 > > > -end;\\
14648 > > > -define F -finite_field -q 4 -end;\\
14649 > > > -define P -projective_space 3 F -end;\\
14650 > > > -with C -do;\\
14651 > > > -combinatorial_object_activity;\\
14652 > > > > -canonical_form PG P;\\
14653 > > > > > -classification_prefix-Hirschfeld_surface_q4;\\
14654 > > > > > -save_ago;\\
14655 > > > > > -max_TDO_depth 10;\\
14656 > > > > > -end;\\
14657 > > > > > -report;\\
14658 > > > > > -prefix Hirschfeld_surface_q4;\\
14659 > > > > > -export_flag_orbits;\\
14660 > > > > > -show_TDO;\\
14661 > > > > > -show_TDA;\\
14662 > > > > > -dont_show_incidence_matrices;\\
14663 > > > > > -export_group;\\
14664 > > > > > -end;\\
14665 > > > > -end
14666 > pdflatex Hirschfeld_surface_q4_classification.tex
14667 > open-Hirschfeld_surface_q4_classification.pdf
14668
14669
14670 # group-order-is-51840
14671
14672
14673 Hirschfeld_q4_set_c:
14674 > $(ORBITER_PATH)orbiter.out -v 4;\\
```

748
14722 \openDickson_sets_classification.pdf
14723
14724
14725
14726 Endrass_7c::Endrass_F7.txt
14727  $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 2\$
14728  \define C\ -\text{combinatorial\_objects}\$
14729  \define_file_of_points Endrass_F7.txt$
14730  \define -end$
14731  \define F\ -\text{finite\_field}\ -q\ 7\ -\text{end}\$
14732  \define P\ -\text{projective\_space}\ -3\ F\ -\text{end}\$
14733  \with C\ -\text{do}\$
14734  \define combinatorial\_object\_activity$
14735  \define canonical\_form PG P$
14736  \define classification\_prefix Endrass_F7$
14737  \define save\_ago$
14738  \define -end$
14739  \define report$
14740  \define -end
14741  \pdflatex\ Endrass_F7.classification.tex
14742  \open Endrass_F7.classification.pdf
14743
14744
14745 \# \text{group\_order\_is\_32}
14746
14747
14748 hyperoval_16\_c:
14749  $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 2\$
14750  \define C\ -\text{combinatorial\_objects}\$
14751  \define set_of_points $(\text{HYPEROVAL\_16\_16320})$
14752  \define set_of_points $(\text{HYPEROVAL\_16\_144})$
14753  \define -end$
14754  \define F\ -\text{finite\_field}\ -q\ 16\ -\text{end}\$
14755  \define F\ -\text{projective\_space}\ -2\ F\ -\text{end}\$
14756  \define -with C\ -\text{do}\$
14757  \define combinatorial\_object\_activity$
14758  \define canonical\_form PG P$
14759  \define classification\_prefix hyperoval_q16$
14760  \define label hyperoval_q16$
14761  \define save\_ago$
14762  \define save\_transversal$
14763  \define max_TDO\_depth\_10$
14764  \define -end$
14765  \define report$
14766  \define -prefix hyperoval_q16$
14767  \define -export\_flag\_orbits$
14768  \define -show TDO$

750
14769 ▷ ▷ ▷ -show_TDA\ 
14770 ▷ ▷ ▷ -dont_show_incidence_matrices\ 
14771 ▷ ▷ ▷ -export_group\ 
14772 ▷ ▷ -end\ 
14773 ▷ -end\ 
14774 ▷ pdflatex hyperoval_q16_classification.tex\ 
14775 ▷ open-hyperoval_q16_classification.pdf\ 
14776 ▷ $(ORBITER_PATH)/orbiter.out-v.2-draw_matrix\ 
14777 ▷ -input_csv_file-hyperoval_q16_object0_TDA_flag_orbits.csv\ 
14778 ▷ -secondary_input_csv_file-hyperoval_q16_object0_TDA.csv\ 
14779 ▷ -box_width=4-bit_depth=24\ 
14780 ▷ -end\ 
14781 ▷ open-hyperoval_q16_object0_TDA_flag_orbits_draw.bmp\ 
14782 ▷ $(ORBITER_PATH)/orbiter.out-v.2-draw_matrix\ 
14783 ▷ -input_csv_file-hyperoval_q16_object1_TDA_flag_orbits.csv\ 
14784 ▷ -secondary_input_csv_file-hyperoval_q16_object1_TDA.csv\ 
14785 ▷ -box_width=4-bit_depth=24\ 
14786 ▷ -end\ 
14787 ▷ open-hyperoval_q16_object1_TDA_flag_orbits_draw.bmp\ 
14788 \ 
14789 \ 
14790 \ 
14791 \ 
14792 \ 
14793 cubic_curves_PG_2.8.canon:\ 
14794 ▷ $(ORBITER_PATH)/orbiter.out-v.6\ 
14795 ▷ -define-C=combinatorial_objects\ 
14796 ▷ -set_of_points=2,3,28,46,51,61,40,71\ 
14797 ▷ -end\ 
14798 ▷ -define-F=finite_field-q=8\ 
14799 ▷ -define-P=projective_space=2-F\ 
14800 ▷ -with-C-do\ 
14801 ▷ -combinatorial_object_activity\ 
14802 ▷ -canonical_form_PG_P\ 
14803 ▷ -classification_prefix-cc_8\ 
14804 ▷ -save_ago\ 
14805 ▷ -max_TDO_depth=10\ 
14806 ▷ -end\ 
14807 ▷ -report\ 
14808 ▷ -end\ 
14809 ▷ pdflatex cc_8_classification.tex\ 
14810 ▷ open-cc_8_classification.pdf\ 
14811 \ 
14812 \ 
14813 F_alpha_beta_gamma_delta_classify_q7_nauty:F_alpha_beta_gamma_delta_q7_points.txt\ 
14814 ▷ $(ORBITER_PATH)/orbiter.out-v.6\ 

751
-define C -combinatorial_objects:\n-define F -finite_field -q 7 -end\n-define P -projective_space 3 F -end\n-with C -do\n-combinatorial_object_activity\n-canonical_form PG P\n-classification_prefix surface 15 lines q7\n-save ago\n-save_transversal\n-end\n
-end
#pdflatex surface 15 lines q7 classification.tex
#open surface 15 lines q7 classification.pdf

#User.time: 4:12 on Mac
#6-orbits

ovoid q8_canon: ovoid q8.txt
$ (ORBITER_PATH) orbiter.out -v 6\n-define C -combinatorial_objects\n-file_of_points ovoid q8.txt\n-end\n-define F -finite_field -q 8 -end\n-define P -projective_space 3 F -end\n-with C -do\n-combinatorial_object_activity\n-canonical_form PG P\n-classification_prefix ovoid\n-label ovoid\n-save ago\n-max TDO depth 4\n-end\n-report\n-prefix ovoid\n-show TDO\n-show TDA\n-dont show incidence matrices\n-export group\n-end\n
ovoid_ST_q8_canon::ovoid_ST_q8.txt
$(ORBITER_PATH)orbiter.out--v.6
-define C -combinatorial_objects
-file_of_points-ovoid_ST_q8.txt
-define F -finite_field -q 8
-define P -projective_space -3 F
-with C -do
-combinatorial_object_activity
-canonical_form_PG P
-classification_prefix-ovoid_ST
-label-ovoid_ST
-save ago
-max_TDO_depth 4
-end
-report
-prefix-ovoid_ST
-show_TDO
-show_TDA
-dont_show_incidence_matrices
-export_group
-report
-prefix-ovoid_ST
-show_TDO
-show_TDA
-dont_show_incidence_matrices
-export_group
-end

group order 87360 = 3 * 3 * 29120
SUZUKI_8_GENERATORS="
1,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,"
1,0,0,0,6,0,0,0,0,2,0,0,0,0,3,0,"
1,0,0,0,1,1,1,0,0,0,1,0,1,0,0,1,0,"
1,0,0,0,3,6,2,2,5,0,2,0,3,0,6,3,2,"
Suzuki_8:

\begin{verbatim}
$(ORBITER_PATH)orbiter.out\ -v.6\ \\
define F -fine_field -q 8 -end \ 
define gens -vector -field F \ 
compact $(SUZUKI_8_GENERATORS) -end \ 
define G -linear_group -PGGL 4:8 \ 
subgroup_by_generators "Sz8" "87360" -gens \ 
with G -do \ 
-define $\cdot F -finite_field -q 8 -end \ 
-define gens -vector -field F \ 
compact $(SUZUKI_8_GENERATORS) -end \ 
-define G -linear_group -PGGL 4:8 \ 
subgroup_by_generators "Sz8" "87360" -gens \ 
with G -do \ 
-group_theoretic_activity \ 
end \ 
pdflatex PGGL_4_8_Subgroup_Sz8_87360_report.tex \ 
open PGGL_4_8_Subgroup_Sz8_87360_report.pdf \ 
end
\end{verbatim}

#### Section 14.3: Incidence Geometries

SECTION INCIDENCE_GEOMETRIES:

geo_7_3_c:

\begin{verbatim}
$(ORBITER_PATH)orbiter.out\ -v.10\ \\
-draw incidence structure description \ 
-width 60 -with 10 6 -end \ 
define C -combinatorial_objects \ 
-file_of incidence geometries 7_3.inc 7 7 21 \ 
-end \ 
-with C -do \ 
-combinatorial_object_activity \ 
-canoncal_form \ 
-classification_prefix 7_3 \ 
-label 7_3 \ 
-save ago \ 
-save_transversal \ 
-end \ 
-report \ 
-prefix 7_3 \ 
-export_flag_orbits \ 
-show incidence_matrices \ 
\end{verbatim}

754
geo_10_3_c_lex_least:
$\text{(ORBITER PATH)}$orbiter.out\-v\-10\.
$\text{-draw incidence structure description}\$
$\text{-width 60}\text{-with 10 6}\text{-end}\$
$\text{-define Test lines}\text{-set -loop 4 11 1}\text{-end}\$
$\text{-define Geo}\text{-geometry builder}\$
$\text{-V 10}\text{-B 10}\text{-TDO 3}\text{-fuse 1}\$
$\text{-fname GEO 10_3}\$
$\text{-test Test lines}\$
$\text{-end}\$
$\text{-define C}\text{-combinatorial objects}\$
$\text{-file of incidence geometries 10_3.inc 10 10 30}\$
$\text{-end}\$
$\text{-with C -do}\$
$\text{-combinatorial object activity}\$
$\text{-canonical form}\$
$\text{-classification prefix 10_3}\$
$\text{-label 10 3}\$
$\text{-save ago}\$
$\text{-save transversal}\$
$\text{-end}\$
$\text{-report}\$
$\text{-prefix 10_3}\$
$\text{-export flag orbits}\$
$\text{-show incidence matrices}\$
$\text{-export group}\$
$\text{-show TDO}\$
$\text{-show TDA}\$
$\text{-lex least Geo}\$
$\text{-end}\$
$\text{-end}\$
pdflatex\-10_3_classification.tex
$\text{(ORBITER PATH)}$orbiter.out\-v\-2\-draw matrix\$
$\text{-input csv file 10_3 object7 INP flag orbits.csv}\$
$\text{-box width 16}\text{-bit depth 24}\$
$\text{-end}\$

756
15050 #> -secondary_input_csv_file:10_3_object7_TDA.csv
15051 #> -box_width:16-bit_depth:24
15052 #> -end
15053 $(ORBITER_PATH)orbiner.out-v:2-draw_matrix-
15054 #> -input_csv_file:10_3_object7_INP_flag_orbits.csv
15055 #> -secondary_input_csv_file:10_3_object7_INP.csv
15056 #> -box_width:16-bit_depth:24
15057 #> -end
15058
15059 #10_3_object7_TDA_flag_orbits.csv
15060
15062
15063
15064
15065 geo_14_3.c:
15066 #> $(ORBITER_PATH)orbiner.out-v:2-
15067 #> -draw_incidence_structure_description-
15068 #> -width:60-with:10:6:-end-
15069 #> -define:Test_lines-set:loop:15:1:-end-
15070 #> -define:C=--combinatorial_objects-
15071 #> -file_of_incidence_geometries:14_3.inc:14:14:42-
15072 #> -end-
15073 #> -with:C=do-
15074 #> -combinatorial_object_activity-
15075 #> -canonical_form-
15076 #> -classification_prefix:14_3-
15077 #> -label:14_3-
15078 #> -save:ago-
15079 #> -save:transversal-
15080 #> -end-
15081 #> -end
15082
15083
15084 #> -report-
15085 #> -prefix:14_3-
15086 #> -export_flag_orbits-
15087 #> -show_incidence_matrices-
15088 #> -export_group-
15089 #> -end-
15090
15091
15092 geo_15_3.c:
15093 #> $(ORBITER_PATH)orbiner.out-v:2-
15094 #> -draw_incidence_structure_description-
15095 #> -width:50-with:10:5:-end-
15096 #> -define:C=--combinatorial_objects-
-define C -combinatorial_objects:
-define C -combinatorial_objects:
-file_of_incidence_geometries 40_4_g4.inc 40_40_160:
-end:
-with C -do:
-combinatorial_object_activity:
-canononical_form:
-classification_prefix 40_g4:
-label 40_g4:
-save ago:
-end:
-report:
-prefix 40_4_g4:
-export_flag_orbits:
-show_TDO:
-show_TDA:
-show_incidence_matrices:
-end:
-end
pdflatex 40_4_g4_classification.tex
open 40_4_g4_classification.pdf

geo_17_3_g4.c:
$(ORBITER_PATH)orbiter.out -v 2:
-draw incidence_structure_description:
-width 50 -with 10.5 -end:
-with C -do:
-combinatorial_object_activity:
-canononical_form:
-classification_prefix 17_3_g4:
-label 17_3_g4:
-save ago:
-end:
-report:
-prefix 17_3_g4:
-export_flag_orbits:
-show_TDO:
-show_TDA:
-show_incidence_matrices:
-end:
-end
pdflatex 17_3_g4_classification.tex
open 17_3_g4_classification.pdf

759
AG_2_3.c: AG_2_3.inc
$(ORBITER_PATH) orbiter.out -v:2 \n-define C - combinatorial_objects \n-file_of_incidence_geometries \nAG_2_3.inc-9.12:36 \n-end \n-with C -do \n-combinatorial_object_activity \n-canonical_form \n-classification_prefix LSQ6 \n-label LSQ6 \n-save ago \n-max_TDO_depth:10 \n-end \n-prefix AG_2_3 \n-export_flag_orbits \n-show_TDO \n-show_TDA \n-show_incidence_matrices \n-end \n-report \n-prefix AG_2_3 \n-export_flag_orbits \n-show_TDO \n-show_TDA \n-show_incidence_matrices \n-end \n-pdflatex AG_2_3_classification.tex
open AG_2_3_classification.pdf
$(ORBITER_PATH) orbiter.out -v:2 -draw_matrix \n-input_csv_file AG_2_3_object0_INP_flag_orbits.csv \n-secondary_input_csv_file AG_2_3_object0_INP.csv \n-box_width:40 -bit_depth:24 \n-end \n-open AG_2_3_object0_INP_flag_orbits_draw.bmp
geo_LSQ6.c:
$(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-LSQ6.inc:18:39:126 \n-end \n-with C -do \n-combinatorial_object_activity \n-canonical_form \n-classification_prefix LSQ6 \n-label LSQ6
15238 \gg \gg \gg -save_age\backslash
15239 \gg \gg \gg -save_transversal\backslash
15240 \gg \gg \gg -end\backslash
15241 \gg \gg \gg -report\backslash
15242 \gg \gg \gg -prefix:LSQ6\backslash
15243 \gg \gg \gg -export_flag_orbits\backslash
15244 \gg \gg \gg -show_incidence_matrices\backslash
15245 \gg \gg \gg -export_group\backslash
15246 \gg \gg \gg -end\backslash
15247 \gg \gg -end
15248 \gg pdflatex:LSQ6_classification.tex
15249 \gg #open:LSQ6_classification.pdf
15250 \gg $(\textsc{ORBITER\_PATH})\texttt{orbit}er.out-v.2-draw_matrix\backslash
15251 \gg \gg -input.csv_file:LSQ6.object0.TDA.flag_orbits.csv\backslash
15252 \gg \gg -secondary_input.csv_file:LSQ6.object0.TDA.flag_orbits.csv\backslash
15253 \gg \gg -box_width-32-bit_depth-24\backslash
15254 \gg \gg -end
15255 \gg $(\textsc{ORBITER\_PATH})\texttt{orbit}er.out-v.2-draw_matrix\backslash
15256 \gg \gg -input.csv_file:LSQ6.object0.INP.flag_orbits.csv\backslash
15257 \gg \gg -secondary_input.csv_file:LSQ6.object0.INP.flag_orbits.csv\backslash
15258 \gg \gg -box_width-32-bit_depth-24\backslash
15259 \gg \gg -end
15260 \gg open:LSQ6.object0.INP.flag_orbits_draw.bmp
15261
15262
15263
15264 \# ToDo:
15265
15266 \texttt{quartic\_curve\_25\_0\_0\_canonical:}\backslash
15267 \gg $(\textsc{ORBITER\_PATH})\texttt{orbit}er.out-v.3\backslash
15268 \gg \gg -define:F\_finite_field\_q\_25\_end\backslash
15269 \gg \gg -define:P\_projective_space\_2\_F\_end\backslash
15270 \gg \gg -with:P\_do\backslash
15271 \gg \gg -projective\_space\_activity\backslash
15272 \gg \gg -canonical\_form\_PG\backslash
15273 \gg \gg -input\backslash
15274 \gg \gg -set_of_points:"10,11,59,63,124,135,136,170,206,257,275,284,285,367,378,393,433,619,641,644."ackslash

761
-set_of_points:"2,12,48,65,87,120,189,246,305,323,354,375,434,435,455,482,496,557,586,595"

15291 #Section 14.4: Objects from Design Theory

15295 SECTION_OBJECTS_FROMDESIGN_THEORY:

15298 LS_AG_2_3_solutions_classify:
15300 $(ORBITER_PATH)orbiert.out\-v.2\-
15301 \-draw_incidence_structure_description\-
15302 \-width.20\-width.10.2\-end\-
15303 \-define-C\-combinatorial_objects\-
15304 \-file_of_designs\-
15305 \-solutions.csv.9.84.3.12\-
15306 \-end\-
15307 \-with-C\-do\-
15308 \-combinatorial_object_activity\-
15309 \-canonical_form\-
15310 \-save Ago\-
15311 \-save_transversal\-
15312 \-classification_prefix-LS_AG_2_3\-
15313 \-label-LS_AG_2_3\-
15314 \-max_TDO_depth.10\-
15315 \-end\-
15316 \-report\-
15317 \-prefix-LS_AG_2_3\-
15318 \-export_flag_orbits\-
15319 \-show_TDO\-
15320 \-end\-
15321 \-end
15322 pdflatex-LS_AG_2_3_classification.tex
15323 open-LS_AG_2_3_classification.pdf
design_27c:

```
\$\$(\text{ORBITER\_PATH})\text{orbiter.out\,-v\,-2\,-draw\_matrix}\$
\$\$-\text{input\_csv\_file}\text{\text{LS\_AG}\_2\_3\_object0\_INP\_flag\_orbits.csv}\$
\$\$-\text{secondary\_input\_csv\_file}\text{\text{LS\_AG}\_2\_3\_object0\_INP.csv}\$
\$\$-\text{box\_width}12\,-\text{bit\_depth}24$\$
\$\$\text{open-LS\_AG\_2\_3\_object0\_INP\_flag\_orbits\_draw.bmp}$
\$\$-\text{define-C-\text{-combinatorial\_objects}}$
\$\$-\text{set\_of\_points}2,56,30,112,253,90,440,508$
\$\$\text{-end}$
\$\$-\text{define-F-finite\_field\,-q\,-27\,-override\_polynomial\,-46}$
\$\$\text{-with-C-do}$
\$\$\text{-combinatorial\_object\_activity}$
\$\$\text{-canonical\_form\_PG\,-P}$
\$\$\text{-classification\_prefix\_design}$
\$\$\text{-end}$
\$\$\text{pdflatex\,design\_classification.tex}$
\$\$\text{open-design\_classification.pdf}$
```

# Section 14.5: Linear Codes

SECTION CANONICAL FORMS OF LINEAR CODES:
code_3_2_aut:

$\text{(ORBITER\_PATH)\ orbiter.out.-v.20}\$

$\text{-define\_F\_finite\_field\_q.2\_end}\$

$\text{-define\_genma\_vector\_field\_F\_format.2}\$

$\text{-dense\_$(CODE_N3.K2.Q2.GENMA)\$}\$

$\text{-end}\$

$\text{-define\_P\_projective\_space.1\_F\_end}\$

$\text{-with\_P\_do}\$

$\text{-projective\_space\_activity}\$

$\text{-canonical\_form\_of\_code}\$

$\text{-"3_2\"\_genma\_save\_ago\_label\_"3_2\"}\$

$\text{-classification\_prefix\_"3_2\"}\$

$\text{-end}\$

$\text{-box\_width.16\_bit\_depth.24}\$

$\text{pdflatex\_3_2\_classification.tex}$

$\text{open\_3_2\_classification.pdf}$

$\text{\$(ORBITER\_PATH)\ orbiter.out.-v.2\_draw\_matrix\$}$

$\text{-input\_csv\_file.3_2\_object0\_TDA\_flag\_orbits.csv}\$

$\text{-secondary\_input\_csv\_file.3_2\_object0\_TDA.csv}\$

$\text{-box\_width.16\_bit\_depth.24}\$

$\text{open\_3_2\_object0\_TDA\_flag\_orbits\_draw.bmp}$

$\text{code_6_3_aut:}$

$\text{\$(ORBITER\_PATH)\ orbiter.out.-v.20\$}$

$\text{-define\_F\_finite\_field\_q.2\_end}\$

$\text{-define\_genma\_vector\_field\_F\_format.3}\$

$\text{-compact\_$(CODE_N6.K3.Q2.GENMA)\$}\$

$\text{-end}\$

$\text{-define\_P\_projective\_space.2\_F\_end}\$

$\text{-with\_P\_do}\$

$\text{-projective\_space\_activity}\$

$\text{-canonical\_form\_of\_code}\$

$\text{-"6_3\"\_genma\_save\_ago\_label\_"6_3\"}\$

$\text{-classification\_prefix\_"6_3\"}\$

$\text{-end}\$

$\text{-end}\$

$\text{pdflatex\_6_3\_classification.tex}$

$\text{open\_6_3\_classification.pdf}$

$\text{\$(ORBITER\_PATH)\ orbiter.out.-v.2\_draw\_matrix\$}$

$\text{-input\_csv\_file.6_3\_object0\_TDA\_flag\_orbits.csv}\$

$\text{-secondary\_input\_csv\_file.6_3\_object0\_TDA.csv}\$

$\text{-box\_width.16\_bit\_depth.24}\$
RM_3_1_group:
$\text{(ORBITER_PATH)orbiter.out}\ -v\ 2$
\begin{verbatim}
\define F = finite_field -q 2 -end
\define genma = vector -field F -format 4 -compact $(CODE_RM_3_1_GENMA)
\define P = projective_space -3F -end
\with P -do
\projective_space_activity
\canonical_form_of_code
"RM_3_1" genma -save_ago -label "RM_3_1"
\classification_prefix "RM_3_1"
\end
\end
\define P = projective_space -3F -end
\end
\define P = projective_space -3F -end
\end
pdflatex RM_3_1_classification.tex
open RM_3_1_classification.pdf
# group of order 1344
RM_3_1_object0_INP_flag_orbits.csv
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_INP_flag_orbits.csv
\input csv_file RM_3_1_object0_INP.csv
box_width 16 -bit_depth 24
- end
$(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\ 
> -end\ 
open:RM_3.1_object0_INP_flag_orbits_draw.bmp\ 
open:RM_3.1_object0_TDA_flag_orbits_draw.bmp\ 
RM_4.1_group:\ 
$(ORBITER PATH)orbiter.out -v.2\ 
> -define_F:finite_field_q:2 -end\ 
> -define_genma:vector:field:F -format:5\ 
> -compact $(CODE_RM_4.1_GENMA)\ 
> -end\ 
> -define_P:projective_space:4 -F -end\ 
> -with_P:do\ 
> -projective_space_activity\ 
> -canonical_form_of_code\ 
> -RM_4.1:genma:save_ago: -label:"RM_4.1"\ 
> -classification_prefix:"RM_4.1"\ 
> -end\ 
> -end\ 
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\ 
$(ORBITER PATH)orbiter.out -v.2 -draw_matrix\ 
> -input_csv_file:RM_4.1_object0_TDA_flag_orbits.csv\ 
> -secondary_input_csv_file:RM_4.1_object0_TDA.csv\ 
> -box_width:16 -bit_depth:24\ 
> -end\ 
open:RM_4.1_object0_INP_flag_orbits_draw.bmp\ 
open:RM_4.1_object0_TDA_flag_orbits_draw.bmp\ 
# group-order: 322560 = 24*30*28*16\ 
 RS_6.4.7_group:\ 
$(ORBITER PATH)orbiter.out -v.20\ 
> -define_F:finite_field_q:7 -end\ 
> -define_genma:vector:field:F -format:4\ 

code_n15_k6_d6_a_group:
\$\text{ORBITER\_PATH}\text{\_out} -v 20\$
\begin{verbatim}
-define F -finite_field -q 2 -end
-define genma -vector -field F -format 6
-compact $(CODE_GV_N15_K6)
-define P -projective_space -F -end
-define P -do
-define P -end
-define P -projective_space -5 -F -end
-define P -do
-define P -end
-define P -do
-define P -end
\end{verbatim}

pdflatex GV_n15_k6_d5_classification.tex

open GV_n15_k6_d5_classification.pdf

#ago=12
Hamming_graph_7_with_Hamming_code:

Hamming_graph_7_with_Hamming_code:

Hamming_graph_7_with_Hamming_code:

Hamming_graph_7_with_Hamming_code:

Hamming_graph_7_with_Hamming_code:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:

SECTION_CANONICAL_FORMS_OF_GENERAL_CODES:
# Section 14.7: Graphs

SECTION_CANONICAL_FORMS_OF_GRAPHS:

JK_graph_pp16_1:

JK_graph_pp16_1: $(ORBITER\_PATH)orbiter.out -v:2\n
JK_graph_pp16_1: -define-Gamma -graph -load_dimacs\n
JK_graph_pp16_1: ../JUNTTILA_KASKI/benchmarks/pp/pp16-1\n
JK_graph_pp16_1: -end\n
JK_graph_pp16_1: -with-Gamma -do\n
JK_graph_pp16_1: -graph_theoretic_activity -automorphism_group\n
JK_graph_pp16_1: -save -end\n
JK_graph_pp16_1: -with-Gamma -do\n
JK_graph_pp16_1: -graph_theoretic_activity -automorphism_group -end\n
769
15653
15654  # go=34217164800
15655  #nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling::nb_backtrack1==6
15656  #nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling::nb_backtrack2==134
15657  JK_graph_pp16_2:
15658  >> $(ORBITER_PATH)orbiter.out-v.2\n15659  >>  -define-Gamma=graph-load_dimacs\n15660  >>  ../JUNTTILA_KASKI/benchmarks/pp/pp16-2\n15661  >>  -end\n15662  >>  -with-Gamma-do\n15663  >>  -graph_theoretic_activity-save-end\n15664  >>  -with-Gamma-do\n15665  >>  -graph_theoretic_activity-automorphism_group-end\n15666  # does not finish
15667  JK_graph_pp16_9:
15668  >> $(ORBITER_PATH)orbiter.out-v.2\n15669  >>  -define-Gamma=graph-load_dimacs\n15670  >>  ../JUNTTILA_KASKI/benchmarks/pp/pp16-9\n15671  >>  -end\n15672  >>  -with-Gamma-do\n15673  >>  -graph_theoretic_activity-save-end\n15674  >>  -with-Gamma-do\n15675  >>  -graph_theoretic_activity-automorphism_group-end\n15676  JK_graph_grid_3_3:
15677  >> $(ORBITER_PATH)orbiter.out-v.2\n15678  >>  -define-Gamma=graph-load_dimacs\n15679  >>  ../JUNTTILA_KASKI/benchmarks/grid/grid-w-3-3\n15680  >>  -end\n15681  >>  -with-Gamma-do\n15682  >>  -graph_theoretic_activity-save-end\n15683  >>  -with-Gamma-do\n15684  >>  -graph_theoretic_activity-automorphism_group-end\n15685  # Written file grid-w-3-3_group.makefile of size 579
15686  # User time: 0 of a second, dt=0 tps=100
# nb_calls_to_densenaughty=1

```
JK_graph_sts_13:
15700  > $(ORBITER_PATH)orbiter.out -v 2 \\
15701  > -define-Gamma -graph -load_dimacs \\
15702  > ../JUNTTILA_KASKI/benchmarks/srg/sts-13 \\
15703  > -end \\
15704  > -with-Gamma -do \\
15705  > -graph_theoretic_activity -save -end \\
15706  > -with-Gamma -do \\
15707  > -graph_theoretic_activity -automorphism_group -end \\
15708  > make ORBITER_PATH=$(ORBITER_PATH) -f sts-13_group.makefile sts-13
15709
15710
15711 # nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack1 = 3
15712 # nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack2 = 24
15713
15715 HJ_aut:
15716  > $(ORBITER_PATH)orbiter.out -v 6 \\
15717  > -define-G -graph \\
15718  > -load_csv_no_border \\
15719  > halljanko315.csv \\
15720  > -end \\
15721  > -with-G -do \\
15722  > -graph_theoretic_activity -automorphism_group \\
15723  > -end \\
15724  > -with-G -do \\
15725  > -graph_theoretic_activity -properties \\
15726  > -end \\
15727
15728
15729
15730 HJ_group_and_orbits:
15731  > $(ORBITER_PATH)orbiter.out -v 2 \\
15732  > -define-gens-vector-file \\
15733  > halljanko315.gens.csv -end \\
15734  > -define-G -permutation_group \\
15735  > -bsgs halljanko315 "File\ halljanko315" \\
15736  > 315 1209600 "0 1 2" 6 gens \\
15737  > -end \\
15738  > -with-G -do \\
15739  > -group_theoretic_activity \\
15740  > -poset_classification_control \\
```
HJ orbital graph 3:

$\text{define-gens-vector-file}$

$\text{define-G-permutation_group}$

$\text{bsgs-halljanko315-"File\\_halljanko315"}$

$315\cdot1209600\cdot"0,1,2"\cdot6\cdot\text{gens}$

$\text{define-Gamma-graph}$

$\text{orbital_graph_G-3}$

$\text{with-Gamma-do}$

$\text{graph_theoretic_activity}$

$\text{properties}$

$\text{with-Gamma-do}$

$\text{graph_theoretic_activity}$

$\text{save}$

$\text{end}$

$\text{GroupPerm315 Orbital_3 colored}$

$\text{Degree.type: (64-\{315\})}$
Section 14.8: Quartic Curves

SECTION_CANONICAL_FORMS_OF_QUARTIC_CURVES:

F_17_edge:
$($(ORBITER_PATH)orbiter.out-v.3\$)
\%define F - finite field - q.17 - end\%
\%with F - do - finite field activity\%
\%cheat sheet GF\%
pdflatex GF_17.tex
open GF_17.pdf

Edge_curve_17_nauty:
$($(ORBITER_PATH)orbiter.out-v.3\$)
\%define C - combinatorial objects\%
\%file of points Edge_q17.txt\%
\%end\%
\%define F - finite field - q.17 - end\%
\%define P - projective space - 2 F\%

15835 ▶ ▶  -with-C-do-
15836 ▶ ▶  -combinatorial_object_activity-
15837 ▶ ▶ ▶ -canonical_form_PG-P-
15838 ▶ ▶ ▶ ▶ -classification_prefix:Edge_curve_q17-
15839 ▶ ▶ ▶ ▶ -label:Edge_curve_q17-
15840 ▶ ▶ ▶ ▶ -save_ago-
15841 ▶ ▶ ▶ ▶ -save_transversal-
15842 ▶ ▶ ▶ ▶ -max_TDO_depth:10-
15843 ▶ ▶ ▶ ▶ -end-
15844 ▶ ▶ ▶ -report-
15845 ▶ ▶ ▶ ▶ -prefix:Edge_curve_q17-
15846 ▶ ▶ ▶ ▶ -export_flag:orbits-
15847 ▶ ▶ ▶ ▶ -show_TDO-
15848 ▶ ▶ ▶ ▶ -show_TDA-
15849 ▶ ▶ ▶ ▶ -dont_show_incidence_matrices-
15850 ▶ ▶ ▶ ▶ -export_group-
15851 ▶ ▶ ▶ -end-
15852 ▶ ▶ -end
15853 ▶ pdflatex:Edge_curve_q17_classification.tex
15854 ▶ open-Edge_curve_q17_classification.pdf
15855 ▶ $(ORBITER_PATH):orbiter.out-\v:2-draw_matrix-
15856 ▶ ▶ -input_csv_file:Edge_curve_q17_object0_TDA_flag_orbits.csv-
15857 ▶ ▶ -secondary_input_csv_file:Edge_curve_q17_object0_TDA.csv-
15858 ▶ ▶ -box_width:4-bit_depth:24-
15859 ▶ ▶ -end
15860 ▶ open-Edge_curve_q17_object0_TDA_flag_orbits_draw.bmp
15861
15862 #9-backtrack-nodes-total
15863
15864
15865 #aut.=:24
15866 #User-time:.04-of-a-second, dt=4-tps=.100
15867
15868
15869 #generators-for-a-group-of-order:24:
15870 #1,0,0,0,13,0,0,0,4,-
15871 #1,0,0,0,16,0,16,0,-
15872 #0,1,16,2,4,4,15,4,4,-
15873
15874
15875 Edge_curve_17_group:
15876 ▶ $:(ORBITER_PATH):orbiter.out-\v:3-
15877 ▶ ▶ -define-G-linear_group-PGL:3:17-
15878 ▶ ▶ -subgroup_by_generators:"Stab_Edge"."24":3-
15879 ▶ ▶ ▶ "1,0,0,0,13,0,0,0,4-
15880 ▶ ▶ ▶ "1,0,0,0,0,16,0,16,0-
15881 ▶ ▶ ▶ "0,1,16,2,4,4,15,4,4-

774
SECTION GRAPHICAL OUTPUT:

F_7.tables:

\$\text{ORBITER_PATH}/\text{orbiter.out} -v 3 \$

-define F -finite_field -q 7 -end

-with F -do -finite_field_activity

-cheat_sheet_GF

-end

\$\text{ORBITER_PATH}/\text{orbiter.out} -v 2 \$

-draw_matrix

-input_csv_file GF_q7_addition_table.csv

-box_width 40

-bit_depth 24

-partition 3 7 7

-end

open GF q7 addition_table_draw.bmp

PG_2.4_cyclic_incma:

\$\text{ORBITER_PATH}/\text{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\-
1 - 1.0,1,0,0,1,2,1,1,0:"PG_2.4_singer".\-
1 - -end
1 $(ORBITER_PATH)orbiter.out-v.4\-
1 - -list_arguments\-
1 - -define-R-vector-repea\-
1 - -define-C-vector-repea\-
1 - -draw_matrix\-
1 - -input_csv_file-PG_2.4_singer_incma_cyclic.csv\-
1 - -box_width-40-\-bit_depth-24\-
1 - -partition-3-R:C\-
1 - -end
1 open-PG_2.4_singer_incma_cyclic_draw.bmp
1 1
1 1
1 PGL_4.2_Wedge_4.0_graphical_output:\-
1 $(ORBITER_PATH)orbiter.out-v.4\-
1 - -define-G-linear_group-PGL_4.2\-
1 - -wedge_detached\-
1 - -end\-
1 - -with-G-do\-
1 - -group_theoretic_activity\-
1 - -report\-
1 - -end
1 pdflatex-PGL_4.2_Wedge_4.2_detached_report.tex
1 1
1 open-PGL_4.2_Wedge_4.2_detached_report.pdf
1 1
1 # - -draw_options:-radius-200-\-end\-
1 1
1 schreier_tree_graphical_output:\-
1 $(ORBITER_PATH)orbiter.out-v.4\-
1 - -draw_options\-
1 - -yout-500000\-
1 - -radius-15-\-nodes_empty\-
1 - -line_width-0.5-\-y_stretch-0.25\-
1 - -end\-
1 - -define-G-linear_group-PGL_4.2-end\-
1 - -with-G-do\-
1 - -group_theoretic_activity\-
1 - -orbits_on_polynomials-3\-
1 - -orbits_on_polynomials_draw_tree-6\-
1 - -end\-
1 pdflatex-poly_orbits_d3_n3_q2.tex
1 open-poly_orbits_d3_n3_q2.pdf
1 1
1 776
15976 Queens_graph:
15977 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15978 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15979 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15980 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15981 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15982 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15983 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15984 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15985 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15986 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15987 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15988 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15989 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15990 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15991 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15992 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15993 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15994 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15995 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15996 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15997 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15998 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
15999 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16000 SECTION_POVRAY:
16001 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16002 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16003 cube:
16004 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16005 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16006 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16007 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16008 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16009 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16010 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16011 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16012 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16013 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16014 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16015 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16016 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16017 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16018 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16019 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16020 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16021 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
16022 $\langle ORBITER\_PATH\rangle orbiter.out -v.2:\$
monkey:

$\{(\text{ORBITER\_PATH})\text{orbiter.out}\ -v.2\ -povray\}$

- round \ -nb\_frames\_default\-30\$
- output\_mask\_monkey\_\%d\_\%03d.pov$
- video\_options\-W1024\-H768$
- global\_picture\_scale\-0.8$
- 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\$\{(\text{MONKEY\_SADDLE\_CUBIC})\}$$
- plane\_by\_dual\_coordinates\-0,0,1,0$
- group\_of\_things\-0$
- group\_of\_things\-0$
- cubs\-0\"\text{texture}\{\text{pigment}\-\text{Gold}\}\text{finish}\$
- ambient\-0.4\-diffuse\-0.5\-roughness\-0.001$
- reflection\-0.1\-specular\-0.8$\}$
- planes\-1\"\text{texture}\{\text{pigment}\-\text{Blue}\$
- transmit\-0.5\}\text{finish}\{\text{diffuse\-0.9\-phong\-0.2}\}$
- scene\_objects\_end$
- povray\_end$
- povray\_end
mkdir
·
POV

mv · monkey · *.pov · POV

mv · makefile · animation · POV

Eckardt:
$(ORBITER_PATH)orbiter.out -v:2 -povray \n-round:0 -nb_frames_default:30 \n-output_mask:Eckardt\%d\%03d.pov \n-video_options:-W:1024 -H:768 \n-global_picture.scale:0.9 \n-default_angle:75 \n-clipping_radius:2.4 \n-camera:0 -1,1,1 -3,1,3 -0.12,0.12,0.12 \n-end \n-scene_objects \n-Hilbert_Cohn_Vossen_surface \n-group_of_things:"0" \n-cubics:0 -texture{pigment{White*0.5 -transmit:0.5}} \n-reflection:0.1 -specular:.8 
-group_of_things_as_interval:0-6 \n-group_of_things_as_interval:6-6 \n-group_of_things_as_interval_with_exceptions:12-15 \n-lines:0 -0.02 -texture{pigment{color:Red}} \n-lines:2 -0.02 -texture{pigment{color:Blue}} \n-lines:3 -0.02 -texture{pigment{color:Yellow}} 
-label:0 "a1" \n-label:2 "a2" \n-label:4 "a3" \n-label:6 "a4" \n-label:8 "a5" \n-label:10 "a6" \n-label:12 "b1" \n-label:14 "b2" \n-label:16 "b3" \n-label:18 "b4" \n-label:20 "b5" \n-label:22 "b6" \n-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"
-group_of_things_as_interval-12.12
-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
Eckardt_deform:
$(ORBITER_PATH)orbiter.out\-v:2\-povray
-round-0\-nb_frames.default:93
-output_mask-Eckardt_deform\_\%\_03d.pov
-video_options\-W:1024\-H:768
-W:2560\-H:1920
-W:4096\-H:3072
default_angle:75
-clipping_radius:2.4
-camera:0\:"1,1,1"\:"-3,1,3\."0.12,0.12,0.12"
```bash
16163  -end:\n16164  -scene_objects:\n16165  -Hilbert_Cohn_Vossen_surface:\n16166  -group_of_things:"0":\n16167  deformation_of_cubic_lex-93-1.107148718-1.570796327-0:\n16168  $(ECKARDT_CUBIC_DEFORM1_LEX):\n16169  $(ECKARDT_CUBIC_DEFORM2_LEX):\n16170  -group_of_things_as_interval-0.93:1\n16171  -group_is_animated-1:\n16172  -cubics-1:"texture{pigment{White*0.5-transmit-0.5}:}\n16173  finish-{ambient-0.4-diffuse-0.5-roughness-0.001-reflection-0.1-specular-.8}:".\n16174  -scene_objects_end:\n16175  -povray_end\n16176  -rm-POV\n16177  mkdir-POV\n16178  mv-Eckardt_deform_0_.pov-POV\n16179  mv-makefile_animation-POV\n16180\n16181\n16182\n16183\n16184\n16185  Eckardt_deform_2:\n16186  $(ORBITER_PATH)oribter.out-2-povray:\n16187  -round-0-nb_frames_default-30:\n16188  -output_mask-Eckardt_deform_%d_%03d.pov:\n16189  -video_options-W-1024-H-768:\n16190  -global_picture_scale-0.9:\n16191  -default_angle-75:\n16192  -clipping_radius-2.4:\n16193  -camera-0.1,1.1,3.1,3.12,0.12,0.12:\n16194  -end:\n16195  -scene_objects:\n16196  -Hilbert_Cohn_Vossen_surface:\n16197  -group_of_things-"0":\n16198  deformation_of_cubic_lex-93-1.107148718-1.570796327-0:\n16199  $(ECKARDT_CUBIC_DEFORM1_LEX):\n16200  $(ECKARDT_CUBIC_DEFORM2_LEX):\n16201  --group_of_things_as_interval-0.93:\n16202  --group_is_animated-1:\n16203  --group_of_things-"0":\n16204  --cubics-1:"texture{pigment{White*0.5-transmit-0.5}:}\n16205  finish-{ambient-0.4-diffuse-0.5-roughness-0.001-reflection-0.1-specular-.8}:".\n16206  --group_of_things-"24":\n16207  --cubics-2:"texture{pigment{Red*0.5-transmit-0.5}:}\n16208  finish-{ambient-0.4-diffuse-0.5-roughness-0.001-reflection-0.1-specular-.8}:".\n16209  --group_of_things-"70":\n```
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}}\{ambient:0.4,\diffuse:0.9,\phong:1\}}"
-lines=2:0.02:"texture{pigment{\color{Blue}}\{ambient:0.4,\diffuse:0.9,\phong:1\}}"
-lines=3:0.02:"texture{pigment{\color{Yellow}}\{ambient:0.4,\diffuse:0.9,\phong:1\}}"
-spheres=4:0.08:"texture{pigment{Cyan*1.3}}\{ambient:0.4,\diffuse:0.6,\roughness:0.001\}
-reflection=0.9,\specular=.8\}
-scene_objects_end
-povray_end
-mkdir-POV
mv-Clebsch_0_*.pov-POV
mv-makefile_animation-POV

endrass8:

$(ORBITER_PATH)orbiter.out -v -2 -povray
>> round-0 -nb_frames_default-30
-output_mask-endrass_octic_%d_%03d.pov
-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-0.15 "texture{pigment{\-color\-Red}}"
-lines-1.0.15 "texture{pigment{\-color\-Green}}"
-lines-2.0.15 "texture{pigment{\-color\-Blue}}"
-lines-3.0.05 "texture{pigment{\-color\-Black}}"
-lines-0.9.0.9-phony-1
-lines-0.9.0.9-phony-1
-octic_lex_165 $(ENDRASS_OCTIC_LEX_165)
-plane_by_dual_coordinates-0,0,1,0
-group_of_things-0
-group_of_things-0
-octics-4 "texture{pigment{\-White\-0.5\-transmit\-0.5}}"
-ambient-0.4-diffuse-0.5-roughness-0.001
-reflection-0.1-specular-.8
-planes-5 "texture{pigment{\-color\-Blue\-transmit\-0.5}}"
-endscene_objects_end

-povray_end

>> rm -rf-POV

mkdir-POV

mv-endrass_octic_0_*.pov-POV
mv-makefile_animation-POV
SECTION ANIMATIONS:

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
- ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png
  -f 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
- 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
Kummer_surface:

```
16389 \$\{\text{ORBITER\_PATH}\}\\text{orbiter.}\\text{out} -v 2 -povray\
16400 -round 0 -nb\_frames default 30\
16401 -output\_mask Kummer\_6d\_%03d.pov\
16402 -video\_options -W 1024 -H 768\
16403 -global\_picture\_scale 0.9\
16404 -default\_angle 75\
16405 -clipping\_radius 2.4\
16406 -camera 0.1,1,1 3,1,3 0.12,0.12,0.12\
16407 -end\end{verbatim}
```

```
Kummer_video:

```
16397 -rm -rf FRAMES
```

Maple:

```
\# Kummer := expand((x0^2 + x1^2 + x2^2 + x3^2)^2 - 3*(x0^4 + x1^4 + x2^4 + x3^4))
```

Beauville_surface:

```bash
$(ORBITER_PATH)orbiter.out
-prepare_frames:
  -i 0:30-KUMMER/Kummer_0_%03d.png
  -output_starts_at 0:
  -o FRAMES/frame%04d.png

ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png -f mp4 -q:v 0 -vcodec mpeg4 Kummer.mp4
```

```bash
$(ORBITER_PATH)orbiter.out
-v 2 -povray:
  -round 0 -nb_frames_default 30:
  -output_mask Beauville_%d_%03d.pov:
  -video_options -W 1024 -H 768:
  -global_picture_scale 0.3:
  -default_angle 75:
  -clipping_radius 2.4:
  -camera 0."1,1,1"."-3,1,3"."0.12,0.12,0.12":

finish {ambient 0.4 diffuse 0.5 roughness 0.001:
  reflection 0.1 specular .8}:
```

```bash
# 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:
```
\#-b=-c=1.618033988=(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, 0], [-1.6180340022062127, 0, 0, 1.6180340022062127])

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
T31=-2.6180340022062127
T32=-1.6180340022062127
T33=0

CLEBSCH_CUBICS=
push-b-push-b-mult-push-d-push-c-push-m-mult-add-mult-
push-b-push-c-push-d-push-d-push-m-mult-add-mult-
push-a-push-d-push-d-push-m-mult-mult-add-mult-
push-a-push-c-push-m-mult-add-mult-
store-c001-
push-b-push-d-mult-
push-b-push-1-push-m-push-c-mult-add-mult-
16488 push d
16489 push a
16490 push m
16491 multiply
16492 store c002
16493 push b
16494 push c
16495 push a
16496 push m
16497 push 1
16498 push m
16499 store c001
16500 push m
16501 push b
16502 push b
16503 push m
16504 push b
16505 push c
16506 store c001
16507 push m
16508 push b
16509 push b
16510 push b
16511 push b
16512 store d001
16513 push m
16514 push c
16515 push m
16516 push m
16517 push b
16518 store d001
16519 push m
16520 push c
16521 push c
16522 push b
16523 push b
16524 store d001
16525 push d
16526 push b
16527 push c
16528 push b
16529 store d112
16530 push m
16531 push b

788
t.add.mul\n16532 ▷ ▷ ▷ ▷ push-m.mul-add.push-a.mul-push-b.push-c.mul-push-d.push-1.push-m.mul.
    add mul add\n16533 ▷ ▷ ▷ ▷ push-b.push-d.push-m.mul.push-m.mul.add mul add\n16534 ▷ ▷ ▷ ▷ store-m002\n16535 ▷ ▷ ▷ ▷ push-m\n16536 ▷ ▷ ▷ ▷ push-d.push-c.push-m.mul.push-a.push-a.mul.push-m.mul.
16537 ▷ ▷ ▷ ▷ push-c.push-c.mul.push-d.push-m.mul.add push-a.mul.add\n16538 ▷ ▷ ▷ ▷ push-b.push-c.push-m.mul.push-d.push-1.push-m.mul.add mul add\n16539 ▷ ▷ ▷ ▷ push-b.push-d.push-m.mul.add mul add\n16540 ▷ ▷ ▷ ▷ store-m012\n16541 ▷ ▷ ▷ ▷ push-m\n16542 ▷ ▷ ▷ ▷ push-c.push-d.mul.push-d.push-m.mul.add push-a.push-a.mul.push-m.mul.
16543 ▷ ▷ ▷ ▷ push-m.push-c.push-c.mul.push-d.push-m.mul.add push-a.push-b.mulmul.
    t.add mul add\n16544 ▷ ▷ ▷ ▷ push-m.push-b.push-d.push-c.push-m.mul.add push-c.mul.mul.add mul add\n16545 ▷ ▷ ▷ ▷ push-b.push-d.push-m.mul.add mul add\n16546 ▷ ▷ ▷ ▷ store-m022\n16547 ▷ ▷ ▷ ▷ push-d.push-1.push-m.mul.add push-a.mul\n16548 ▷ ▷ ▷ ▷ push-m.push-b.mul.push-1.push-c.mul.add\n16549 ▷ ▷ ▷ ▷ push-b.add push-m.push-d.mul.add\n16550 ▷ ▷ ▷ ▷ push-a.push-c.mul.mul\n16551 ▷ ▷ ▷ ▷ push-b.push-d.push-m.mul.add mul add\n16552 ▷ ▷ ▷ ▷ store-m122\n16553 ▷ ▷ ▷ ▷ push-m.push-a.mul.push-c.add push-d.mul.push-c.push-a.push-1.push-m.mul.
    t.add mul add\n16554 ▷ ▷ ▷ ▷ push-b.mul\n16555 ▷ ▷ ▷ ▷ push-m.push-a.push-d.mul.push-c.push-1.push-m.mul.add mul add\n16556 ▷ ▷ ▷ ▷ push-b.push-d.push-m.mul.add mul add\n16557 ▷ ▷ ▷ ▷ store-m002\n16558 ▷ ▷ ▷ ▷ push-m\n16559 ▷ ▷ ▷ ▷ push-c.push-d.push-m.mul.add push-b.mul.push-m.push-d.push-c.push-1.push
    m.mul.add mul add\n16560 ▷ ▷ ▷ ▷ push-a.mul\n16561 ▷ ▷ ▷ ▷ push-b.push-c.mul.push-d.push-1.push-m.mul.add mul add mul add\n16562 ▷ ▷ ▷ ▷ push-a.push-b.push-c.push-m.mul.push-d.push-m.mul.add add add mul add\n16563 ▷ ▷ ▷ ▷ store-n012\n16564 ▷ ▷ ▷ ▷ push-c.push-d.push-m.mul.add push-b.mul\n16565 ▷ ▷ ▷ ▷ push-m.push-d.push-c.push-1.push-m.mul.add mul add\n16566 ▷ ▷ ▷ ▷ push-a.mul\n16567 ▷ ▷ ▷ ▷ push-b.push-c.mul.push-d.push-1.push-m.mul.add mul add\n16568 ▷ ▷ ▷ ▷ push-a.push-d.mul.push-m.push-b.push-c.mul.mul.add mul add\n16569 ▷ ▷ ▷ ▷ store-n022\n16570 ▷ ▷ ▷ ▷ push-m\n16571 ▷ ▷ ▷ ▷ push-c.push-d.push-m.mul.add push-b.mul\n16572 ▷ ▷ ▷ ▷ push-m.push-d.push-c.push-1.push-m.mul.add mul add mul add\n16573 ▷ ▷ ▷ ▷ push-a.mul\n
789
Clebsch_up_create_points:

\(\theta(\text{ORBITER\_PATH})\text{orbiter.out -v.2}\)

- smooth_curve:"Clebsch_map.of.circle.to_defininig_eqn_r2"\n
- const-a-$(\text{CLEBSCH}_A)-b-$(\text{CLEBSCH}_B)-c-$(\text{CLEBSCH}_C)-d-$(\text{CLEBSCH}_D)\)

\(t00-$(T00)\cdot t01-$(T01)\cdot t02-$(T02)\cdot t03-$(T03)\)

\(t10-$(T10)\cdot t11-$(T11)\cdot t12-$(T12)\cdot t13-$(T13)\)

\(t20-$(T20)\cdot t21-$(T21)\cdot t22-$(T22)\cdot t23-$(T23)\)

\(t30-$(T30)\cdot t31-$(T31)\cdot t32-$(T32)\cdot t33-$(T33)\)

\(r.2\cdot \text{one-1}\cdot m.-1\)

- const_end-

- var.t.-

- c001\cdot c002\cdot c011\cdot c012-

- d001\cdot d011\cdot d012\cdot d112-

- m002\cdot m012\cdot m022\cdot m122-

- n002\cdot n012\cdot n112\cdot n022\cdot n122-

- y0\cdot y1\cdot y2-

- y001\cdot y002\cdot y011\cdot y012\cdot y022\cdot y112\cdot y122-

- x0\cdot x1\cdot x2\cdot x3-

- -var_end-

- -code.-

- push.t\cdot \cos.r\cdot \mult\cdot \store.y0.-

- push.t\cdot \sin.r\cdot \mult\cdot \store.y1.-

- push.one\cdot \store.y2.-

- push.y0\cdot push.y0\cdot push.y1\cdot \mult\cdot \mult\cdot \store.y001.-

- push.y0\cdot push.y0\cdot push.y2\cdot \mult\cdot \mult\cdot \store.y002.-

- push.y0\cdot push.y1\cdot push.y1\cdot \mult\cdot \mult\cdot \store.y011.-

- push.y0\cdot push.y1\cdot push.y2\cdot \mult\cdot \mult\cdot \store.y012.-

- push.y0\cdot push.y0\cdot push.y2\cdot \mult\cdot \mult\cdot \store.y022.-

- push.y1\cdot push.y1\cdot push.y2\cdot \mult\cdot \mult\cdot \store.y112.-

- push.y1\cdot push.y1\cdot push.y2\cdot \mult\cdot \mult\cdot \store.y122.-

- $(\text{CLEBSCH}_CUBICS})-

- push.c001\cdot push.y001\cdot \mult.-

- push.c002\cdot push.y002\cdot \mult\cdot \add.-

- push.c011\cdot push.y011\cdot \mult\cdot \add.-

- push.c012\cdot push.y012\cdot \mult\cdot \add.-
16621 ▶ ▶ ▶ ▶ store:x0:\n16622 ▶ ▶ ▶ ▶ push-d001.push-y001.mult\n16623 ▶ ▶ ▶ ▶ push-d011.push-y011.mult.add\n16624 ▶ ▶ ▶ ▶ push-d012.push-y012.mult.add\n16625 ▶ ▶ ▶ ▶ push-d112.push-y112.mult.add\n16626 ▶ ▶ ▶ ▶ store:x1:\n16627 ▶ ▶ ▶ ▶ push-m002.push-y002.mult\n16628 ▶ ▶ ▶ ▶ push-m012.push-y012.mult.add\n16629 ▶ ▶ ▶ ▶ push-m022.push-y022.mult.add\n16630 ▶ ▶ ▶ ▶ push-m122.push-y122.mult.add\n16631 ▶ ▶ ▶ ▶ store:x2:\n16632 ▶ ▶ ▶ ▶ push-n002.push-y002.mult\n16633 ▶ ▶ ▶ ▶ push-n012.push-y012.mult.add\n16634 ▶ ▶ ▶ ▶ push-n022.push-y022.mult.add\n16635 ▶ ▶ ▶ ▶ push-n112.push-y112.mult.add\n16636 ▶ ▶ ▶ ▶ push-n122.push-y122.mult.add\n16637 ▶ ▶ ▶ ▶ store:x3:\n16638 ▶ ▶ ▶ ▶ push-x0.push-t00.mult\n16639 ▶ ▶ ▶ ▶ push-x1.push-t10.mult.add\n16640 ▶ ▶ ▶ ▶ push-x2.push-t20.mult.add\n16641 ▶ ▶ ▶ ▶ push-x3.push-t30.mult.add\n16642 ▶ ▶ ▶ ▶ return\n16643 ▶ ▶ ▶ ▶ push-x0.push-t01.mult\n16644 ▶ ▶ ▶ ▶ push-x1.push-t11.mult.add\n16645 ▶ ▶ ▶ ▶ push-x2.push-t21.mult.add\n16646 ▶ ▶ ▶ ▶ push-x3.push-t31.mult.add\n16647 ▶ ▶ ▶ ▶ return\n16648 ▶ ▶ ▶ ▶ push-x0.push-t02.mult\n16649 ▶ ▶ ▶ ▶ push-x1.push-t12.mult.add\n16650 ▶ ▶ ▶ ▶ push-x2.push-t22.mult.add\n16651 ▶ ▶ ▶ ▶ push-x3.push-t32.mult.add\n16652 ▶ ▶ ▶ ▶ return\n16653 ▶ ▶ ▶ ▶ push-x0.push-t03.mult\n16654 ▶ ▶ ▶ ▶ push-x1.push-t13.mult.add\n16655 ▶ ▶ ▶ ▶ push-x2.push-t23.mult.add\n16656 ▶ ▶ ▶ ▶ push-x3.push-t33.mult.add\n16657 ▶ ▶ ▶ ▶ return\n16658 ▶ ▶ -code_end
16659
16660
16661 Clebsch_surface:
16662 ▶ $(ORBITER_PATH)orbiter.out-v.2.-povray\n16663 ▶ -round0.-nb_frames_default.30\n16664 ▶ -output_mask.Clebsch.%d%03d.pov\n16665 ▶ -video_options.-W.1024.-H.768\n16666 ▶ -global_picture_scale.0.9\n16667 ▶ -default_angle.75\n
791
Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:

Clebsch_surface_defining_equation:
Clebsch_surface_defining_equation_and_curves:

```plaintext
$(ORBITER\_PATH)orbiter.out -v 2 -povray
-round 0 -nb_frames_default 30
-object_mask Clebsch_2curves_%d_%03d.pov
-video_options -W 1024 -H 768
-global_picture_scale 0.6
-default_angle 75
-clipping_radius 1.6
-camera 0.1,1,1" -2,0,2" 0,0,0"
-end
-scene_objects
   -cubic_orbiter "0,0,0,0,1,1,1,1,1,1,1,1,2,2,2,2"
   -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}
```

```plaintext
function Clebsch_map_of_circle_to_defining_eqn_N1000_points.csv
   -group_of_things_as_interval 0-656
   -spheres 1.0.0.7: "texture{pigment{color Red}}"
   finish {diffuse 0.9 -phong 1}
```

```plaintext
function Clebsch_map_of_circle_to_defining_eqn_r2_N1000_points.csv
   -group_of_things_as_interval 656-1042
   -spheres 2.0.0.7: "texture{pigment{color Blue}}"
   finish {diffuse 0.9 -phong 1}
```

```plaintext
-point_list_from_csv_file
```

```plaintext
Clebsch_2curves_0_*.pov
mv -makefile_animation.pov
```

```plaintext
# -point_list_from_csv_file:function_Clebsch_map_of_circle_N1000_points.csv
# -group_of_things_as_interval 0-954
# -spheres 1.0.0.7: "texture{pigment{color Red}}" finish {diffuse 0.9 -phong 1}
```

F7_povray:

```plaintext
$(ORBITER\_PATH)orbiter.out -v 2 -povray
-round 0 -nb_frames_default 30
-object_mask F7_15_lines_%d_%03d.pov
```

793
16795
16796
16797 F7_video:
16798 ▷ -rm -r FRAMES
16799 ▷ -mkdir FRAMES
16800 ▷ -rm fifteen_with_lines.mp4
16801 ▷ $(ORBITER_PATH)orbiter.out\n16802 ▷ ▷ -prepare_frames:\n16803 ▷ ▷ ▷ -i 0-30 F7b/F7_15_lines_0_%03d.png\n16804 ▷ ▷ ▷ -output_starts_at 0\n16805 ▷ ▷ ▷ -o FRAMES/frame%04d.png\n16806 ▷ ▷ ▷ -end
16807 ▷ ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png\n16808 ▷ ▷ -f mp4 -q:v 0 -vcodec mpeg4 fifteen_with_lines.mp4
16809
16810
16811 McKean_povray:
16812 ▷ $(ORBITER_PATH)orbiter.out -v 2 -povray\n16813 ▷ ▷ -round 0 -nb frames default 30\n16814 ▷ ▷ -output mask McKean_%d_%03d.pov\n16815 ▷ ▷ -video_options -W 1024 -H 768\n16816 ▷ ▷ -global picture scale 1.5\n16817 ▷ ▷ -default angle 80\n16818 ▷ ▷ -clipping radius 4.4\n16819 ▷ ▷ -omit_bottom_plane\n16820 ▷ ▷ -camera 0 "1,1,1" "-4.5,3.5,6" "0,0,0"\n16821 ▷ ▷ -end\n16822 ▷ ▷ -scene_objects\n16823 ▷ ▷ ▷ -cubic lex "0,0,1,0,0,-1,-2,1,\n16824 2,0,-1,1,1,-1,1,0,0,0,0"\n16825 ▷ ▷ ▷ -group_of_things "0"\n16826 ▷ ▷ ▷ -cubics 0 "texture pigment White*0.5" \n16827 finish "ambient 0.4 diffuse 0.5 roughness 0.001" \n16828 reflection 0.1 specular .8"\n16829 ▷ ▷ -scene_objects_end\n16830 ▷ -povray_end
16831 ▷ -rm -rf POV
16832 ▷ mkdir POV
16833 ▷ mv McKean_0*.pov POV
16834 ▷ mv makefile_animation-POV
16835
16836 McKean_video:
16837 ▷ -rm -r FRAMES
16838 ▷ -mkdir FRAMES
16839 ▷ -rm McKean.mp4
16840 ▷ $(ORBITER_PATH)orbiter.out\n16841 ▷ ▷ -prepare_frames:\n
795
Section 15.4: Continuous Function Plotter

### lissajous:

```sh
$(ORBITER_PATH) orbiter.out -v 2
```

```sh
-smooth_curve "lissajous":0.07.2000:15.0:18.85:
```

```sh
-const:a:3:b:2:c:1.57:r:7:-const_end:
```

```sh
-var:t:-var_end:
```

```sh
-code:
```

```sh
push t push a mult push c add sin push r mult return
```

```sh
push t push b mult sin push r mult return
```

```sh
-code_end
```

### lissajous_plot:

```sh
$(ORBITER_PATH) orbiter.out -v 2 -povray
```

```sh
-round:0:-nb_frames_default:1:
```

```sh
-output_mask lissajous_%d.png
```

```sh
-video_options:W:1024:H:768:
```

```sh
-global_picture_scale:0.40:
```

```sh
-default_angle:45:
```

```sh
-clipping_radius:5:
```

```sh
-omit_bottom_plane:
```

```sh
-camera:0,-1,0.0,0,12.0,0,0,0:
```

```sh
-rotate about z axis:
```

```sh
-end:
```

```sh
-scene_objects:
```

```sh
-line through two points recentered from csv file:
```

```sh
-coordinate_grid.csv:
```

```sh
-group of things:"0":"1".
```

```sh

ffmpeg -r 5 -f image2 FRAMES/frame%04d.png
```

```sh
-f mp4 -q:v 0 -vcodec mpeg4 McKean.mp4
```

#function_lissajous_N2000_points.csv

#scene_objects:

```sh
-line through two points recentered from csv file:
```

```sh
-coordinate_grid.csv:
```

```sh
-group of things:"0":"1".
```
16888  
16889  
16890  
16891  
16892  
16893  
16894  
16895  
16896  
16897  
16898  
16899  
16900  
16901  
16902  
16903  
16904  
16905  
16906  
16907  
16908  
16909  
16910  
16911  
16912  
16913  
16914  
16915  
16916  
16917  
16918  
16919  
16920  
16921  
16922  
16923  
16924  
16925  
16926  
16927  
16928  
16929  
16930  
16931  
16932  
16933  
16934  
16935  

797
# Chapter 16 - Miscellaneous

### Section 16.1: Miscellaneous

misc_select:

```bash
$(ORBITER_PATH)orbiter.out -v 3
```

```bash
$ -define F -finite_field -q 7 -end
```

```bash
$ -with F -do -finite_field_activity --cheat_sheet_GF -end
```

```bash
$(ORBITER_PATH)orbiter.out -v 4 -csv_file select_rows_and_cols
```

```bash
GF_q7_multiplication_table_reordered.csv
```
"0,2,4","0,2,4".

misc_join:

$(ORBITER_PATH)orbiter.out -v -4

- csv_file_join: poly_orbits.d3_n3_q2_select_F2.csv Orbit.idx
- csv_file_join: poly_orbits.d3_n3_q2_select_F4.csv Orbit.idx
- csv_file_join: poly_orbits.d3_n3_q2_select_F8.csv Orbit.idx
- csv_file_join: poly_orbits.d3_n3_q2_select_F16.csv Orbit.idx
- csv_file_join: poly_orbits.d3_n3_q2_select_F32.csv Orbit.idx

# Section 16.2: Limitations

SECTION LIMITATIONS:

###
Bibliography


[57] L. Schläffi. 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.


