User’s Guide
Build Number 1377

Anton Betten

December 29, 2021
## Contents

1 Introduction ............................................. 7

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

3 Basic Algebra ........................................... 31
   3.1 Basic Number Theory ............................... 31
   3.2 Prime Fields .................................... 35
   3.3 Polynomials Over Finite Fields ..................... 39
   3.4 Extension Fields ................................ 44
   3.5 Linear Algebra Over Finite Fields ................. 51

4 Geometry ................................................ 55
   4.1 Finite Projective Spaces ......................... 55
   4.2 Indexing Points and Lines ......................... 62
   4.3 Finite Desarguesian Projective Planes ............. 70
   4.4 The Grassmannian ................................ 74
   4.5 Algebraic Sets ................................... 75
   4.6 The Klein Quadric and the Plücker Map ............ 79
   4.7 Orthogonal Spaces ................................ 81
   4.8 Hermitian Varieties ............................... 88
   4.9 Advanced Topics ................................ 90

5 Group Theory ........................................... 97
   5.1 Permutation Groups ................................ 97
   5.2 Linear Groups .................................... 104
   5.3 Subgroups ....................................... 120
   5.4 Linear Groups, Advanced Topics .................... 127
   5.5 Induced Actions ................................ 131
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6</td>
<td>Group Theoretic Activities</td>
<td>140</td>
</tr>
<tr>
<td>5.7</td>
<td>Group Theoretic Activities Based on Magma</td>
<td>145</td>
</tr>
<tr>
<td>6</td>
<td>Orbit Algorithms</td>
<td>151</td>
</tr>
<tr>
<td>6.1</td>
<td>Schreier Trees</td>
<td>151</td>
</tr>
<tr>
<td>6.2</td>
<td>Poset Classification</td>
<td>159</td>
</tr>
<tr>
<td>6.3</td>
<td>Orbits on Subsets</td>
<td>163</td>
</tr>
<tr>
<td>6.4</td>
<td>Orbits on Subspaces</td>
<td>166</td>
</tr>
<tr>
<td>6.5</td>
<td>Arcs and Caps in Projective Spaces</td>
<td>168</td>
</tr>
<tr>
<td>6.6</td>
<td>Cubic Curves</td>
<td>173</td>
</tr>
<tr>
<td>7</td>
<td>Cubic Surfaces</td>
<td>175</td>
</tr>
<tr>
<td>7.1</td>
<td>Creation</td>
<td>175</td>
</tr>
<tr>
<td>7.2</td>
<td>Quartic Curves</td>
<td>180</td>
</tr>
<tr>
<td>7.3</td>
<td>Classification</td>
<td>188</td>
</tr>
<tr>
<td>7.4</td>
<td>Isomorphism Testing and Recognition</td>
<td>203</td>
</tr>
<tr>
<td>7.5</td>
<td>Dickson Surfaces</td>
<td>206</td>
</tr>
<tr>
<td>7.6</td>
<td>ATLAS and Tables</td>
<td>208</td>
</tr>
<tr>
<td>8</td>
<td>Applications</td>
<td>209</td>
</tr>
<tr>
<td>8.1</td>
<td>Number Theory</td>
<td>209</td>
</tr>
<tr>
<td>8.2</td>
<td>Combinatorics</td>
<td>212</td>
</tr>
<tr>
<td>8.3</td>
<td>Representation Theory</td>
<td>214</td>
</tr>
<tr>
<td>8.4</td>
<td>Cryptography</td>
<td>216</td>
</tr>
<tr>
<td>9</td>
<td>Coding Theory</td>
<td>225</td>
</tr>
<tr>
<td>9.1</td>
<td>Introduction</td>
<td>225</td>
</tr>
<tr>
<td>9.2</td>
<td>Hamming Codes</td>
<td>229</td>
</tr>
<tr>
<td>9.3</td>
<td>Golay Codes</td>
<td>234</td>
</tr>
<tr>
<td>9.4</td>
<td>CRC Codes</td>
<td>235</td>
</tr>
<tr>
<td>9.5</td>
<td>Reed-Muller Codes</td>
<td>239</td>
</tr>
<tr>
<td>9.6</td>
<td>BCH Codes</td>
<td>241</td>
</tr>
<tr>
<td>9.7</td>
<td>Reed-Solomon Codes</td>
<td>243</td>
</tr>
<tr>
<td>9.8</td>
<td>Bounds</td>
<td>248</td>
</tr>
<tr>
<td>9.9</td>
<td>Classification of Optimal Linear Codes</td>
<td>250</td>
</tr>
<tr>
<td>10</td>
<td>Incidence Geometry</td>
<td>253</td>
</tr>
<tr>
<td>10.1</td>
<td>Diophantine Systems</td>
<td>253</td>
</tr>
<tr>
<td>10.2</td>
<td>Combinatorial Linear Spaces</td>
<td>257</td>
</tr>
<tr>
<td>10.3</td>
<td>Classification of Configurations and Geometries</td>
<td>260</td>
</tr>
<tr>
<td>10.4</td>
<td>Design Theory</td>
<td>265</td>
</tr>
<tr>
<td>10.5</td>
<td>Design Theory – Large Sets</td>
<td>269</td>
</tr>
<tr>
<td>10.6</td>
<td>Design Theory – Delandtsheer-Doyen</td>
<td>272</td>
</tr>
<tr>
<td>10.7</td>
<td>Tactical Decompositions</td>
<td>274</td>
</tr>
<tr>
<td>10.8</td>
<td>Spreads</td>
<td>278</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 [9].
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. [6]). Docker [21] 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. [24]). This is a software tool that allows collecting short command snippets in the form of text files that can easily be handled. However, the conventions in the tool involve some subtleties regarding the use of whitespace, which can cause problems to novice users. We will point out possible pitfalls along the way. Note that it is not necessary to use makefiles. Another possibility would be to use shell scripts. Ultimately, it would be possible to type out all commands into a terminal window. This could be a little tedious though, considering the fact that most Orbiter commands expect lengthy parameters from the command line.

Let us start by discussing how to run Orbiter as a native application. To do so, a unix-like compile environment is required, including a modern C++ compiler and the tools git and make. Windows users may need to install Cygwin [18]. 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/\_/ossr.\_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:

```
├── orbiter
│   └── work
```

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

test:
   $(ORBITER_PATH)orbiter.out

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

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

test:
   $(ORBITER_PATH)orbiter.out

is the makefile target “test.” It is important that the indentation after the makefile target is
done using tab characters (no spaces). There can be multiple targets in one makefile, as long
as they are separated by an empty line. For more information about the syntax of makefiles,
see [24].

A second way to run Orbiter is through Docker [21]. This does not require a compile environ-
ment. However, it comes at a small performance cost when running Orbiter commands that
are computationally heavy. Orbiter has already been precompiled (by the Orbiter developer)
into an image, which is a completely self-sustained copy of a unix-environment that can run
by the user under the docker front-end. The image is stored on a docker server under the
name abetten/orbiter. Docker will receive the name of the image from the command line,
pull a local copy of the image, and run the image in an encapsulated environment called a
container. A copy of the image is stored locally, so that subsequent calls to Orbiter can be
satisfied using the local copy, which increases turnaround speed. For instance, the following
bare-bones makefile sets up Orbiter for use through Docker:

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

test:
   $(ORBITER_PATH)orbiter.out

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

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

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

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

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

```make
1  OP=~/orbiter
2  OP2=$(OP)/src/apps/orbiter/
3  DOCKER_OPTS=run-it-
4     -it
5     -v
6     $(OP)/src/apps/orbiter:
7     -w
8     mnt
9     -w
10    mnt
11    abetten/orbiter
12   #ORBITER PATH=ORBITER PATH=ORBITER
13   ORBITER PATH=$(OP2)
```

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

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

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

(c) Type

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

This will produce an output similar to the following:

```
sh-3.2$ docker run -it --volume $PWD:/mnt -w /mnt abetten/orbiter orbiter.out
Unable to find image 'abetten/orbiter:latest' locally
latest: Pulling from abetten/orbiter
004f1eed87df: Pull complete
5d6f1e8117db: Pull complete
48c2fa66abe: Pull complete
234b70d0479d: Pull complete
6fa07a00e2f0: Pull complete
9187bd98e241: Pull complete
ae87b7ef500b: Pull complete
260a2765fa99: Pull complete
27d6ff93a58: Pull complete
7a09ec574418: Pull complete
1336494f74e1: Pull complete
Digest: sha256:889099d7e0b0a9ee168b7cb261d2da8ff64bd7d861c357e1caec59580d629ee9
Status: Downloaded newer image for abetten/orbiter:latest
Welcome to Orbiter! Your build number is 1311.
A user’s guide is available here:
https://www.math.colostate.edu/~betten/orbiter/users_guide.pdf
The sources are available here:
https://github.com/abetten/orbiter
An example makefile with many commands from the user’s guide is here:
https://github.com/abetten/orbiter/tree/master/examples/users_guide/makefile
SYSTEMUNIX is defined
sizeof(int)=4
sizeof(long int)=8
Orbiter session finished.
User time: 0:00
```

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

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

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

```
$ 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:

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

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

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

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

One tool that stands out in the unix world is called make. Make is a command that allows to execute certain processes on a need basis. The need is defined by means of time stamps on files. The rules are defined in a file called makefile. Make is very popular in software engineering, where there are dependencies between source code, object code and executable files. We note that it is not necessary to use makefiles. However, because of the convenience they offer in defining lightweight commands, this user’s guide will rely on 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 15.1. 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

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 15.1. 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:
\$\left(\text{ORBITER\_PATH}\right)\text{orbiter\_out} -v 3 -define F -finite\_field -q 16 -end \ 
-\text{with F} -do -finite\_field\_activity -cheat\_sheet\_GF -end

We recoomend 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

\text{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

\text{ORBITER\_PATH=\$\left(\text{MY\_PATH}\right)/src/apps/orbiter/}

to

\text{ORBITER\_PATH=}

in the makefile.
2.4 Objects and Activities

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

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

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

The syntax to create an object is

\texttt{-define \textit{LABEL} \textit{KEYWORD} \textit{EXTRAS} -end}

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

\begin{verbatim}
object_F.2:
▷ $\$(ORBITER_PATH)orbi\textunderscore t.out\:-v\:3\:-define\textunderscore F\:-finite\textunderscore field\:-q\:2\:-end
\end{verbatim}

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

\begin{verbatim}
object_PG.3.2:
▷ $\$(ORBITER_PATH)orbi\textunderscore t.out\:\\n▷ ▷ -define\textunderscore F\:-finite\textunderscore field\:-q\:2\:-end\:\\n▷ ▷ -define\textunderscore P\:-projective\textunderscore space\:3\:F\:-end
\end{verbatim}

creates the same finite field \(F\) as well as an object \(P\) representing \(\text{PG}(3,2)\). Note how the creation of \(P\) relies on the existence of \(F\). The \texttt{-projective_space} option requires two parameters, the dimension of the projective space and the field over which it is defined. In
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-finite_field</td>
<td>A finite field $\mathbb{F}_q$. See Sections 3.2 and 3.4.</td>
</tr>
<tr>
<td>-projective_space</td>
<td>A projective space of dimension $n$ over a finite field $F$. See Section 4.1.</td>
</tr>
<tr>
<td>-orthogonal_space</td>
<td>A non-degenerate orthogonal space. See Section 4.7.</td>
</tr>
<tr>
<td>-linear_group</td>
<td>A linear group. See Section 5.2.</td>
</tr>
<tr>
<td>-permutation_group</td>
<td>A permutation group. See Section 5.1.</td>
</tr>
<tr>
<td>-formula</td>
<td>A symbolic expression. See Section 9.6.</td>
</tr>
<tr>
<td>-collection</td>
<td>A collection of objects.</td>
</tr>
<tr>
<td>-graph</td>
<td>A graph. See Section 11.1.</td>
</tr>
<tr>
<td>-spread_table</td>
<td>A table of spreads. See Section 10.10.</td>
</tr>
<tr>
<td>-packing_with_symmetry_assumption</td>
<td>A generator for packings with assumed symmetry. See Section 10.10.</td>
</tr>
<tr>
<td>-packing_choose_fixed_points</td>
<td>A selection of fixed orbits for packings with assumed symmetry. See Section 10.10.</td>
</tr>
<tr>
<td>-packing_long_orbits</td>
<td>A search for long orbits for packings with assumed symmetry. See Section 10.10.</td>
</tr>
<tr>
<td>-graph_classification</td>
<td>An object which allows classifying graphs and tournaments. See Section 11.3.</td>
</tr>
<tr>
<td>-diophant</td>
<td>A diophantine system, i.e., a system of positive integer equations). See Section 10.1.</td>
</tr>
<tr>
<td>-design</td>
<td>A combinatorial design. See Section 10.1.</td>
</tr>
<tr>
<td>-design_table</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 10.4.</td>
</tr>
<tr>
<td>-large_set_with_symmetry_assumption</td>
<td>An object to create a large set of designs. See Section 10.4.</td>
</tr>
<tr>
<td>-set</td>
<td>A set. See Section 2.6.</td>
</tr>
<tr>
<td>-vector</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 10.8 and 10.9.

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 11.1.</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 10.10.</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 10.10.</td>
</tr>
<tr>
<td><code>-packing_fixed_points_activity</code></td>
<td>An activity related to creating packings with assumed symmetry group, see Section 10.10.</td>
</tr>
<tr>
<td><code>-graph_classification_activity</code></td>
<td>An activity for a classification of graphs problem, see Section 11.3.</td>
</tr>
<tr>
<td><code>-diophant_activity</code></td>
<td>An activity for a diophantine system, see Section 10.1.</td>
</tr>
<tr>
<td><code>-design_activity</code></td>
<td>An activity for a combinatorial design, see Section 10.4.</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 10.5.</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 ( F_2, F_3, F_5 ). Orbiter knows the classification of cubic surfaces with 27 lines for all fields ( 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 ( F_q ) for ( q = 9, 13, 19, 25, 27, 29, 31 ).</td>
</tr>
<tr>
<td>Spreads</td>
<td>A spread is a set of ( q^k+1 ) pairwise non-intersecting ( k )-dimensional subspaces of ( \mathbb{F}_q^{2k} ). Spreads are related to translation planes of order ( q^k ). Orbiter knows the classification of spreads for ( (q, k) \in {(2,2), (3,2), (2,4), (4,2), (5,2), (3,3)} ).</td>
</tr>
<tr>
<td>Hyperovals</td>
<td>A hyperoval in PG((2, 2^e)) is a set of (2^e+2) points, no three collinear. Orbiter knows the classification of hyperovals for (e = 3, 4, 5).</td>
</tr>
<tr>
<td>Dual hyperovals</td>
<td>A ( k )-dimensional dual hyperoval in an ambient space ( \mathbb{F}_2^n ) is called a DH((k, n)). Orbiter knows the classification of dual hyperovals DH((4, 7)) and DH((4, 8)).</td>
</tr>
<tr>
<td>Packings</td>
<td>Orbiter knows the classification of packings of PG((3, 3)).</td>
</tr>
</tbody>
</table>

Table 2.4: Mathematical Data Available in Orbiter

2.5 Mathematical Data

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

The command

```bash
create_BLT_5_1:
 ▷ $(ORBITER_PATH)orbiter.out -v 2 \n ▷ ▷ -define:F:-finite_field:-q:5:-end  
 ▷ ▷ -define:0:-orthogonal_space:0:5:F:-end  
```


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 10.11.

The command

```
create_surface_4_0:
   $(ORBITER_PATH)orbiter.out-v.3:\n   -defineF:-finite_field-q4:-end:\n   -defineP:-projective_space3F:-end:\n   -withP:-do:\n   -projective_space_activity:\n   -define_surfaceS4_0:-q4:-catalogue0:-end:\n   -end:\n   -withS4_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 14.2.

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

vector_example1:

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

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

vector_example2:

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

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

```
matrix_example1:
▷ $(ORBITER_PATH)orbiter.out -v 2
▷ ▷ -define F : finite_field : q : 2 : end
▷ ▷ -print_symbols
```

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

```
CONWAY_GEN1="110111000100001010000\n111100111010001011100\n01010100000011001101\n00001000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101\n00100000000100010101";
```

```
matrix_example_co_1:
▷ $(ORBITER_PATH)orbiter.out -v 2
▷ ▷ -define F : finite_field : q : 2 : end
▷ ▷ -print_symbols
```

Using the dense option, spaces in the input string are ignored. For large vectors, the `sparse` command can be used to enter non-zero coefficients as a list of pairs. For instance,
vector_example_sparse:
  ▶ $(ORBITER_PATH)orbiter.out -v.2\n  ▶ ▶ -define:F:-finite_field:-q.5:-end\n  ▶ ▶ -define:v:-vector-field:F:-format:4:-sparse:20:"1,0,1,19":-end\n  ▶ ▶ -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:

\[
\begin{align*}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{align*}
\]

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

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

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

\[(0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2)\].

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

vector_example_all_one_11:
  ▶ $(ORBITER_PATH)orbiter.out -v.2\n  ▶ ▶ -define:v:-vector-repeat:1:11:-end\n  ▶ ▶ -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. For instance, the command

PR29:
\[
\text{ORBITER,out\,-v\,-smallest\,primitive\,root\,29}
\]

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

PR2:
\[
\text{ORBITER,out\,-v\,-primitive\,root\,915839}
\]

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

PM2:
\[
\text{ORBITER,out\,-v\,-power\,mod\,43085\,49842\,915839}
\]

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

DL2:
\[
\text{ORBITER,out\,-v\,-discrete\,log\,487320\,43085\,915839}
\]

The answer to this command is 49842. This command is a brute force search, and can be quite expensive. The command
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-power_mod</td>
<td>(a \ p)</td>
<td>Computes (a^n \mod p).</td>
</tr>
<tr>
<td>-primitive_root</td>
<td>(p)</td>
<td>Computes a primitive root modulo (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>
</tbody>
</table>
| -extended_gcd        | \(a \ b\) | Computes integers \(g, u,\) and \(v\) such that 
|                      |           | \(g = \gcd(a, b) = ua + vb\).                 |
| -square_root_mod     | \(a \ p\) | Computes a square root of \(a\) modulo \(p\), i.e. 
|                      |           | an integer \(b\) such that \(b^2 \equiv a \mod p\). |
| -square_root         | \(a\)    | Computes \(\lfloor \sqrt{a} \rfloor\) of an integer \(a\). |
| -inverse_mod         | \(a \ p\) | Computes the modular inverse of \(a\) modulo \(p\), i.e. an integer \(b\) with \(ab \equiv 1 \mod p\). |
| -draw_mod_n          | descr     | Draws the integers modulo \(n\) on a circle.    |

Table 3.1: Basic Number Theory Commands

**IM:**

\[\texttt{orbiter.out-\(-v\-5\-inverse\_mod\1865025205\-2147483647}\]

computes the inverse of 1865025205 modulo 2147483647 which is 579785381. A different way of computing the inverse is using the 1-trick. The \(-\text{extended\_gcd}\) command can be used:

**IM\_gcd:**

\[\texttt{orbiter.out-\(-v\-5\-\text{extended\_gcd}\1865025205\-2147483647}\]

This command produces the output

\[1 = -503526232 \times 2147483647 + 579785381 \times 1865025205\]

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

\[a^e \mod n,\]

the \(-\text{power\_mod}\) command can be used. For instance,

**PM3a:**

\[\texttt{orbiter.out-\(-v\-5\-\text{power\_mod}\16807\-1073741823\-2147483647}\]

32
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:
$\text{(ORBITER_PATH)orbiter.out.-v.2.-square_root_mod.33.41}$
```

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

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

This command applies the algorithm of Tonelli and Shanks (cf. [16]).

The command

```
draw_mod_13:
$\text{(ORBITER_PATH)orbiter.out.-v.2.\}
\text{-draw_options.-embedded.-end.\}
\text{-draw_mod_n.-n.13.-file_mod_13.-power_cycle.2.-end}
\text{pdflatex-mod_13_draw.tex}
\text{open-mod_13_draw.pdf}$
```

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:
$\text{(ORBITER_PATH)orbiter.out.-v.2.\}
\text{-draw_options.-scale.0.8.-embedded.-end.\}
\text{-draw_mod_n.-n.127.-file_mod_127.-power_cycle.3.-end}
\text{pdflatex-mod_127_draw.tex}
\text{open-mod_127_draw.pdf}$
```

creates the drawing shown in Figure 3.2.
Figure 3.1: Cycle of powers of $b$ modulo 13

Figure 3.2: Cycle of powers of 3 modulo 127
3.2 Prime Fields

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

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 14.2 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 $\mathbb{F}_p$ are the residue classes of the ideal given by the integer multiples of $p$. Each residue class has the form

$$\{a + kp \mid k \in \mathbb{Z}\}.$$

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

A finite field $\mathbb{F}_q$ can be created using the `-finite_field` command. Table 3.2 lists Orbiter commands for creating a finite field that can come after `-finite_field`. For instance,

```
F_2:
▷ $(\text{ORBITER:\text{PATH}})\text{orbiter.out}\cdot-v\cdot3\cdot\text{-define F}\cdot\text{-finite_field}\cdot-q\cdot2\cdot\text{-end}\cdot\$
▷ ▷ -with\cdot F\cdot\text{-do}\cdot\text{-finite_field_activity}\cdot\text{-cheat_sheet_GF}\cdot\text{-end}
▷ pdflatex\cdot GF_2.tex
▷ open\cdot GF_2.pdf
```

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

Table 3.3 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.
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>
</tbody>
</table>

Table 3.2: Options for Creating Finite Fields
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-cheat_sheet_GF</code></td>
<td></td>
<td>Produce a cheat sheet in latex which shows information about the field, including addition and multiplication tables.</td>
</tr>
</tbody>
</table>

Table 3.3: Finite Field Activities

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

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
i & \gamma_i & -\gamma_i & \gamma_i^{-1} & \log_\alpha(\gamma_i) & \alpha_i & Z_i \\
\hline
0 & 0 = 0 & 0 & DNE & DNE & 1 & 2 \\
1 & 1 = 1 & 6 & 1 & 0 & 3 & 4 \\
2 & 2 = \alpha^2 & 5 & 4 & 2 & 2 & 1 \\
3 & 3 = \alpha & 4 & 5 & 1 & 6 & DNE \\
4 & 4 = \alpha^4 & 3 & 2 & 4 & 4 & 5 \\
5 & 5 = \alpha^5 & 2 & 3 & 5 & 5 & 3 \\
6 & 6 = \alpha^3 & 1 & 6 & 3 & 1 & 2 \\
\hline
\end{array}
\]

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

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\cdot & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 1 & 2 & 3 & 4 & 5 & 6 \\
2 & 2 & 4 & 6 & 1 & 3 & 5 \\
3 & 3 & 6 & 2 & 5 & 1 & 4 \\
4 & 4 & 1 & 5 & 2 & 6 & 3 \\
5 & 5 & 3 & 1 & 6 & 4 & 2 \\
6 & 6 & 5 & 4 & 3 & 2 & 1 \\
\hline
\end{array}
\]
There is a second way of labeling the elements which is sometimes used. 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.
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

```
-finite_field_activity ...
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 `-override_polynomial a` option can be used to specify the polynomial $m(X)$ as integer $a$ in the base $p$ representation. This option can be omitted, in which case Orbiter will use a precomputed and built-in polynomial. Table 3.4 lists Orbiter activities for polynomials over finite fields. For instance, the command

```
poly_division:
  $\$(ORBITER_PATH)orbiter.out-\v.2\$
  -define:F:-finite_field:-q.2:-end.\$
  -with:F:-do.\$
  -finite_field_activity.\$
  -polynomial_division:"1,0,0,0,0,0,0,0,0,0,1"."1,0,1,1".-end
```

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 `-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.

```
poly_division2:
  $\$(ORBITER_PATH)orbiter.out-\v.2\$
  -define:F:-finite_field:-q.2:-end.\$
  -define:A:-vector:-field:F:-sparse:11:"1,0,1,10".-end.\$
```

39
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-polynomial_division</td>
<td>$A(X) \ B(X)$</td>
<td>Polynomial division of $A(X)$ by $B(X)$ over $\mathbb{F}_q$. $A(X)$ and $B(X)$ are given as coefficient list, starting from the lowest coefficient.</td>
</tr>
<tr>
<td>-extended_gcd_for_polynomials</td>
<td>$A(X) \ B(X)$</td>
<td>Extended gcd for polynomials $A(X)$ and $B(X)$ over $\mathbb{F}_q$. $A(X)$ and $B(X)$ are given as coefficient list, starting from the lowest coefficient.</td>
</tr>
<tr>
<td>-polynomial_mult_mod</td>
<td>$A(X) \ B(X)$</td>
<td>Multiply the polynomials $A(X)$ and $B(X)$ modulo $M(X)$ in $\mathbb{F}_q[X]$.</td>
</tr>
<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 3.4: 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,

\[
poly\_gcd:
\]

\[
\begin{align*}
\texttt{define}\!&\!::F\!\cdot\!\text{finite}\!\cdot\!\text{field}\!\cdot\!\text{q}\!\cdot\!2\!\cdot\!\text{end}\!\cdot\!\\
\texttt{with}\!&\!::F\!\cdot\!\text{do}\!\cdot\!\\
\texttt{finite}\!&\!\cdot\!\text{field}\!\cdot\!\text{activity}\!\cdot\!\\
\texttt{-extended}\!&\!\cdot\!\text{gcd}\!\cdot\!\text{for}\!\cdot\!\text{polynomials}\!\cdot\!\text{"1,0,0,0,0,0,0,0,0,1".\!\text{"1,0,1,1".\!\text{end}}}
\end{align*}
\]

computes

\[
U(X) = X + 1, \quad V(X) = X^8 + X^5 + X^4 + X^3 + X, \quad G(X) = 1.
\]

The Berlekamp matrix can be used to test if a polynomial is irreducible over a given finite field. The polynomial is irreducible if and only if the rank of the Berlekamp matrix is \( d - 1 \), where \( d \) is the degree of the polynomial. For instance, the command

\[
\text{Berlekamp\_matrix\_2,3:}
\]

\[
\begin{align*}
\texttt{define}\!&\!::F\!\cdot\!\text{finite}\!\cdot\!\text{field}\!\cdot\!\text{q}\!\cdot\!2\!\cdot\!\text{end}\!\cdot\!\\
\texttt{with}\!&\!::F\!\cdot\!\text{do}\!\cdot\!\\
\texttt{finite}\!&\!\cdot\!\text{field}\!\cdot\!\text{activity}\!\cdot\!\\
\texttt{-Berlekamp\_matrix\!\cdot\!\text{"1,1,0,1".\!\text{end}}}
\end{align*}
\]

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:
▷ $(ORBITER_PATH)orbiter.out-v.3-
▷ ▷ -search_for_primitive.polynomial_in_range:2:2:2:10:#\|grep://
```

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:
▷ $(ORBITER_PATH)orbiter.out-v.6-
▷ ▷ -define:F:-finite_field:-q.4:-end-
▷ ▷ -with:F:-do-
▷ ▷ -finite_field_activity-
▷ ▷ -make_table_of_irreducible_polynomials:3:-end
▷ pdflatex:Irred_q4_d3.tex
▷ open:Irred_q4_d3.pdf
```

It produces a table of all irreducible polynomials of degree 3 over \( \mathbb{F}_4 \). The output is:
There are 20 irreducible polynomials of degree 3 over the field $\mathbb{F}_4$:

0 : 1123 : 91
1 : 1031 : 77
2 : 1213 : 103
3 : 1323 : 123
4 : 1322 : 122
5 : 1222 : 106
6 : 1021 : 73
7 : 1101 : 81
8 : 1333 : 127
9 : 1232 : 110
10 : 1113 : 87
11 : 1233 : 111
12 : 1301 : 113
13 : 1003 : 67
14 : 1112 : 86
15 : 1002 : 66
16 : 1312 : 118
17 : 1011 : 69
18 : 1132 : 94
19 : 1201 : 97
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

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

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

- $X \cdot X = X^2 \equiv X + 1 \mod X^2 + X + 1$,
- $X \cdot (X + 1) = X^2 + X \equiv 1 \mod X^2 + X + 1$,
- $(X + 1) \cdot X = X^2 + X \equiv 1 \mod X^2 + X + 1$,
- $(X + 1) \cdot (X + 1) = X^2 + 1 \equiv X \mod X^2 + X + 1$,

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

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

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

Table 3.5 lists Orbiter activities for finite fields. This extends Table 3.3 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 $\mathbb{F}_{q^d}$.</td>
</tr>
</tbody>
</table>

Table 3.5: More Finite Field Activities

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

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

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

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

The primitive polynomials used by Orbiter to create small finite fields are listed in Table 3.6. 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>$\delta^4 = \delta^3 + 1$</td>
</tr>
<tr>
<td>16</td>
<td>$X^4 + X^3 + 1$</td>
<td>25</td>
<td>$\eta^5 = \eta^2 + 1$</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>$\zeta^7 = \zeta^6 + 1$</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></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 3.6: Orbiter primitive polynomials for fields $\mathbb{F}_q$ with $q \leq 1024$
Table 3.7: The field $\mathbb{F}_{16}$

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Polynomial</th>
<th>Numerical Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_4$</td>
<td>$X^2 + X + 1$</td>
<td>$7$</td>
</tr>
</tbody>
</table>

Unlike other computer algebra systems (GAP [25] and Magma [13]), 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 3.8
Table 3.8: The subfields of \( \mathbb{F}_{64} \)

<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>

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 \( \mathbb{F}_9 \), see Figure 3.7.

A normal basis for a field extension \( \mathbb{F}_{p^d} \) over \( \mathbb{F}_p \) is a basis of \( \mathbb{F}_{p^d} \) as vector space over \( \mathbb{F}_p \) which consists of one cycle of the Frobenius automorphism of \( \mathbb{F}_{p^d} \) over \( \mathbb{F}_p \). For instance, the command
Figure 3.7: Addition and multiplication table of \( \mathbb{F}_9 \) using the cyclic ordering of elements

\[ \begin{array}{ccc}
\oplus & \bullet & \circ \\
\circ & \oplus & \bullet \\
\bullet & \circ & \oplus \\
\end{array} \]

normal_basis_2_3:

\[
\text{\$ (ORBITER\_PATH) orbiter.out \ -v\_2\$}
\]

\[
\text{\$ -define \ F \ -finite\_field \ -q\_2 \ -end \$}
\]

\[
\text{\$ -with \ F \ -do \ -finite\_field\_activity \$}
\]

\[
\text{\$ -normal\_basis\_3 \ -end \$}
\]

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

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

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

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

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

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

\[
b_2^\Phi = X^2 = b_3,
\]

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

Thus,

\[
b_1 \mapsto b_2 \mapsto b_3 \mapsto b_1
\]

under \( \Phi \), as required.
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-RREF</td>
<td>$m \ n \ L$</td>
<td>Compute the RREF of the $m \times n$ matrix $L$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td>-nullspace</td>
<td>$m \ n \ L$</td>
<td>Compute a basis for the right nullspace of the $m \times n$ matrix $L$</td>
</tr>
<tr>
<td>-normalize_from_the_right</td>
<td></td>
<td>Normalizes the result of -RREF or nullspace from the right</td>
</tr>
<tr>
<td>-normalize_from_the_left</td>
<td></td>
<td>Normalizes the result of -RREF or nullspace from the left</td>
</tr>
<tr>
<td>-eigenstuff</td>
<td>$d \ M$</td>
<td>Computes the eigenvalues and eigenvectors of the given $d \times d$ matrix $M$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td>-all_rational_normal_forms</td>
<td>$d$</td>
<td>Produces a report of all rational normal forms of endomorphisms of $\mathbb{F}_q^d$</td>
</tr>
</tbody>
</table>

Table 3.9: Finite Field Activities for Linear Algebra

### 3.5 Linear Algebra Over Finite Fields

In Table 3.9, some finite field activities regarding linear algebra are shown. For instance, the command

```
RREF:
  ▶ $(\text{ORBITER_PATH})$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

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

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

nullspace:
> $(ORBITER_PATH)orbiter.out--v.2\n> -define:F2--finite_field--q.2--end\n> -define:v--vector--field:F2--format:2\n> -dense:"1,1,1,0,1,1,0,0,1"\n> -end\n> -with:F2--do\n> -finite_field_activity\n> -nullspace:v\n> -normalize_from_the_right\n> -end

computes the nullspace of the same matrix. The output is the matrix

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

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

eigenstuff:
> $(ORBITER_PATH)orbiter.out--v.6\n> -define:F--finite_field--q.5--end\n> -eigenstuff:F.4:"0,1,0,2,0,1,2,1,4,2,3,1,2,0,4,3".

computes all eigenvectors and eigenvalues of the matrix

\[
\begin{bmatrix}
0 & 1 & 0 & 2 \\
0 & 1 & 2 & 1 \\
4 & 2 & 3 & 1 \\
2 & 0 & 4 & 3
\end{bmatrix}
\]
over $\mathbb{F}_5$.

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

```
classes_GL_3_2:
  $\$(ORBITER_PATH)orbiter.out\$\cdot-v\cdot7\$
  $\$-define:F=finite_field=-q=2-\$\cdot-end\$
  $\$-all_rational_normal_forms=F=3\$
  $\$#pdflatex-Class_reps_GL_3_2.tex
  $\$#open-Class_reps_GL_3_2.pdf
```

produces a list of all conjugacy classes of $\text{GL}(3, 2)$. 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 [37]. This command relies on the Orbiter catalogue of irreducible polynomials. For an introduction to the rational normal form of endomorphisms, see [43].

**Conjugacy Classes of $\text{GL}(3, 2)$**

The number of conjugacy classes of $\text{GL}(3, 2)$ is 6:

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

Class 0 / 6

\[3, 1, 0\]

centralizer order 7

class size 24

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

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

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

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

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

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

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

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

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

centralizer order 168
class size 1
Chapter 4

Geometry

4.1 Finite Projective Spaces

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

Orbiter can create cheat sheets, which summarize the properties of $\text{PG}(n, q)$ and list the various elements. There are two different ways to create a cheat sheet for $\text{PG}(n, q)$. The first way is to create a finite field object and to then use the $\text{-cheat\_sheet\_PG}$ command inside a finite field activity. The second way is to create a projective space object and to use the $\text{-cheat\_sheet\_PG}$ command inside a projective space activity. Projective space objects contain information about the incidence structure and about the group. This can make them a bit more demanding in terms of time and memory. For large spaces, may be useful to create cheat sheets using finite field objects. Let us look at an example.

The following command creates a cheat sheet for $\text{PG}(2, 4)$ using a finite field object:

```plaintext
PG_2_4_cheat_sheet_F:
  $(\text{ORBITER\_PATH})\text{oribter.out}\cdot-v\cdot2\cdot$
  $\cdot-d\text{-define}\cdot-F\cdot-d\text{-finite\_field}\cdot-q\cdot4\cdot-end\cdot$
  $\cdot-d\text{-with}\cdot-F\cdot-d\text{-do}\cdot-d\text{-finite\_field\_activity}\cdot$
  $\cdot-d\text{-cheat\_sheet\_PG}\cdot2\cdot-end$
  pdf\text{latex}\cdot\text{PG}_2_4\cdot\text{tex}\cdot$
  open\cdot\text{PG}_2_4\cdot\text{pdf}$
```

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.
Figure 4.1: The plane PG(2, 4)

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:

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

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:

\[ P_0 = (1, 0, 0) \quad P_1 = (0, 1, 0) \quad P_2 = (0, 0, 1) \quad P_3 = (1, 1, 1) \quad P_4 = (1, 1, 0) \quad P_5 = (1, 3, 0) \]
\[ P_6 = (1, 2, 0) \quad P_7 = (1, 0, 1) \quad P_8 = (1, 0, 3) \quad P_9 = (1, 0, 2) \quad P_{10} = (0, 1, 1) \quad P_{11} = (1, 3, 3) \]
\[ P_12 = (1, 2, 2) \quad P_13 = (0, 1, 3) \quad P_14 = (1, 2, 1) \quad P_15 = (1, 1, 3) \quad P_{16} = (1, 3, 2) \quad P_{17} = (0, 1, 2) \]
\[ P_{18} = (1, 3, 1) \quad P_{19} = (1, 2, 3) \quad P_{20} = (1, 1, 2) \]

The Lines of PG(2, 4). PG(2, 4) has 21 1-subspaces:

\[
\begin{align*}
L_0 &= \begin{bmatrix} 100 \\ 010 \end{bmatrix} & L_7 &= \begin{bmatrix} 101 \\ 012 \end{bmatrix} & L_{14} &= \begin{bmatrix} 120 \\ 001 \end{bmatrix} \\
L_1 &= \begin{bmatrix} 100 \\ 011 \end{bmatrix} & L_8 &= \begin{bmatrix} 101 \\ 013 \end{bmatrix} & L_{15} &= \begin{bmatrix} 103 \\ 010 \end{bmatrix} \\
L_2 &= \begin{bmatrix} 100 \\ 012 \end{bmatrix} & L_9 &= \begin{bmatrix} 110 \\ 001 \end{bmatrix} & L_{16} &= \begin{bmatrix} 103 \\ 011 \end{bmatrix} \\
L_3 &= \begin{bmatrix} 100 \\ 013 \end{bmatrix} & L_{10} &= \begin{bmatrix} 102 \\ 010 \end{bmatrix} & L_{17} &= \begin{bmatrix} 103 \\ 012 \end{bmatrix} \\
L_4 &= \begin{bmatrix} 100 \\ 001 \end{bmatrix} & L_{11} &= \begin{bmatrix} 102 \\ 011 \end{bmatrix} & L_{18} &= \begin{bmatrix} 103 \\ 013 \end{bmatrix} \\
L_5 &= \begin{bmatrix} 101 \\ 010 \end{bmatrix} & L_{12} &= \begin{bmatrix} 102 \\ 012 \end{bmatrix} & L_{19} &= \begin{bmatrix} 130 \\ 001 \end{bmatrix} \\
L_6 &= \begin{bmatrix} 101 \\ 011 \end{bmatrix} & L_{13} &= \begin{bmatrix} 102 \\ 013 \end{bmatrix} & L_{20} &= \begin{bmatrix} 010 \\ 001 \end{bmatrix}
\end{align*}
\]

The next command creates a cheat sheet using a projective space object:

```
PG_2_4_cheat_sheet_P:
▷ $(ORBITER_PATH)orbiter.out -v 2 \n▷ ▷ -define_F:finite_fieldhq:4 -end:\n▷ ▷ -define_P:projective_space:2:q:4 -end:\n▷ ▷ -with_P:do_projective_space_activity:\n▷ ▷ ▷ -cheat_sheet:\n```

57
Here is a slightly larger example. The following command creates a cheat sheet for PG(3, 2).

```
PG_3_2:
▷ $(ORBITER_PATH)oribiter.out-\text{-v-0-}\
▷ \text{-define F-\text{-finite_field-}\text{-q-2-}\text{-end-}}\
▷ \text{-define P-\text{-projective_space-}3-F-\text{-end-}}\
▷ \text{-with P-\text{-do-}projective_space_activity-}\text{-cheat_sheet-}\text{-end}\n▷ pdflatex PG_3_2.tex\
▷ 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** 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:

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

Lines sorted by Plücker coordinates:

\[
0 = \text{Pl}(1, 0, 0, 0, 0, 0) = L_0 = \begin{bmatrix} 1000 \\
0100 \end{bmatrix} \\
1 = \text{Pl}(0, 1, 0, 0, 0, 0) = L_{34} = \begin{bmatrix} 0010 \\
0001 \end{bmatrix} \\
2 = \text{Pl}(0, 0, 1, 0, 0, 0) = L_4 = \begin{bmatrix} 1000 \\
0010 \end{bmatrix} \\
3 = \text{Pl}(0, 0, 0, 1, 0, 0) = L_{30} = \begin{bmatrix} 0100 \\
0001 \end{bmatrix} \\
4 = \text{Pl}(0, 0, 0, 0, 1, 0) = L_6 = \begin{bmatrix} 1000 \\
0001 \end{bmatrix} \\
\]
5 = P{l}(0, 0, 0, 0, 0, 1) = L_{28} = \begin{pmatrix} 0100 \\ 0010 \end{pmatrix}

\vdots

34 = P{l}(0, 1, 1, 1, 1, 1) = L_{26} = \begin{pmatrix} 1101 \\ 0011 \end{pmatrix}

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

\begin{align*}
L_0 &= \begin{pmatrix} 1000 \\ 0100 \end{pmatrix} = P{l}(1, 0, 0, 0, 0, 0) \\
L_4 &= \begin{pmatrix} 1000 \\ 0010 \end{pmatrix} = P{l}(0, 0, 1, 0, 0, 0) \\
L_6 &= \begin{pmatrix} 1000 \\ 0001 \end{pmatrix} = P{l}(0, 0, 0, 0, 1, 0) \\
L_{28} &= \begin{pmatrix} 0100 \\ 0010 \end{pmatrix} = P{l}(0, 0, 0, 0, 0, 1) \\
L_{30} &= \begin{pmatrix} 0100 \\ 0001 \end{pmatrix} = P{l}(0, 0, 0, 1, 0, 0) \\
L_{34} &= \begin{pmatrix} 0010 \\ 0001 \end{pmatrix} = P{l}(0, 1, 0, 0, 0, 0)
\end{align*}

The planes of PG(3, 2)

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

\begin{align*}
L_0 &= \begin{pmatrix} 1000 \\ 0100 \\ 0011 \end{pmatrix} \\
L_1 &= \begin{pmatrix} 1000 \\ 0100 \\ 0011 \end{pmatrix} \\
\vdots
\end{align*}

\begin{align*}
L_{14} &= \begin{pmatrix} 0100 \\ 0010 \\ 0001 \end{pmatrix}
\end{align*}
The polynomial rings associated with $\text{PG}(3, 2)$

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

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

Let us describe the two functions rank and unrank to perform the actual mappings between \(\text{PG}(n, q)\) and \(\mathbb{Z}_N\), where \(N = \theta_n(q)\). For this, assume that ranking and unranking functions have already been defined for the elements of the finite field \(\mathbb{F}_q\). Thus, we assume that for \(x \in \mathbb{F}_q\), \(\text{Rank}(\mathbb{F}_q, x)\) is a number \(b\) in \(\mathbb{Z}_q\). Also, for \(b \in \mathbb{Z}_q\), we assume that \(\text{Unrank}(\mathbb{F}_q, b)\) is the corresponding \(x \in \mathbb{F}_q\). So, we assume that \(\text{Rank}\) and \(\text{Unrank}\) 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 \(\text{PG}(2, 4)\):

\[
\text{PG\_2\_4\_rank\_point:}
\]

\[
\text{\$}\text{\(\text{ORBITER\_PATH}\)}\text{\text{\texttt{\textbackslash \text\{path\}\text\}}\text{\texttt{\textbackslash \text\{feature\}}}\text{\textbackslash \text\{letter\}\text\}}\text{\text\{-v.2\}}\text{\text\}\text{\}}}\text{\text\{letter\}\
\]

\[
\text{\text\{-define\:\text\{feature\}\:\text\{-finite\:\text\{field\}\:\text\{-q.4\}\:\text\{-end\}\}\}}}\text{\text\{letter\}\
\]

\[
\text{\text\{-with\:\text\{feature\}\:\text\{-do\:\text\{-finite\:\text\{field\}\:\text\{activity\}\}}}\text{\text\{letter\}}
\]

\[
\text{\text\{-rank\:\text\{feature\}\:\text\{in\:\text\{feature\}\:\text\{-PG\:\text\{letter\}\}::\text\{number\}\:\text\{-quote\}\:\text\{number\}\:\text\{-quote\}\:\text\{-end\}}}\text{\text\{letter\}}
\]

\[
\text{\text\{-end\}}}\text{\text\{letter\}\
\]

62
Algorithm 1 Rank

1: procedure Rank(vector : x, field : $\mathbb{F}_q$, int : n)
2:     assert x is a nonzero vector in $\mathbb{F}_q^n$.
3:     if x = e, then
4:         return i
5:     if x = 1 then
6:         return n
7:     $i \leftarrow \max\{j \in \mathbb{Z}_n \mid x_j \neq 0\}$
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 $a \geq \sum_{j=0}^{i-1} q^j$ then
15:        $a \leftarrow a - 1$
16:    $a \leftarrow a + n - i + \sum_{j=0}^{i-1} q^j$
17:    return a

\[
\begin{array}{|c|c|}
\hline
a = \text{Rank}(x) & x = \text{Unrank}(a) \\
\hline
0 & (1, 0, 0) \\
1 & (0, 1, 0) \\
2 & (0, 0, 1) \\
3 & (1, 1, 1) \\
4 & (1, 1, 0) \\
5 & (1, 0, 1) \\
6 & (0, 1, 1) \\
\hline
\end{array}
\]

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

1: procedure \texttt{Unrank}(int : \texttt{a}, field : \mathbb{F}_q, int : \texttt{n})
2: \hspace{1cm} \textbf{assert} \ a \in \mathbb{Z}_N \text{ where } N = \theta_{n-1}(q).
3: \hspace{1cm} \textbf{if} \ a < \texttt{n} \ \textbf{then}
4: \hspace{2cm} \textbf{return} \ e_a
5: \hspace{1cm} \texttt{a} \leftarrow \texttt{a} - \texttt{n}
6: \hspace{1cm} \textbf{if} \ a = 0 \ \textbf{then}
7: \hspace{2cm} \textbf{return} 1
8: \hspace{1cm} \texttt{a} \leftarrow \texttt{a} - 1
9: \hspace{1cm} \texttt{x} \leftarrow 0
10: \hspace{2cm} \textbf{for} \ i = 1, \ldots, \texttt{n} - 1 \ \textbf{do}
11: \hspace{3cm} \textbf{if} \ a \geq \sum_{j=1}^{i-1} q^j \ \textbf{then}
12: \hspace{4cm} \texttt{a} \leftarrow \texttt{a} - \sum_{j=1}^{i-1} q^j
13: \hspace{3cm} \textbf{else}
14: \hspace{4cm} \texttt{x}_i \leftarrow 1
15: \hspace{3cm} \textbf{break}
16: \hspace{2cm} \textbf{for} \ k = i + 1, \ldots, \texttt{n} - 1 \ \textbf{do}
17: \hspace{3cm} \texttt{x}_k \leftarrow 0
18: \hspace{2cm} \texttt{a} \leftarrow \texttt{a} + 1
19: \hspace{2cm} \textbf{if} \ i = \texttt{n} - 1 \ \textbf{and} \ a \geq \sum_{j=0}^{i-1} q^j \ \textbf{then}
20: \hspace{3cm} \texttt{a} \leftarrow \texttt{a} + 1
21: \hspace{2cm} \texttt{j} \leftarrow 0
22: \hspace{2cm} \textbf{while} \ a > 0 \ \textbf{do}
23: \hspace{3cm} \texttt{r} \leftarrow a \mod q
24: \hspace{3cm} \texttt{x}_j \leftarrow \texttt{Unrank}(\mathbb{F}_q, \texttt{r})
25: \hspace{3cm} \texttt{j} \leftarrow \texttt{j} + 1
26: \hspace{3cm} \texttt{a} \leftarrow (a - \texttt{r})/q
27: \hspace{2cm} \textbf{for} \ h = \texttt{j}, \ldots, \texttt{i} - 1 \ \textbf{do}
28: \hspace{3cm} \texttt{x}_h \leftarrow 0
29: \hspace{2cm} \textbf{return} \ \texttt{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\$
  >> -define:F:-finite\_field-q:8:-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:1:73:-end\$
  >> -draw\_matrix\$
  >> >> -input\_csv\_file:PG\_n2\_q8\_incidence\_matrix.csv\$
  >> >> -box\_width:20:-bit\_depth:8\$
  >> >> >> all\_one\_all\_one\$
  >> >> -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.

Orbiter can create various objects in projective space. To do so, we invoke the \texttt{-projective\_space\_activity} command followed by \texttt{define\_object}. After that, we specify the label under which the object will be stored. The object itself is encoded as a vector of integers. The commands shown in Tables 4.2 and 4.3 can be used to create various different types of objects. Modifier options as shown in Table 4.4 apply. For instance, the command sequence

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

```
$\$(ORBITER\_PATH)\$orbiter.out-\$v.2\$
  >> -define:F:-finite\_field-q:11:-end\$
  >> -define:P:-projective\_space-3:F:-end\$
  >> -with:P:-do:\$
  >> -projective\_space\_activity\$
  >> > -define\_object:EC\$
  >> >> -elliptic\_curve:1:3\$
  >> >> -end\$
  >> >> -end\$
  >> >> -with:EC:-do:-combinatorial\_object\_activity:-save\$
```

65
Figure 4.2: Incidence matrix of $PG(2, 8)$ in Orbiter ordering
<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-hyperoval</td>
<td></td>
<td>To create a hyperoval</td>
</tr>
<tr>
<td>-subiaco_oval</td>
<td>f_short</td>
<td>Create the Subiaco oval</td>
</tr>
<tr>
<td>-subiaco_hyperoval</td>
<td></td>
<td>Create the Subiaco hyperoval</td>
</tr>
<tr>
<td>-adelaide_hyperoval</td>
<td></td>
<td>Create the Adalaide hyperoval</td>
</tr>
<tr>
<td>-translation</td>
<td>i</td>
<td>Create the translation hyperoval with exponent i</td>
</tr>
<tr>
<td>-Segre</td>
<td></td>
<td>Create the Segre hyperoval</td>
</tr>
<tr>
<td>-Payne</td>
<td></td>
<td>Create the Payne hyperoval</td>
</tr>
<tr>
<td>-Cherowitzo</td>
<td></td>
<td>Create the Cherowitzo hyperoval</td>
</tr>
<tr>
<td>-OKeefe_Penttila</td>
<td></td>
<td>Create the O’Keefe, Penttila hyperoval</td>
</tr>
<tr>
<td>-BLT_database</td>
<td>k</td>
<td>Create the kth BLT-set of order q from the database (k = 0, 1, ...)</td>
</tr>
<tr>
<td>-ovoid</td>
<td></td>
<td>Create an ovoid</td>
</tr>
<tr>
<td>-Baer</td>
<td></td>
<td>Create the (standard) Baer subgeometry</td>
</tr>
<tr>
<td>-orthogonal</td>
<td>ε</td>
<td>Create the $Q^ε(n,q)$ quadric</td>
</tr>
<tr>
<td>-hermitian</td>
<td></td>
<td>Create the Hermitian variety given by $\sum_{i=0}^{n} X_i^{\sqrt{q}+1} = 0$</td>
</tr>
<tr>
<td>-cuspidal_cubic</td>
<td></td>
<td>Create the cuspidal cubic $(s^3, ts^2, t^3)$ in PG(2,q)</td>
</tr>
<tr>
<td>-twisted_cubic</td>
<td></td>
<td>Create a twisted cubic $(s^3, s^2t, st^2, t^3)$ in PG(3,q)</td>
</tr>
<tr>
<td>-elliptic_curve</td>
<td>a b</td>
<td>Create the elliptic curve $y^2 = x^3 + ax + b$</td>
</tr>
<tr>
<td>-ttp_construction_A</td>
<td></td>
<td>Create the twisted tensor product code of type A [7]</td>
</tr>
<tr>
<td>-ttp_construction_A_hyperoval</td>
<td></td>
<td>Create the twisted tensor product code of type A [7]</td>
</tr>
<tr>
<td>-ttp_construction_B</td>
<td></td>
<td>Create the twisted tensor product code of type B [7]</td>
</tr>
</tbody>
</table>

Table 4.2: Orbiter Objects (Part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-unital_XXq_YZq_ZYq</td>
<td></td>
<td>Create the unital with equation $XX^q + YZ^q + ZY^q = 0$</td>
</tr>
<tr>
<td>-desarguesian_line_spread_in_PG_3_q</td>
<td></td>
<td>Create the desarguesian line spread in $\text{PG}(3, q)$ as a set of 2-subspaces</td>
</tr>
<tr>
<td>-Buekenhout_Metz</td>
<td></td>
<td>Create the Buekenhout Metz unital</td>
</tr>
<tr>
<td>-Uab</td>
<td>a b</td>
<td>Create the Buekenhout Metz unital in the form of Barwick and Ebert [5]</td>
</tr>
<tr>
<td>-whole_space</td>
<td></td>
<td>Create the whole space</td>
</tr>
<tr>
<td>-hyperplane</td>
<td></td>
<td>Create the hyperplane given by dual coordinates associated with the given point</td>
</tr>
<tr>
<td>-segre_variety</td>
<td></td>
<td>Create the Segre variety</td>
</tr>
<tr>
<td>-Maruta_Hamada_arc</td>
<td></td>
<td>Create the Maruta Hamada arc</td>
</tr>
<tr>
<td>-projective Variety</td>
<td></td>
<td>Create the projective variety of degree $d$ with label $l$, with coefficient vector $C$, see Section 4.5.</td>
</tr>
<tr>
<td>-intersection_of_zariski_open_sets</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>-projective_curve</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 4.3: Orbiter Objects (Part 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-embedded_in_PG_4_q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-BLT_in_PG</td>
<td></td>
<td>Create the BLT-set with ranks in $\text{PG}(n, q)$ instead of orthogonal point ranks</td>
</tr>
<tr>
<td>-monomial_type_LEX</td>
<td></td>
<td>Monomials are in lexicographical ordering.</td>
</tr>
<tr>
<td>-monomial_type_PART</td>
<td></td>
<td>Monomials are in partition ordering.</td>
</tr>
</tbody>
</table>

Table 4.4: Orbiter Objects: Modifiers
creates the elliptic curve

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

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

`elliptic_curve_b1_c3_q11.txt`.

For a more general way to create projective varieties, see Section 4.5.
4.3 Finite Desarguesian Projective Planes

The projective spaces $\text{PG}(2, q)$ deserve special attention. They are examples of a more general structure called projective planes. The $\text{PG}(2, F)$, $F$ a field, are distinguished in the class of projective planes by the fact that the theorem of Desargues always holds. They are called the desarguesian projective planes. For other projective planes, see Section 10.9.

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

The command

```
PG_2.16:
```

produces the drawing of $\text{PG}(2, 16)$ 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 13.2 that the default values are 10000), which has the effect that the text appears smaller relative to the grid.

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

```
PG_2.4_with_decomposition:
```

The output is shown below:
Considering the cyclic group generated by
\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\omega & 1 & 1
\end{bmatrix}
\begin{bmatrix}
010 \\
001 \\
211
\end{bmatrix},
\]

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:
\[
\rightarrow \begin{array}{c|c}
21 \\
21 \\
\hline
5
\end{array}
\]
Column scheme:
Figure 4.4: Cyclic incidence matrix of PG(2, 4)

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

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

```bash
PG_2_4_incma_cyclic:
  $(ORBITER_PATH)orbiter.out -v 4 \
  -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.
Figure 4.5: Tactical decomposition of the incidence matrix of PG(2, 4)

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

```
PG_2_4_incma_singer_sub_3:
  @(ORBITER_PATH)orborler.out.-v.4.\n  -list_arguments.\n  -define R.-vector.-repeat.3.-7.-end.\n  -define C.-vector.-repeat.3.-7.-end.\n  -draw_matrix.\n  -input_csv_file PG_2_4_singer_incma_subgroup_index_3.csv.\n  -box_width.40.-bit_depth.24.\n  -partition.3-R-C.\n  -end
open PG_2_4_singer_incma_subgroup_index_3_draw.bmp
```

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

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

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

using the $q$-binomial coefficient.

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

The Orbiter cheat sheets for $\text{PG}(n,q)$ (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:

```
$\text{(ORBITER_PATH)}\text{orbiter.out}$
$\text{\texttt{-define F \text{-finite_field -q 2 -end}}}$
$\text{\texttt{-with F \text{-do}}}$
$\text{\texttt{-finite_field_activity}}$
$\text{\texttt{-cheat_sheet_Gr 3 2 -end}}$
```

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

$L_0 = \begin{bmatrix} 100 \\ 010 \end{bmatrix}, \quad L_3 = \begin{bmatrix} 101 \\ 010 \end{bmatrix}, \quad L_6 = \begin{bmatrix} 010 \\ 001 \end{bmatrix}$

$L_1 = \begin{bmatrix} 100 \\ 011 \end{bmatrix}, \quad L_4 = \begin{bmatrix} 101 \\ 011 \end{bmatrix}$

$L_2 = \begin{bmatrix} 100 \\ 001 \end{bmatrix}, \quad L_5 = \begin{bmatrix} 110 \\ 001 \end{bmatrix}$
Homogenizing yields -monomial_type_LEX then uses the lexicographic ordering as a tie breaker. The lexicographic ordering (use option -monomial_type_PART) is groupings 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 4.5 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 \( x^3 + 3y^3 - y^2Z + XZ = 0 \). 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 4.5, we record the coefficient vector of the equation as sequence
\[(1, 0, 3, 0, 0, 0, 10, 1, 0, 0).\]

The Orbiter command

```
EC_11.EQUATION="1,0,3,0,0,0,10,1,0,0"
```

creates the algebraic set associated to the cubic curve \(y^2 = x^3 + x + 3\) in \(\text{PG}(2, 11)\). 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 4.6 shows the Orbiter monomial orderings for degrees 2 and 3 in PG(3, q). Based on the partition ordering, the equation is coded as coefficient vector

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

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

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

**Hirschfeld\_surface\_q4.txt:**

- $(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 2 -v$
- $-\text{define}\ F: \text{-finite_field}\ -q\ 4\ -end\$
- $-\text{define}\ P: \text{-projective_space}\ 3\ F\ -end\$
- $-\text{with}\ P\ :-do\$
- $-\text{-projective\_space\_activity}\$
- $\text{define}\_object\_H4$
- $\text{-projective\_variety}: "\text{Hirschfeld\_surface\_q4}\"$

77
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:

```plaintext
del_Pezzo_F13ab_report:
  $(ORBITER_PATH)orbiter.out -v.3 -
  -define:F:finite_field:q.13 -
  -define:P:projective_space:3-F -
  -define:f3:formula:"del_Pezzo_F13a":"del\_Pezzo\_F13a":"x,y,z" -
  -define:f4:formula:"del_Pezzo_F13b":"del\_Pezzo\_F13b":"x,y,z" -
  -define:del_Pezzo13:collection:"f3,f4" -
  -with:P:do -
  -projective_space:activity -
  -analyze_del_Pezzo_surface:del_Pezzo13:"" -
  -end
  #pdflatex del_Pezzo_F13a_report.tex
  #pdflatex del_Pezzo_F13b_report.tex
  #open del_Pezzo_F13a_report.pdf
  #open del_Pezzo_F13b_report.pdf
```

The third argument after the `-formula` command specifies the managed variables, which are \( x, y, z \). The command `-collection` is used to group objects together. In this case, both surfaces are group together under new name. That way, we can issue the `-analyze_del_Pezzo_surface` once, and it applies to both surfaces.
4.6 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 $\mathfrak{G}_{r,2}(V)$, additional functionality is possible. The Plücker coordinates allow to identify $\mathfrak{G}_{r,2}(V)$ with the $Q^+(5,q)$ quadric.

The command

\begin{verbatim}
GR_4_2_2:
▷ $(ORBITER_PATH)orbiter.out -v 2 -
▷ define F:finite_field q 2 -end -
▷ with F:do:finite_field_activity -
▷ cheat_sheet_Gr_4_2_2 -
▷ pdflatex Gr_4_2_2.tex
▷ open Gr_4_2_2.pdf
\end{verbatim}

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

There are 35 lines:

\begin{align*}
L_0 &= \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} = \mathbf{Pl}(1,0,0,0,0,0) \\
L_1 &= \begin{bmatrix} 1000 \\ 0110 \end{bmatrix} = \mathbf{Pl}(1,0,1,0,0,0) \\
L_2 &= \begin{bmatrix} 1000 \\ 0101 \end{bmatrix} = \mathbf{Pl}(1,0,0,0,1,0) \\
\vdots \\
L_{34} &= \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} = \mathbf{Pl}(0,1,0,0,0,0)
\end{align*}

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):
\[ 0 = \mathbf{P}l(1, 0, 0, 0, 0) = L_0 = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \]
\[ 1 = \mathbf{P}l(0, 1, 0, 0, 0) = L_{34} = \begin{bmatrix} 0010 \\ 0001 \end{bmatrix} \]
\[ 2 = \mathbf{P}l(0, 0, 1, 0, 0, 0) = L_4 = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} \]
\[ \vdots \]
\[ 34 = \mathbf{P}l(0, 1, 1, 1, 1) = L_{26} = \begin{bmatrix} 1101 \\ 0011 \end{bmatrix} \]
4.7 Orthogonal Spaces

Orbiter can create and work with orthogonal spaces and their groups. An orthogonal space is created by a quadratic form. We assume that the form is nondegenerate. There are three types of nondegenerate quadratic forms in $\text{PG}(n,q)$. Two when $n$ is odd (hyperbolic and elliptic) and one if $n$ is even (parabolic). Basic information about these quadrics and their representative quadratic forms in Orbiter is given in Table 4.7. 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

```plaintext
-orthogonal_space $\epsilon$ $d$ $q$ -end
```

command can be used. Here, $d = n + 1$, $q$ is the order of the finite field, and

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

In order to create an object of type orthogonal space, the `-orthogonal_space` command is used inside a `-definition .. -end` command sequence. In Table 4.8, Orbiter command options for creating orthogonal spaces are shown.

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

```plaintext
Op_4.2:
     $(ORBITER_PATH)orbiter.out.-v.2\$
     > -define:F.--finite_field.--q.2.--end:
     > > -define:O.--orthogonal_space:1.4:F.--without_group.--end:
     > > > -with:O.--do--orthogonal_space_activity:
     > > > > -cheat_sheet_orthogonal.--end
     > pdflatex:0.1.4.2_report.tex:
     > open:0.1.4.2_report.pdf
```
The table below shows command options to create an orthogonal space:

<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 4.8: 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
$(ORBITER_PATH)orbiter.out -v.2 -
 -define:F:-finite_field:-q.2:-end -
 -define:O:-orthogonal_space:0.5:F:-without_group:-end -
 -with:O:-do:-orthogonal_space_activity -
 -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 -
 -all_one_r:all_one_c -
 -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 $PΓO(n+1, q)$. 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

```
-def 1de 0 -orthogonal_space 1 6 2 -end
```
creates an object $O$ of type $Q^+(5,2)$. In Table 4.9, Orbiter activities for orthogonal spaces are shown.

The command

```
Op_6_2:
> $(ORBITER_PATH)orbiter.out -v 2 \n> -define F:finit\\finite_field =q 2 =-end \n> -define O:orthogonal_space 1 6 F:without_group -end \n> -with O -do orthogonal_space_activity \n> -cheat_sheet_orthogonal -end
> pdflatex 0_1_6_2_report.tex
> open 0_1_6_2_report.pdf
```

produces a cheat sheet for the quadric $Q^+(5,2)$. This is the Klein quadric from Section 4.6. Orbiter produces the following output. At the top is the tactical decomposition of the incidence matrix between points and lines with respect to a hyperbolic pair. After that, the points and lines are listed (output shortened):
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cheat_sheet_orthogonal</td>
<td></td>
<td>Create a latex report of the orthogonal space. If the group has been created, the report will contain information about the group also.</td>
</tr>
<tr>
<td>-unrank_line_through_two_points</td>
<td>p1 p2</td>
<td>Determine the rank of the line through p1 and p2.</td>
</tr>
<tr>
<td>-perp</td>
<td>L</td>
<td>Determine the common perp of a set of points. The point ranks are given in the list L.</td>
</tr>
<tr>
<td>-create_BLT_set</td>
<td>descr</td>
<td>Creates a BLT-set of $Q(4,q)$. See Section 10.11.</td>
</tr>
</tbody>
</table>

Table 4.9: Activities related to orthogonal spaces

<table>
<thead>
<tr>
<th>$\rightarrow$</th>
<th>9</th>
<th>36</th>
<th>18</th>
<th>18</th>
<th>6</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>$\downarrow$</td>
<td>9</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The number of points is 35 points:

$P_0 = (1,0,0,0,0,0)$
$P_1 = (0,1,0,0,0,0)$
$P_2 = (0,0,1,0,0,0)$
$P_3 = (1,0,1,0,0,0)$
$P_4 = (0,1,1,0,0,0)$
$P_5 = (0,0,0,1,0,0)$
$P_6 = (1,0,0,1,0,0)$

84
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

\[
\text{Op}_6.64\text{.line}_\text{rank}:
\]

\[
\text{=$(ORBITER\_PATH)orbiter.out\,-v\,-l\,104}\]

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

\[
\text{Op}_6.64\text{.line}_\text{rank}:
\]

\[
\text{=$(ORBITER\_PATH)orbiter.out\,-v\,-l\,104}\]

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

\[
\text{Op}_6.64\text{.line}_\text{rank}:
\]

\[
\text{=$(ORBITER\_PATH)orbiter.out\,-v\,-l\,104}\]

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

\[
\text{Op}_6.64\text{.line}_\text{rank}:
\]

\[
\text{=$(ORBITER\_PATH)orbiter.out\,-v\,-l\,104}\]

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

\[
\text{Op}_6.64\text{.line}_\text{rank}:
\]

\[
\text{=$(ORBITER\_PATH)orbiter.out\,-v\,-l\,104}\]
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 $Q(5,64)$:

```
Op_6.64_report:
  $ $(ORBITER_PATH)orbiter.out -v 4 \n  $ -define:F:-finite_field:-q 64 -end \n  $ -define:O:-orthogonal_space:1 6 F:-without_group:-end \n  $ -with:O:-do:-orthogonal_space_activity \n  $ -unrank_line_through_two_points:15447347 15225451 \n  $ -end
```

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

<table>
<thead>
<tr>
<th></th>
<th>16769025</th>
<th>1090252800</th>
<th>532350</th>
<th>532350</th>
<th>130</th>
<th>4225</th>
<th>4225</th>
</tr>
</thead>
<tbody>
<tr>
<td>16511040</td>
<td>65</td>
<td>4160</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>266175</td>
<td>0</td>
<td>4096</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>266175</td>
<td>0</td>
<td>4096</td>
<td>0</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4225</td>
<td>3969</td>
<td>0</td>
<td>126</td>
<td>126</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4225</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4225</td>
<td></td>
</tr>
</tbody>
</table>

The group is not available.
The quadratic form is:

$$X_0 X_1 + X_2 X_3 + X_4 X_5 = 0$$
The number of points is 17047617
Too many points to print.
The number of lines is 1108095105

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

To study BLT-sets in $Q(4, q)$, see Section 10.11.
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:
▷ $(\text{ORBITER\_PATH})$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
```

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

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

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

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

The command

```
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
```

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

88
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 $\mathbb{F}_4$. 

89
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-transversal</td>
<td>$L_1 L_2$</td>
<td>Computes the intersection of two lines.</td>
</tr>
<tr>
<td>-intersection_of_two_lines</td>
<td>$L_1 L_2$</td>
<td>Computes the intersection of two lines.</td>
</tr>
<tr>
<td>-move_two_lines_in_hyperplane_stabilizer</td>
<td>$M_1 M_2 N_1 N_2$</td>
<td>Computes a projectivity of $\text{PG}(3,q)$ fixing the hyperplane $v(X_3)$ moving line $m_i$ to $n_i$ for $i = 1, 2$. It assumes that both $m_1, m_2$ and $n_1, n_2$ are skew. The lines $m_i$ and $n_i$ are given through Orbiter line numbers.</td>
</tr>
<tr>
<td>-move_two_lines_in_hyperplane_stabilizer_text</td>
<td>$M_1 M_2 N_1 N_2$</td>
<td>Like -move_two_lines_in_hyperplane_stabilizer, but now each line is given by a $2 \times 4$ generator matrix for the subspace.</td>
</tr>
<tr>
<td>-inverse_isomorphism_klein_quadric</td>
<td>$L_{36}$</td>
<td>Computes the orbiter point rank of the vector $L$ in $\text{PG}(n,q)$.</td>
</tr>
</tbody>
</table>

Table 4.10: Finite Field Activities

4.9 Advanced Topics

The orbiter commands related to finite projective spaces are grouped into three classes: finite field activities, projective space activities and otherwise. In Table 4.10, some commands regarding projective spaces over finite fields are shown. In Tables 4.11-4.12, the Orbiter commands associated with a projective space over a finite field are shown. Table 4.14 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

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

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mapsto N_2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The command

trans:
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-canonical_form_PG</td>
<td>Descr</td>
<td>Computes the canonical form and the automorphism group of the objects. See Section 12.2 for details.</td>
</tr>
<tr>
<td>-table_of_cubic_surfaces_compute_properties</td>
<td>fname q₀ col-offset</td>
<td>See Section 7.5.</td>
</tr>
<tr>
<td>-cubic_surface_properties_analyze</td>
<td>fname q₀</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></td>
</tr>
<tr>
<td>-define_object</td>
<td>label descr</td>
<td>To create a geometric object and add it to the symbol table under the given label. See this section.</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 4.11: 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 4.12: 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 10.8.</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><code>fname-mask N</code></td>
<td>Classify quartic curves using Nauty.</td>
</tr>
<tr>
<td><code>-classify_quartic_curves_with_substructure</code></td>
<td><code>fname-mask N k d frame</code></td>
<td>Classify quartic curves using substructure algorithm.</td>
</tr>
<tr>
<td><code>-set_stabilizer</code></td>
<td><code>k</code> <code>fname-mask N col-label</code></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><code>polarity</code></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 4.13: 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 $\text{PG}(2, q)$. Requires -control_arcs. See Section 6.6.</td>
</tr>
<tr>
<td>-control_arcs</td>
<td>description</td>
<td>Poset classification control for arcs used during the classification of cubic curves. See Table 6.2.</td>
</tr>
<tr>
<td>-create_points_on_quartic</td>
<td>$\epsilon$</td>
<td>Creates a table of points on a specific quartic curve. Consecutive points are no more than $\epsilon$ apart.</td>
</tr>
<tr>
<td>-create_points_on_parabola</td>
<td>$\epsilon$ $a$ $b$ $c$</td>
<td>Creates a table of points on the parabola $y = ax^2 + bx + c$. Consecutive points are no more than $\epsilon$ apart.</td>
</tr>
<tr>
<td>-smooth_curve</td>
<td>$\epsilon$ $N$ $b$ $t_{\text{min}}$ $t_{\text{max}}$ function</td>
<td>Creates at least $N$ points on a continuous curve given by “function”. Consecutive points are no more than $\epsilon$ apart. The function must be in terms of a parameter $t$. The values of $t$ are taken from the interval $[t_{\text{min}}, t_{\text{max}}]$.</td>
</tr>
<tr>
<td>-create_spread</td>
<td>description</td>
<td>Creates a spread according to the description. See Section 10.8.</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 4.14: Orbiter commands related to projective geometries
$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot5\cdot$

-define F\cdot finite\_field \cdot q\cdot16 \cdot end \cdot end\•

-define F\cdot do\cdot finite\_field\_activity\•

-move\_two\_lines\_in\_hyperplane\_stabilizer\_text\•

"1,0,0,0,0,0,0,1"\•"1,1,0,2,0,0,1,0"\•

"1,0,0,0,0,0,0,1"\•"0,1,0,1,0,0,1,0"\•

-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 $\delta^4 = \delta^3 + 1$. 
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. [32, 56]). 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 gnerators and the generators themselves. The labels will be used in reports about the group, for instance.

```
GEN_C13="1,2,3,4,5,6,7,8,9,10,11,12,0"
#(0,1,2,3,4,5,6,7,8,9,10,11,12)
```

```
C13:
  $(ORBITER_PATH)orbiter.out.-v.10:\
  · -define.gens--vector--dense$(GEN_C13).-end.\
  · -define.G--permutation_group.\
  · -bsgs-C13-C._13.13.0.1.\
  · gens.\
  · -end.\
  · -with.G--do.\
  · -group_theoretic_activity.\
  · -export_orbiter.\
  · -end.\
  · -with.G--do.\
  · -group_theoretic_activity.\
  · -report.\
  · -end.\
  · -with.G--do.\
  · -group_theoretic_activity.\
  · -save_elements_csv."C13_elts.csv".\
  · -end
```

The makefile variable `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}
\textbf{Stabilizer chain}
\begin{tabular}{|c|c|c|c|}
\hline
Level & Base pt & Orbit length & Subgroup order \\
\hline
0 & 0 & 13 & 13 \\
\hline
\end{tabular}
\end{center}

\begin{center}
\textbf{Basic Orbit 0}
\end{center}

\begin{itemize}
\item 0
\item 1
\item 2
\item 3
\item 4
\item 5
\item 6
\item 7
\item 8
\item 9
\item 10
\item 11
\item 12
\end{itemize}
Basic orbit 0 has size 13
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

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

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

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

Symmetric_4:
▷ $(ORBITER_PATH)orbiter.out-\text{-v.10}\$
▷ -define G-permutation_group-Symmetric_group4-end\$
▷ -with G-do\$
▷ -group_theoretic_activity\$
▷ -export_orbiter\$
▷ -end\$
▷ -with G-do\$
▷ -group_theoretic_activity\$
▷ -report\$
▷ -end\$
▷ -with G-do\$
▷ -group_theoretic_activity\$
▷ -save_elements_csv-Symmetric4_elts.csv\$
▷ -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**

- 1
- 2
- 3
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 PGL(4, 2) in the action on points. We use the following Orbiter command to create the group:

PGL\_4\_2\_export:
\begin{verbatim}
$\$(ORBITER\_PATH)orbiter\_out:-v.2\$
\end{verbatim}

The command invokes two activities. The first creates a latex report for the group in the file PGL\_4\_2\_report.tex. The second activity exports the permutation representation in Orbiter makefile format. The file PGL\_4\_2\_makefile is created:

PGL\_4\_2:
\begin{verbatim}
$\$(ORBITER\_PATH)orbiter\_out:-v.2\$
\end{verbatim}
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}:

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

END

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\_symmetric\_group.13}\$
  $\text{-subgroup\_by\_generators\_C13.13.1\$(\text{GEN\_C13})\_end}\$
  $\text{-with\_G\_do}\$
  $\text{-group\_theoretic\_activity}\$
  $\text{-export\_orbiter}\$
  $\text{-end}\$
  $\text{-with\_G\_do}\$
  $\text{-group\_theoretic\_activity}\$
  $\text{-report}\$
  $\text{-end}\$
  $\text{-with\_G\_do}\$
  $\text{-group\_theoretic\_activity}\$
  $\text{-save\_elements\_csv\"C13\_elts.csv\"}\$
  $\text{-end}\$
  \text{#pdflatex-Perm13\_Subgroup\_C13.13\_report.tex}
  \text{#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. [61]. 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 \( V \cong \mathbb{F}_q^n \) 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

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

is a generator for the cyclic group \( C_h \cong \text{Aut}(\mathbb{F}_q) \). 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

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

\[ M_k. \] (5.1)

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

\[ M_k \cdot N_l = \left( M \cdot N^{\Phi^{-k}} \right)_{k+l}, \] (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 semi-linear group \( A\Gamma L(n, q) \) is the semidirect product of \( AGL(n, q) \) with \( Aut(F_q) \). The elements of \( A\Gamma L(n, q) \) are triples

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

which act on \( F_q^n \):

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

The multiplication in \( A\Gamma L(n, q) \) 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 \( F_q \) is created as described in in two steps. In the first step, the field \( F_q \) is created as described in Sections 3.2 and 3.4. 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 $GL(d,q) \wr Sym(n)$</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>$\frac{q^{n-1} - 1}{q - 1}$</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

PGL₄.₂:

> $(\text{ORBITER\_PATH})\text{orbie}\text{r}\_\text{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 PGL₄.₂ report.tex
> \> open PGL₄.₂ report.pdf

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 $G = PGL(3,2)$ 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>$PΓO(n, q)$</td>
</tr>
<tr>
<td>-PGGOp</td>
<td>$n \ q$</td>
<td>$PΓO^+(n, q)$</td>
</tr>
<tr>
<td>-PGGOm</td>
<td>$n \ q$</td>
<td>$PΓO^-(n, q)$</td>
</tr>
<tr>
<td>-GL_d_q_wr_Sym_n</td>
<td>$d \ q \ n$</td>
<td>$GL(d, q) \wr Sym(n)$</td>
</tr>
</tbody>
</table>

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

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

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

The Action

Group action $\text{PGL}(4,2)$ of degree 15
We act on the following set:

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

The group is a matrix group.
The group acts on projective space $\text{PG}(3,2)$
$q = 2$
$p = 2$
$e = 1$
$n = 3$
Number of points = 15
Number of lines = 35
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</td>
<td>0</td>
<td>DNE</td>
<td>DNE</td>
<td>1</td>
<td>DNE</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>DNE</td>
</tr>
</tbody>
</table>

$\begin{array}{ccc}
+ & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0
\end{array}$

$\begin{array}{cc}
\cdot & 1 \\
1 & 1
\end{array}$

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

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:

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

$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,$
$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,$
$1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,$
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, 1,0,0,1],
            [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:

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

Group action $\text{PGO}(5,2)$ of degree 15
We act on the following set:

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

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

114
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>

$1^0 \equiv 1$

$1^1 \equiv 1$

Base and Stabilizer Chain

Group order 720

$t_0 = 15, 8, 3, 1, 1, 2,$

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

Strong generators for a group of order 720:

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

1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,1,
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,
1,0,0,0,0,0,1,0,0,0,0,1,1,1,0,1,1,0,1,0,0,0,0,1,
1,0,0,0,0,0,1,0,0,0,0,0,1,0,1,1,1,0,1,0,0,1,0,1,0,
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:
\begin{verbatim}
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)])
\end{verbatim}

Magma export:

Compact form:

Generators in compact permutation form are:
\begin{verbatim}
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
\end{verbatim}
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 $\text{PSp}(6, 2)$ can be created using the following command:

```
PSP_6_2:
  > $(ORBITER_PATH)orbiter.out -v.2
  > -define:F:-finite_field:-q:2:-end
  > -define:G:-linear_group:-PGL:6:F:-symplectic_group:-end
  > -with:G:-do
  > -group_theoretic_activity
  > -report
  > -end
  > pdflatex PGL_6_2_Sp_6_2_report.tex
  > open PGL_6_2_Sp_6_2_report.pdf
```

The group $\text{PGO}(7, 2)$, isomorphic to $\text{PSp}(6, 2)$, can be created using the following command:

```
PGO_7_2:
  > $(ORBITER_PATH)orbiter.out -v.2
  > -define:F:-finite_field:-q:2:-end
  > -define:G:-linear_group:-PGO:7:F:-end
  > -with:G:-do
  > -group_theoretic_activity
  > -report
  > -end
  > pdflatex PGO_7_2_report.tex
  > open PGO_7_2_report.pdf
```
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Janko1</td>
<td></td>
<td>first Janko group, needs PGL(7, 11)</td>
</tr>
<tr>
<td>-monomial</td>
<td></td>
<td>subgroup of monomial matrices</td>
</tr>
<tr>
<td>-diagonal</td>
<td></td>
<td>subgroup of diagonal matrices</td>
</tr>
<tr>
<td>-null_polarity_group</td>
<td></td>
<td>null polarity group</td>
</tr>
<tr>
<td>-symplectic_group</td>
<td></td>
<td>symplectic group</td>
</tr>
<tr>
<td>-singer</td>
<td>$k$</td>
<td>subgroup of index $k$ in the Singer cycle</td>
</tr>
<tr>
<td>-singer_and_frobenius</td>
<td>$k$</td>
<td>subgroup of index $k$ in the Singer cycle, extended by the Frobenius automorphism of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$</td>
</tr>
<tr>
<td>-borel_upper</td>
<td></td>
<td>Borel subgroup of upper triangular matrices</td>
</tr>
<tr>
<td>-borel_lower</td>
<td></td>
<td>Borel subgroup of lower triangular matrices</td>
</tr>
<tr>
<td>-identity_group</td>
<td></td>
<td>identity subgroup</td>
</tr>
<tr>
<td>-subgroup_from_file</td>
<td>$fl$</td>
<td>read subgroup from file $f$ and give it the label $l$</td>
</tr>
<tr>
<td>-orthogonal</td>
<td>$\epsilon$</td>
<td>orthogonal group $O(\epsilon(n, q))$, with $\epsilon \in {\pm 1}$ when $n$ is even</td>
</tr>
<tr>
<td>-subgroup_by_generators</td>
<td>$l, o, n, s_1 \ldots s_n$</td>
<td>Generate a subgroup from generators. The label “$l$” is used to denote the subgroup; $o$ is the order of the subgroup; $n$ is the number of generators and $s_1, \ldots, s_n$ are the generators for the subgroup in vector form.</td>
</tr>
</tbody>
</table>

Table 5.3: Commands for creating subgroups

### 5.3 Subgroups

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

J1:
```bash
$(ORBITER_PATH)orbiter.out -v -3

$define:G:linear_group:PGL:7:11:-Janko1:-end
$with:G:-do:
$group:theoretic:activity:
$do: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 -
  ▶   -define G: -linear_group -PGL:3:11: -singer:19: -end:
  ▶   -with G: -do:
  ▶   -group_theoretic_activity:
  ▶   -report:
  ▶   -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 $\frac{q^d - 1}{q - 1}$ 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 -
  ▶   -define G: -linear_group -PGL:3:11: -singer_and_frobenius:19: -end:
  ▶   -with G: -do:
  ▶   -group_theoretic_activity:
  ▶   -report:
  ▶   -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 -
  ▶   -define G: -linear_group -SL:2:3 -
```

121
creates the group. The command produces the list of group elements shown below.

<table>
<thead>
<tr>
<th>Element</th>
<th>Order</th>
<th>Matrix</th>
</tr>
</thead>
</table>
| 0 / 8   | 1     | \[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\]
|         |       | (0)(1)(2)(3)(4)(5)(6)(7)(8) |
| 1 / 8   | 4     | \[
\begin{bmatrix}
2 & 1 \\
1 & 1 \\
\end{bmatrix}
\]
|         |       | (0)(1, 5, 2, 7)(3, 4, 6, 8) |
| 2 / 8   | 2     | \[
\begin{bmatrix}
2 & 0 \\
0 & 2 \\
\end{bmatrix}
\]
|         |       | (0)(1, 2)(3, 6)(4, 8)(5, 7) |
| 3 / 8   | 4     | \[
\begin{bmatrix}
1 & 2 \\
2 & 2 \\
\end{bmatrix}
\]
|         |       | (0)(1, 7, 2, 5)(3, 8, 6, 4) |
Element 4 / 8 of order 4:

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

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

Element 5 / 8 of order 4:

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

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

Element 6 / 8 of order 4:

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

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

Element 7 / 8 of order 4:

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

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

The group table is created as csv file:

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

END

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

cube_group:

```
$ (ORBITER_PATH) orbiter.out -v 3
```

123
\begin{verbatim}
\define gens vector dense
\define G linear group GL 3 3
\subgroup by generators "cube"."48".3\end
\define gens
\end
\with G -do
\group theoretic activity
\print elements.tex
\report
\end
\pdflatex GL_3_3_Subgroup_cube_48_report.tex
\open GL_3_3_Subgroup_cube_48_report.pdf
\pdflatex GL_3_3_Subgroup_cube_48_elements.tex
\open GL_3_3_Subgroup_cube_48_elements.pdf
\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
\end

The tetrahedral subgroup can be created as well:

\begin{verbatim}
tetra_group:
\$(ORBITER\_PATH)orbi\_e\_out\_v\_3\end
\define G linear group GL 3 3
\subgroup by generators "tetra"."12".2\end
\define gens
\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
\end
\end

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

\begin{verbatim}
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\end
\end
\end
\end
\end
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="
-1,-1,-1,-1,0,0,0,0,  
0,0,0,1,0,0,0,0,  
1,0,0,0,0,0,0,0,  
0,0,1,0,0,0,0,0,  
0,1,0,1,1,0,0,0,  
0,0,0,0,0,1,0,0,  
0,0,0,0,0,1,0,0,  
0,0,0,0,0,0,1,0,  
-1,0,1,1,1,1,1,1,  
1,0,0,0,0,0,0,0,  
0,0,1,0,0,0,0,0,  
0,0,0,1,0,0,0,0,  
0,0,0,0,1,0,0,0,  
0,0,0,0,0,1,0,0,  
0,0,0,0,0,0,1,0,  
0,0,0,0,0,0,1,0"
```

Weyl_E8:
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 [64].

The electronic Atlas of finite simple groups [64] lists generators for $U_3(3)$ as $3 \times 3$ matrices over the field $\mathbb{F}_9$ using the following short Magma [13] 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 [59])

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

which results in

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

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

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

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

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

The command

```
finite_field -q 9 -override_polynomial "17" -end
```

127
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:\
  ▶  ▶  "1,6,4,5,0,6,8,5,1,1,\"
  ▶  ▶  "6,2,1,7,8,4,0,6,6,1,1,\"
  ▶  ▶  -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 $\text{Co}_3$. Following [58], 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. The following command also creates a report for the group.

```
CONWAY_GEN1="11011100010000010100001111011111010000101111\"
111101011111010000101111\"
000000100000010001010101\"
111110111011000100111110\"
010101000000001001110101\"
000001000000100010010101\"
001000000000100010010101\"
000100001100000011111111\"
11101010100110001000100111\"
00000000000001100010101\"
000000000001000100010101\"
```

128
CONWAY_GEN2="0101000010111010111111
0011001000111101011111
0011011100101101010111
1010010000100011111011
1101000000010101000111
1100101010011110101011
1000110100110101010101
0101100010100000001111
1100000101001010101001
0101110111100001001
0101111010100111100111
1000010101010101010001
0001010000111000111111
0011010011101110111011
0100110011101110111010
1101011101110111011101
0101101010010100100001
1100001010001010000111
0000000011111100011111"
Finally, we create 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 $\mathbb{F}_{27}$:

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

Ree_27:
```bash
$(ORBITER_PATH)orbiter.out -v 6 -
> define F: finite_field -q 27 - override_polynomial "34" -
> define Ree_gen1: vector - field F - format 7 - dense $(Ree_gen1) -
> define Ree_gen2: vector - field F - format 7 - dense $(Ree_gen2) -
> define G: linear_group - PGL 7 - F -
> subgroup_by_generators "Ree_27" . "1007344472" . 2 -
> Ree_gen1 Ree_gen2 -
> - end -
> - with G - do -
> - group_theoretic_activity -
> - report -
> - end
```
### Table 5.4: Commands for creating new actions from old

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

### 5.5 Induced Actions

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

```plaintext
T3_on_tensors:
  ▶ $(ORBITER_PATH)orbiter.out -v 4 \n  ▶ ▶ -define G \n  ▶ ▶ -linear_group -GL_d_q_wreath_Sym_n_2_2_3 -on_tensors -end \n  ▶ ▶ -with G -do \n  ▶ ▶ -group_theoretic_activity \n  ▶ ▶ ▶ -report \n  ▶ ▶ -end \n  ▶ pdflatex GL_2_2_wreath_Sym3_report.tex \n  ▶ open GL_2_2_wreath_Sym3_report.pdf
```

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

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.

\[
\begin{align*}
\left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \right), & \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right); \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); & \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); \\
\left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); & \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right); & \quad \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right);
\end{align*}
\]

0,1,2,1,0,0,1,1,0,0,1,1,0,1,1,1,0,0,1,1,0,1,1,1,0,0,1,1,0,1,1,1,0,0,1,1,0,1,1,1,0,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,
Stabilizer chain

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

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

```latex
\begin{verbatim}
T3r1:
  \$ (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 \$
  \$ -end \$
  \$ pdfflatex GL_2_2_wreath_Sym3_report.tex \$
  \$ open GL_2_2_wreath_Sym3_report.pdf \$
\end{verbatim}
```

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

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

The order of the group \( \text{GL(2, 2)} \Join \text{Sym(3)} \) is 1296

The group acts on a set of size 27

The Action

Group action \( \text{GL(2, 2)} \Join \text{Sym(3)res27} \) 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 $\text{PGL}(2, 8)$ of $\text{PG}(1, 8)$ and act on $\text{PG}(2, 8)$:

```
PGGL_2_8_on_conic:
  ▷ $(\text{ORBITER_PATH})\text{orbiter.out} -v \cdot 4 \cdot$
  ▷  ▷  -define G:
  ▷  ▷  -linear_group -PGGL_2_8 -PGL2OnConic -end:
  ▷  ▷  -with G -do:
  ▷  ▷  -group_theoretic_activity:
  ▷  ▷  -report:
  ▷  ▷  -end
  ▷  pdflatex PGGL_2_8_OnConic_2_8_report.tex
  ▷  open PGGL_2_8_OnConic_2_8_report.pdf
```

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

### The Group $\text{PGL}(2, 8)\text{OnConic}(2, 8)$

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

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

### The Action

Group action $\text{PGL}(2, 8)\text{OnConic}$ of degree 73.
We act on the following set:
\[ 0 = (1, 0, 0) \quad 5 = (2, 1, 0) \]
\[ 1 = (0, 1, 0) \quad : \]
\[ 2 = (0, 0, 1) \quad 72 = (7, 7, 1) \]
\[ 3 = (1, 1, 1) \]
\[ 4 = (1, 1, 0) \]

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</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</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( \alpha = \gamma )</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>( \alpha + 1 = \gamma^5 )</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>( \alpha^2 = \gamma^2 )</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>( \alpha^2 + 1 = \gamma^3 )</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>( \alpha^2 + \alpha = \gamma^6 )</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>( \alpha^2 + \alpha + 1 = \gamma^4 )</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>DNE</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
\[ + \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 1 & 0 & 3 & 2 & 5 & 4 & 7 & 6 \\
2 & 2 & 3 & 0 & 1 & 6 & 7 & 4 & 5 \\
3 & 3 & 2 & 1 & 0 & 7 & 6 & 5 & 4 \\
4 & 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\
5 & 5 & 4 & 7 & 6 & 1 & 0 & 3 & 2 \\
6 & 6 & 7 & 4 & 5 & 2 & 3 & 0 & 1 \\
7 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
\end{array} \]

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

\[
\begin{align*}
2^0 &= 1 \\
2^1 &= 2 \\
2^2 &= 4 \\
2^3 &= 5 \\
2^4 &= 7 \\
2^5 &= 3 \\
2^6 &= 6 \\
2^7 &= 1 \\
\end{align*}
\]

**Base and Stabilizer Chain**

Group order 1512  
\( tl=9, 8, 7, 3, \)

**Stabilizer chain**

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

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

Basic Orbit 1

Basic orbit 1 has size 8
1, 2, 3, 4, 5, 6, 7, 8
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 -group_theoretic_activity option. Tables 5.5 and 5.6 list the possible commands that can come after it.

The command

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

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

The command

```
PGL_3_4_singer:
  $ (ORBITER_PATH) orbiter.out -v 5 -
  -define G:-linear_group -PGL:3:4 -end -
  -with G:-do -
  -group_theoretic_activity -
  -find_singer_cycle -
  -end
```

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 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 [25].</td>
</tr>
<tr>
<td>-export_magma</td>
<td></td>
<td>Exports the group to Magma [13].</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></td>
</tr>
<tr>
<td>-group_table</td>
<td></td>
<td>Stores the group table as csv-file.</td>
</tr>
<tr>
<td>-report</td>
<td></td>
<td>Produce a latex report about the group.</td>
</tr>
<tr>
<td>-sylow</td>
<td></td>
<td>Include Sylow subgroups in the report (requires -report).</td>
</tr>
<tr>
<td>-print_elements</td>
<td></td>
<td>Produces a printout of all group elements.</td>
</tr>
<tr>
<td>-print_elements_tex</td>
<td></td>
<td>Produces a latex report of all group elements.</td>
</tr>
<tr>
<td>-order_of_products</td>
<td>$g_1 \ldots g_n$</td>
<td>Creates a table of the orders of all products $g_ig_j$, $1 \leq i, j \leq n$.</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_isomorphism_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)

```latex
GL_{2,8}\cdot multiply:
\begin{itemize}
  \item \$\$(\text{ORBITER\_PATH})\text{orbiter.out}\cdot -v\cdot 5\cdot \$
  \item \$\$ -\text{define}\cdot G\cdot -\text{linear}\cdot group\cdot -\text{GL}\cdot 2\cdot 8\cdot -\text{end}\cdot \$
  \item \$\$ -\text{with}\cdot G\cdot -do\cdot \$
  \item \$\$ -\text{group}\cdot \text{theoretic}\cdot activity\cdot \$
  \item \$\$ -\text{multiply}\cdot "0,1,2,3"\cdot "4,5,6,7"\cdot \$
  \item \$\$ -\text{end}\$
  \item pdf\text{latex}\cdot GL_{2,8}\cdot mult\.tex
  \item open\cdot GL_{2,8}\cdot mult\.pdf
\end{itemize}
```

The output is

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

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:

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

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

We can compute the inverse of a group element:

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

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

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

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 [13], 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
  -define G
  -linear_group -PGGL_2_4
  -end
  -with G --do
  -group_theoretic_activity
  -classes
  -end
$(MAGMA_PATH)/magma-PGGL_2_4_classes.magma
$(ORBITER_PATH)orbiter.out --v=3
  -define G
  -linear_group -PGGL_2_4
  -end
  -with G --do
  -group_theoretic_activity
  -classes
  -end
```

145
computes the classes of elements in \( \text{PGL}(2, 4) \) using Orbiter-Magma-Orbiter. The first Orbiter command produces the file \( \text{PGGL}_2_4\_\text{classes}.\text{magma} \). The magma command reads this file and produces the file \( \text{PGGL}_2_4\_\text{classes}.\text{out}.\text{txt} \). The second Orbiter command reads the file \( \text{PGGL}_2_4\_\text{classes}.\text{out}.\text{txt} \) and produces the latex report \( \text{PGGL}_2_4\_\text{classes}.\text{out}.\text{tex} \).

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  
\[
c_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_1
\]

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:  
\[
\left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]_1, \left[ \begin{array}{cc} \omega^2 & 0 \\ 0 & 1 \end{array} \right]_1, \left[ \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right]_1
\]

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

The command sequence

```
PGGL_2_4_cent_2A:  
- pdflatex:PGGL_2_4_classes_out.tex  
- open:PGGL_2_4_classes_out.pdf
```

```
- $(\text{ORBITER\_PATH})\text{orbiter.out}.\text{-v.3.}$\backslash
  - -define:G:\backslash
  - -linear_group:-PGGL_2_4:-end:\backslash
  - -with:G:-do:\backslash
  - -group_theoretic_activity:\backslash
  - -centralizer_of_element:"2A":"1,0,0,1,1":\backslash
```
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 PGL(4,2).Z₂. Orbiter produces a list of strong generators, shown below:

<table>
<thead>
<tr>
<th>Strong generators for a group of order 40320:</th>
</tr>
</thead>
<tbody>
<tr>
<td>\begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 1 &amp; 0 \ 1 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 1 &amp; 0 &amp; 0 &amp; 1 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 0 &amp; 1 \ 1 &amp; 1 &amp; 1 &amp; 0 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 1 &amp; 0 \ 1 &amp; 0 &amp; 1 &amp; 0 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 1 &amp; 0 \ 1 &amp; 0 &amp; 1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 1 &amp; 1 \ 1 &amp; 0 &amp; 1 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \end{bmatrix}, \begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 1 &amp; 1 \ 1 &amp; 1 &amp; 1 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 1 \end{bmatrix}</td>
</tr>
</tbody>
</table>

147
The end of the report has a list of generators in coded form. This list can be used to create the centralizer in Orbiter.

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

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

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

\[
\text{PGL}_4\text{.5\_norm\_31:} \\
\text{\indent \indent \indent $(\text{ORBITER\_PATH})\text{orbiter.out} -v -6 -\text{define\_G} -$} \\
\text{\indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \indent \input{normalizer_of_31_in_PGL_4_5.pdf}
Strong generators for a group of order 372:

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

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 \(Z_{22}\) in \(\text{PGL}(2,9)\):

```
$ (ORBITER_PATH)orbiter.out -v 2 \
  \$(' -define G:-linear_group -PGL:2:9 \
  \$(' -subgroup_byGenerators Z_{22}:2:2:0,0,1:0,1,0,-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 $\text{PGL}(2,9)\text{SubgroupZ22order4}$ 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 [15, 32, 56]. 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 12.2.

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)orbiF.0t.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
▷ pdflatex:GL_2_2_wreath_Sym3_res27_orbits.tex
▷ open:GL_2_2_wreath_Sym3_res27_orbits.pdf
```

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

Table 6.1: Basic Orbit algorithms
This command computes the orbits of on all cubic forms in 4 variables, confirming the work of Dickson [20] and an enumerative result of Cooley [17].

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:

\texttt{PGGL\_2\_8\_on\_conic\_orbits:}
\begin{verbatim}
  \$ (ORBITER\_PATH) orbiter.out -v 4 \\
  -define G -linear_group -PGGL\_2\_8 -PGL\_2\_0 Conic -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
\end{verbatim}

Figure 6.1: The Schreier tree for the action on rank-one tensors

\begin{verbatim}
 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
\end{verbatim}
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 PGL(2, 8)OnConic:

Strong generators for a group of order 1512:

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

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,
17, 22, 44, 35, 23, 46, 40, 51, 28, 8, 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:

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

1,0,0,1,1,
1,0,0,6,0,
1,0,2,1,0,
0,1,1,0,0,
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:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \gamma^5 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \gamma^3 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix}$$

1,0,0,1,1,
1,0,3,1,2,
1,0,5,1,0,
1,0,2,1,0,
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 [51]. 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 on layer to the next. This uses the incidence relation between adjacent layers. By iterating this method, one can form a poset of substructures, and the classification problem reduces to the problem of determining the orbits of the poset, which we will henceforth call the poset classification problem. Many classification problem in Combinatorics reduce to poset classification problems.

Orbiter uses the algorithm of Schmalz [55] 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.
Figure 6.3: Subspace lattice of $V(3, 2)$

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

```
-preferred_choice n a b
```

can be used to force Orbiter to choose $b$ instead of $a$ at node $n$. 

160
<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 13.2.</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 Ain.</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_</td>
<td></td>
<td>Produce a bitmap drawing of the neighboring relations in the poset with flag orbits.</td>
</tr>
<tr>
<td>with_flag_orbits</td>
<td></td>
<td></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_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>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_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>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:

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

The join of two subsets in the poset may or may not belong to the poset. Let us consider the action of the Singer cycle on \( \text{PG}(3, 2) \). The following command computes the orbits of the group \( G \) generated by it:

\[
PGL_{3,2}_\text{singer}:
\]

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

- \text{orbiter_path}$(\text{ORBITER\_PATH})$

- \text{-define G:-linear_group-PGL.3.2.-singer.1.-end}$

- \text{-with G:-do}$

- \text{-group\_theoretic\_activity}$

- \text{-poset\_classification\_control}$

- \text{-problem_label-PGL.3.2.singer.1.-W.-depth.7}$

- \text{-draw\_poset}$

- \text{-report.-end}$

- \text{-end}$

- \text{pdflatex PGL.3.2_singer.1_poset.tex}$

- \text{open-PGL.3.2.singer.1_poset.pdf}$
\]

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

\[
\text{PG.3.7_lines}:
\]

\[
\text{- $(\text{ORBITER\_PATH})\text{orbiter.out}$-v.7-

- \text{orbiter_path}$(\text{ORBITER\_PATH})$

- \text{-define F:-finite_field-q.7.-end}$

- \text{-define G:-linear_group-PGL.4.F.-on_k_subspaces.2.-end}$

- \text{-with G:-do}$

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

```
PGO_5_2_on_subsets:
  $(\text{ORBITER\_PATH})\text{orbiter}\_\text{out}\.\text{-v}\_3\$
  $(\text{ORBITER\_PATH})\text{-orbin\_path}$
  $(\text{ORBITER\_PATH})\text{-define}\_F\_\text{-finite\_field}\_q\_2\_\text{-end}\$
  $(\text{ORBITER\_PATH})\text{-define}\_G\_\text{-linear\_group}\_\text{-PGO}\_5\_F\_\text{-end}\$
  $(\text{ORBITER\_PATH})\text{-with}\_G\_\text{-do}\$
  $(\text{ORBITER\_PATH})\text{-group\_theoretic\_activity}\$
  $(\text{ORBITER\_PATH})\text{-poset\_classification\_control}\$
  $(\text{ORBITER\_PATH})\text{-problem\_label}\_\text{-PGO}\_5\_2\$
  $(\text{ORBITER\_PATH})\text{-depth}\_15\$
  $(\text{ORBITER\_PATH})\text{-report}\_\text{-end}\$
  $(\text{ORBITER\_PATH})\text{-draw}\_\text{poset}\$
  $(\text{ORBITER\_PATH})\text{-w}\$
  $(\text{ORBITER\_PATH})\text{-end}\$
  $(\text{ORBITER\_PATH})\text{-orbits\_on\_subsets}\_15\$
  $(\text{ORBITER\_PATH})\text{-report}\$
  $(\text{ORBITER\_PATH})\text{-end}\$
  $(\text{ORBITER\_PATH})\text{pdflatex}\_\text{PGO}\_5\_2\_\text{poset}\_\text{tex}\$
  $(\text{ORBITER\_PATH})\text{open}\_\text{PGO}\_5\_2\_\text{poset}\_\text{pdf}\$
```

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 $PGO^+(4, 2)$ of order 72. The command

\[
\text{subspaces}_\text{Op}_{4,2}:
\]

\[
\begin{align*}
&\textbf{ orbs } \text{ orbs .out: -v .5 } \text{ or. outr.out: -v .5. } \\
&\text{ orbiter .out: -v .5. } \\
&\text{ orbiter .path: orbs .out: -v .5. } \\
&\text{ orbiter .path: orbs .out: -v .5. } \\
&\text{ orbiter .path: orbs .out: -v .5. } \\
&\text{ orbiter .path: orbs .out: -v .5. } \\
&\text{ orbiter .path: orbs .out: -v .5. } \\
&\text{ orbiter .path: orbs .out: -v .5. }
\end{align*}
\]
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}
\texttt{-group_theoretic_activity}\
\texttt{-poset_classification_control}\
\texttt{-node_label_is_element}\
\texttt{-draw_poset\_draw_options\_radius\_200\_end}\
\texttt{-problem_label\_Op\_4\_2\_W\_depth\_4}\
\texttt{-report\_end}\
\texttt{-end}\
\texttt{-orbits_on_subspaces\_4}\
\texttt{-report}\
\texttt{-end}
\end{verbatim}

\texttt{pdflatex\_PGL\_4\_2\_Orthogonal\_plus\_4\_2\_poset.tex}
\texttt{open\_PGL\_4\_2\_Orthogonal\_plus\_4\_2\_poset.pdf}

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 6.4: Commands for Classifying Arcs

<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-q</code></td>
<td><code>q</code></td>
<td>Specify the size of the field $\mathbb{F}_q$.</td>
</tr>
<tr>
<td><code>-d</code></td>
<td><code>d</code></td>
<td>Require that no more than $d$ points lie on a line.</td>
</tr>
<tr>
<td><code>-n</code></td>
<td><code>n</code></td>
<td>The size of the matrix group.</td>
</tr>
<tr>
<td><code>-target_size</code></td>
<td><code>t</code></td>
<td>Specify the size of the arc to be $t$.</td>
</tr>
<tr>
<td><code>-conic_test</code></td>
<td></td>
<td>Require that no 6 points of the arc lie on a conic.</td>
</tr>
<tr>
<td><code>-affine</code></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 than $d$ point lie on a line applies to affine lines only.</td>
</tr>
<tr>
<td><code>-no_arc_testing</code></td>
<td></td>
<td>Do not test the at most $d$ points per line condition.</td>
</tr>
<tr>
<td><code>-forbidden_point_set</code></td>
<td><code>set</code></td>
<td>The arc must not contain any of the given points.</td>
</tr>
</tbody>
</table>

#### 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. [44]) and the Pellegrino cap in $\text{AG}(4,3)$. The uniqueness of this cap was proven by Hill [27].

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

```bash
hyperoval_16:
  ▷ $(ORBITER_PATH) orbiter.out -v 4 \n  ▷ ▷ -orbiter_path $(ORBITER_PATH) \n  ▷ ▷ -define F -finite_field -q 16 -end \n  ▷ ▷ -define P -projective_space 2 F -end \n```
performs the classification of hyperovals in 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):

![Orbit output](image)

### Orbits at Level 18

There are 2 orbits at level 18.

### Orbit 0 / 2 at Level 18

Node number: 4212

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

\[
\begin{align*}
0 : 0 & = ( 1, 0, 0 ) \\
1 : 1 & = ( 0, 1, 0 ) \\
2 : 2 & = ( 0, 0, 1 ) \\
3 : 3 & = ( 1, 1, 1 ) \\
4 : 52 & = ( 3, 2, 1 ) \\
5 : 67 & = ( 2, 3, 1 ) \\
6 : 89 & = ( 8, 4, 1 ) \\
7 : 106 & = ( 9, 5, 1 ) \\
8 : 126 & = ( 13, 6, 1 ) \\
9 : 141 & = ( 12, 7, 1 ) \\
10 : 159 & = ( 14, 8, 1 ) \\
11 : 176 & = ( 15, 9, 1 ) \\
12 : 184 & = ( 7, 10, 1 ) \\
13 : 199 & = ( 6, 11, 1 ) \\
14 : 220 & = ( 11, 12, 1 ) \\
15 : 235 & = ( 10, 13, 1 ) \\
16 : 245 & = ( 4, 14, 1 ) \\
17 : 262 & = ( 5, 15, 1 )
\end{align*}
\]
Strong generators for a group of order 144:

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\delta^4 & \delta^9 & 1
\end{pmatrix},
\begin{pmatrix}
1 & \delta^7 & \delta^{13} \\
\delta^8 & \delta^9 & \delta^{10} \\
\delta & \delta^6 & 1
\end{pmatrix},
\begin{pmatrix}
\delta^5 & \delta^5 & \delta^5 \\
\delta^5 & \delta^2 & \delta^{11} \\
\delta^5 & \delta^{14} & 1
\end{pmatrix}_{10}
\]

1,0,0,0,1,0,9,5,1,1,
1,7,6,14,5,10,2,15,1,3,
1,1,1,3,5,1,5,1,0,0,
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 )
1 : 1 = ( 0, 1, 0 )
2 : 2 = ( 0, 0, 1 )
3 : 3 = ( 1, 1, 1 )
4 : 52 = ( 3, 2, 1 )
5 : 70 = ( 5, 3, 1 )
6 : 83 = ( 2, 4, 1 )
7 : 109 = ( 12, 5, 1 )
8 : 127 = ( 14, 6, 1 )
9 : 139 = ( 10, 7, 1 )

Strong generators for a group of order 16320:

\[
\begin{pmatrix}
\delta^6 & 0 & 0 \\
0 & \delta^3 & 0 \\
0 & 0 & 1
\end{pmatrix}_{2},
\begin{pmatrix}
\delta^9 & 0 & 0 \\
0 & \delta^7 & 0 \\
0 & 0 & 1
\end{pmatrix}_{1},
\begin{pmatrix}
\delta^2 & 0 & 0 \\
0 & \delta^{11} & 0 \\
\delta^4 & \delta^7 & 1
\end{pmatrix}_{3},
\begin{pmatrix}
\delta^{10} & 0 & 0 \\
0 & \delta^3 & 0 \\
\delta & \delta^{11} & 1
\end{pmatrix}_{2},
\begin{pmatrix}
\delta & 0 & 0 \\
\delta^{12} & \delta^2 & \delta^5 \\
\delta^{14} & \delta^{10} & 1
\end{pmatrix}_{1},
\begin{pmatrix}
\delta^5 & 0 & 0 \\
\delta^5 & \delta^3 & \delta^6 \\
\delta^6 & \delta^8 & 1
\end{pmatrix}_{0},
\begin{pmatrix}
\delta^{12} & 1 & \delta^2 \\
\delta^4 & \delta^3 & \delta^7 \\
\delta^6 & \delta^5 & 1
\end{pmatrix}_{2},
\begin{pmatrix}
\delta^{11} & \delta^6 & \delta^{10} \\
\delta^{10} & \delta^6 & 1
\end{pmatrix}_{3}
\]
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:

\begin{verbatim}
nc_arcs_16:
\> $(ORBITER_PATH)orbiter.out -v 4 -
\> -define:F:-finite_field-q:16:-end-
\> -define:P:-projective_space-2:F:-end-
\> -with:P:-do-
\> -projective_space_activity-
\> -classify_arcs-
\> -poset_classification_control-
\> -problem_label:nc_arcs_q16_d2:-W:-depth:6-
\> -report:-end-
\> -end-
\> -target_size:6-
\> -d:2-
\> -conic_test-
\> -end-
\> -end
\> pdflatex:nc_arcs_q16_d2.poset.tex
\> open:nc_arcs_q16_d2.poset.pdf
\end{verbatim}

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 $\mathbb{F}_{32}$:

\begin{verbatim}
nc_arcs_32_E13:
\> $(ORBITER_PATH)orbiter.out -v 4 -
\end{verbatim}
\texttt{-orbiner\_path\$\{ORBITER\_PATH\}\$
\texttt{-define F:\-finite\_field:q\_32:-end\$
\texttt{-define P:\-projective\_space:2:F:\-end\$
\texttt{-with P:-do\$
\texttt{-projective\_space\_activity\$
\texttt{-classify\_arcs\$
\texttt{-poset\_classification\_control\$
\texttt{-problem\_label:nc\_arcs\_q32\_d2:-W:-depth:6\$
\texttt{-draw\_poset:-draw\_options:-end\$
\texttt{-report:-end\$
\texttt{-end\$
\texttt{-target\_size:6\$
\texttt{-test nb\_Eckardt\_points:13\$
\texttt{-d:2\$
\texttt{-conic\_test\$
\texttt{-end\$
\texttt{-end
\texttt{pdflatex:nc\_arcs\_q32\_d2\_poset.tex
\texttt{open:nc\_arcs\_q32\_d2\_poset.pdf}
6.6 Cubic Curves

Orbiter can classify cubic curves in \( \text{PG}(2,q) \). To this end, the \((9,3)\)-arcs in \( \text{PG}(2,q) \) 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:

\[
\begin{array}{l}
\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-\text{-v-3-define-G-\}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-define-F-\text{-finite\_field-\text{-q-8-\text{-end}}}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-define-P-\text{-projective\_space-2-F-\text{-end}}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-with-P-\text{-do-}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-classify_cubic_curves-\text{-q-8-\text{-target\_size-9-\text{-n-3-\text{-d-3\}}}}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-poset\_classification\_control-\text{-problem\_label.cc-8-\text{-W-\text{-depth-9}}}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-draw\_options-\text{-radius-200-\text{-embedded-\text{-end}}}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-recognize-\"0,1,2,3,35,28\"}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-recognize-\"1,2,3,51,28,61,46,71,40\"}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-Kramer_Mesner\_matrix.6-9-}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-Kramer-Mesner\_matrix.6-9.}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-end-}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\$\{ORBITER\_PATH\}orbiter.out-\text{-v-2-\text{-draw\_matrix}}-\text{-end}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\-input\_csv\_file.cc-8_KM_6.9.csv-\text{-end}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\pdflatex\text{-Cubic\_curves.q8.tex}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\open\text{-Cubic\_curves.q8.pdf}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\#pdflatex\text{-cc-8\_tree\_lvl\_9.tex}}}
\end{array}
\]
\[
\begin{array}{l}
\text{\texttt{\#open\text{-cc-8\_tree\_lvl\_9.pdf}}}
\end{array}
\]

classifies the cubic curves in \( \text{PG}(2,8) \).
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:
  ▷ $(ORBITER_PATH)orbiter.out::-v:3::
  ▷  -define:F:finite_field:-q:4:-end::
  ▷  -define:P:projective_space:3:F:-end::
  ▷  -with:P:-do::
  ▷    -projective_space_activity::
  ▷  ▷  -define_surface:S:-q:4:-catalogue:0:-end::
  ▷  ▷  -end::
  ▷  -with:S:-do::
  ▷    -cubic_surface_activity::
  ▷  ▷  -report::
  ▷  ▷  ▷  -report_with_group::
  ▷  ▷  -end
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-catalogue</td>
<td>$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 [11] (where it is called the Hilbert, Cohn-Vossen surface). The equation is $X_3^3 - b^2(X_0^2 + X_1^2 + X_2^2)X_3 + \frac{b^2}{a}(a^2 + 1)X_0X_1X_2 = 0$. 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_</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>abcd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-arc_lifting</td>
<td>$A$</td>
<td>Create the surface associated with the arc $A = a_1, \ldots, a_6$ in $\text{PG}(2,q)$ 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><code>-arc_lifting_with_two_lines</code></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><code>-select_double_six</code></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><code>-transform</code></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><code>-transform_inverse</code></td>
<td>$A$</td>
<td>Transform the surface by the inverse projectivity (or collineation) defined by $A$. This option can be repeated.</td>
</tr>
</tbody>
</table>

Table 7.2: Commands to create a known cubic surface (Part 2)
Two reports are created, one with and the other without information about the group.

Another way of creating surfaces is as members of known infinite families. For instance,

Eckardt.13:

```
$\$(ORBITER\_PATH)\$\text{oriter.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.13.-q.13.-family_Eckardt.3.1.-end.}
\text{-end.}
\text{-with S.13.-do.}
\text{-cubic_surface_activity.}
```

```
\text{-report.}
\text{-report_with_group.}
\text{-end}
```

```
pdflatex\text{-family_Eckardt_q13_a3_b1_with_group.tex}
```

```
open\text{-family_Eckardt_q13_a3_b1_with_group.pdf}
```

creates the member of the Eckardt family described in [11] with parameters \((a, b) = (3, 1)\) over the field \(F_{13}\).

Let us try the 4-parameter normal form of cubic surfaces with four parameters \(a, b, c, d\). The formula can be encoded as makefile variable:

\[
F_{abcd} = -(a*b*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*(b - d)*X_0*X_0*X_2 \\
+ (a*b*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*(a - b)*X_0*X_1*X_2 \\
+ (a*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*(b - d)*X_0*X_1*X_3 \\
+ (a*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*(b - d)*X_0*X_2*X_2 \\
+ (a*b*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*(b - d)*X_0*X_2*X_3 \\
+ (a*b*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*X_1*X_1*X_2 \\
+ (a*b*c - a*b*d - a*c*d + b*c*d + a*d - b*c)*X_1*X_1*X_3 \\
+ (1 + 1) * a*b*c*d - a*a*b*b*d - (1 + 1) * a*a*c*d \\
- (1 + 1) * a*b*b*c*c + a*b*b*c*d + (1 + 1) * a*b*c*c*d + a*b*c*d \\
- b*b*c*c*d - a*a*b*c + a*a*c*d + a*a*d + a*b*b*c + a*b*c*c \\
- (1 + 1 + 1) * a*b*c*d + a*b*c*d + a*b*c*c) * X_1*X_2*X_3 \\
+ c*a*(a*d - b*c - a + b + c - d)*(b - d)*X_1*X_3*X_3
\]

\(178\)
The following command parses the formula and creates the surface with \((a,b,c,d) = (4,2,2,4)\):

```
F_abcd:
\[\begin{array}{l}
> $(ORBITER_PATH)oriter.out-v.3\backslash
> \quad -define F_finite_field-q.7\backslash
> \quad -end\backslash
> \quad -with F\backslash
> \quad -finite_field_activity\backslash
> \quad -parse_and_evaluate "Fabcd"."X0,X1,X2,X3"\backslash
> \quad \quad $(F_abcd_eqn)"a=4,b=2,c=2,d=4"\backslash
> \quad -end
\end{array}
```

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_q7_override_group:
\[\begin{array}{l}
> $(ORBITER_PATH)oriter.out-v.3\backslash
> \quad -define F_finite_field-q.7\backslash
> \quad -define P_projective_space-3-F\backslash
> \quad -with P\backslash
> \quad -define_surface F_2345-q.7\backslash
> \quad \quad -by_equation "F_alpha_beta_gamma_delta"\backslash
> \quad \quad "DF{"\alpha,\beta,\gamma,\delta}D"."x0,x1,x2,x3"\backslash
> \quad \quad $(F_ALPHA_BETA_GAMMA_DELTA)\backslash
> \quad \quad "\alphad=2,\beta=3,\gamma=4,\delta=5"\backslash
> \quad \quad "\D\\alpha=2,\beta=3,\gamma=4,\delta=5\D"\backslash
> \quad \quad -override_group 6\backslash
> \quad \quad 1,5,0,0,3,6,0,0,1,1,3,0,5,5,0,3,\backslash
> \quad \quad 1,0,2,5,0,1,6,1,0,0,3,5,0,0,4,4\backslash
> \quad \quad -end
\end{array}
```

```
pdflatex surface_equation_F_alpha_beta_gamma_delta_q7_report.tex
```

```
open surface_equation_F_alpha_beta_gamma_delta_q7_report.pdf
```

```
pdflatex surface_equation_F_alpha_beta_gamma_delta_q7_with_group.tex
```

```
open surface_equation_F_alpha_beta_gamma_delta_q7_with_group.pdf
```

179
7.2 Quartic Curves

Cubic surfaces with 27 lines are associated with quartic curves with 28 bitangents (see [28]), 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::
> $(ORBITER_PATH)orbiter.out-v.3-
> -define:F:-finite_field:-q:9:-end-
> -define:P:-projective_space:2:F:-end-
> -with:P:-do-
> -projective_space_activity-
> -define_quartic_curve:C:-q:9-
> -catalogue:0:-end-
> -end-
> -with:C:-do-
> -quartic_curve_activity-
> -report-
> -end
> 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:

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

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

**The gradient**

The gradient of the quartic curve is:

$$\alpha^7 X_1^3 + \alpha^2 X_2^3$$

$$(0, 4, 7, 0, 0, 0, 0, 0, 0, 0)$$

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

\[ (2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0) \]

General information

<table>
<thead>
<tr>
<th>Number of bitangents</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of points</td>
<td>28</td>
</tr>
<tr>
<td>Fullness</td>
<td>is full</td>
</tr>
<tr>
<td>Number of Kovalevski points</td>
<td>63</td>
</tr>
<tr>
<td>Bitangent line type ((a_0, a_1, a_2))</td>
<td>((0, 28, 0))</td>
</tr>
<tr>
<td>Number of singular points</td>
<td>0</td>
</tr>
</tbody>
</table>

All points on the curve

The surface has 28 points:

The points on the quartic curve are:

\[
\begin{align*}
0 : P_0 &= (1, 0, 0) \\
1 : P_1 &= (0, 1, 0) \\
2 : P_2 &= (0, 0, 1) \\
3 : P_3 &= (1, 1, 1) \\
4 : P_4 &= (1, 1, 0) \\
5 : P_5 &= (2, 1, 0) \\
6 : P_{14} &= (3, 0, 1) \\
7 : P_{17} &= (6, 0, 1) \\
8 : P_{24} &= (5, 1, 1) \\
9 : P_{25} &= (6, 1, 1) \\
10 : P_{30} &= (2, 2, 1) \\
11 : P_{32} &= (4, 2, 1) \\
12 : P_{34} &= (6, 2, 1) \\
13 : P_{38} &= (1, 3, 1) \\
14 : P_{41} &= (4, 3, 1) \\
15 : P_{44} &= (7, 3, 1) \\
16 : P_{46} &= (0, 4, 1) \\
17 : P_{51} &= (5, 4, 1) \\
18 : P_{53} &= (7, 4, 1) \\
19 : P_{57} &= (2, 5, 1) \\
20 : P_{58} &= (3, 5, 1) \\
21 : P_{62} &= (7, 5, 1) \\
22 : P_{76} &= (3, 7, 1) \\
23 : P_{77} &= (4, 7, 1) \\
24 : P_{78} &= (5, 7, 1) \\
25 : P_{82} &= (0, 8, 1) \\
26 : P_{83} &= (1, 8, 1) \\
27 : P_{84} &= (2, 8, 1) 
\end{align*}
\]

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:

\[
\begin{align*}
0 : P_7 &= (4, 1, 0) = c_{13} \cap c_{14} \cap c_{36} \cap c_{46} \\
1 : P_8 &= (5, 1, 0) = a_2 \cap a_4 \cap c_{25} \cap c_{45} \\
2 : P_9 &= (6, 1, 0) = b_1 \cap b_6 \cap c_{12} \cap c_{26} \\
3 : P_{10} &= (7, 1, 0) = a_3 \cap b_5 \cap c_{35} \cap d \\
4 : P_{11} &= (8, 1, 0) = b_2 \cap b_3 \cap c_{24} \cap c_{34} \\
5 : P_{12} &= (1, 0, 1) = a_3 \cap a_4 \cap c_{23} \cap c_{24} 
\end{align*}
\]
6 : \( P_{13} = (2, 0, 1) = c_{34} \cap c_{36} \cap c_{45} \cap c_{56} \)
7 : \( P_{15} = (4, 0, 1) = b_3 \cap b_6 \cap c_{13} \cap c_{16} \)
8 : \( P_{16} = (5, 0, 1) = a_5 \cap a_6 \cap c_{25} \cap c_{26} \)
9 : \( P_{18} = (7, 0, 1) = a_2 \cap b_1 \cap c_{35} \cap c_{46} \)
10 : \( P_{19} = (8, 0, 1) = b_4 \cap b_5 \cap c_{14} \cap c_{15} \)
11 : \( P_{20} = (0, 1, 1) = a_2 \cap b_3 \cap c_{14} \cap c_{56} \)
12 : \( P_{21} = (2, 1, 1) = b_2 \cap b_4 \cap c_{26} \cap c_{46} \)
13 : \( P_{22} = (3, 1, 1) = a_4 \cap b_5 \cap c_{12} \cap c_{36} \)
14 : \( P_{23} = (4, 1, 1) = a_6 \cap b_1 \cap c_{23} \cap c_{45} \)
15 : \( P_{26} = (7, 1, 1) = c_{16} \cap c_{25} \cap c_{34} \cap d \)
16 : \( P_{27} = (8, 1, 1) = a_3 \cap a_5 \cap c_{13} \cap c_{15} \)
17 : \( P_{28} = (0, 2, 1) = c_{12} \cap c_{13} \cap c_{25} \cap c_{35} \)
18 : \( P_{29} = (1, 2, 1) = b_1 \cap b_5 \cap c_{16} \cap c_{56} \)
19 : \( P_{31} = (3, 2, 1) = a_3 \cap a_6 \cap c_{34} \cap c_{46} \)
20 : \( P_{33} = (5, 2, 1) = a_2 \cap b_4 \cap c_{24} \cap d \)
21 : \( P_{35} = (7, 2, 1) = b_2 \cap b_6 \cap c_{23} \cap c_{36} \)
22 : \( P_{36} = (8, 2, 1) = a_4 \cap b_3 \cap c_{15} \cap c_{26} \)
23 : \( P_{37} = (0, 3, 1) = a_5 \cap b_1 \cap c_{24} \cap c_{36} \)
24 : \( P_{39} = (2, 3, 1) = a_2 \cap a_6 \cap c_{12} \cap c_{16} \)
25 : \( P_{40} = (3, 3, 1) = b_3 \cap b_4 \cap c_{35} \cap c_{45} \)
26 : \( P_{42} = (5, 3, 1) = a_4 \cap b_2 \cap c_{13} \cap c_{56} \)
27 : \( P_{43} = (6, 3, 1) = a_3 \cap b_6 \cap c_{14} \cap c_{25} \)
28 : \( P_{45} = (8, 3, 1) = c_{15} \cap c_{23} \cap c_{46} \cap d \)
29 : \( P_{47} = (1, 4, 1) = a_6 \cap b_2 \cap c_{14} \cap c_{35} \)
30 : \( P_{48} = (2, 4, 1) = b_3 \cap b_5 \cap c_{23} \cap c_{25} \)
31 : \( P_{49} = (3, 4, 1) = a_5 \cap b_6 \cap c_{56} \cap d \)
32 : \( P_{50} = (4, 4, 1) = a_2 \cap a_3 \cap c_{26} \cap c_{36} \)
33 : \( P_{52} = (6, 4, 1) = b_1 \cap b_4 \cap c_{13} \cap c_{34} \)
34 : \( P_{54} = (8, 4, 1) = c_{12} \cap c_{15} \cap c_{24} \cap c_{45} \)
35 : \( P_{55} = (0, 5, 1) = a_4 \cap a_6 \cap b_4 \cap b_6 \)
36 : \( P_{56} = (1, 5, 1) = c_{13} \cap c_{26} \cap c_{45} \cap d \)
37 : \( P_{59} = (4, 5, 1) = c_{24} \cap c_{25} \cap c_{46} \cap c_{56} \)
38 : \( P_{60} = (5, 5, 1) = c_{12} \cap c_{14} \cap c_{23} \cap c_{34} \)
39 : \( P_{61} = (6, 5, 1) = a_2 \cap a_5 \cap b_2 \cap b_5 \)
40 : \( P_{63} = (8, 5, 1) = c_{15} \cap c_{16} \cap c_{35} \cap c_{36} \)
41 : \( P_{64} = (0, 6, 1) = a_1 \cap b_5 \cap c_{26} \cap c_{34} \)
42 : \( P_{65} = (1, 6, 1) = a_1 \cap b_4 \cap c_{25} \cap c_{36} \)
43 : \( P_{66} = (2, 6, 1) = a_1 \cap b_6 \cap c_{24} \cap c_{35} \)
44 : \( P_{67} = (3, 6, 1) = a_1 \cap a_2 \cap c_{13} \cap c_{23} \)
45 : \( P_{68} = (4, 6, 1) = a_1 \cap b_2 \cap c_{12} \cap d \)
46 : \( P_{69} = (5, 6, 1) = a_1 \cap a_3 \cap b_1 \cap b_3 \)
47 : \( P_{70} = (6, 6, 1) = a_1 \cap a_4 \cap c_{16} \cap c_{46} \)
48 : \( P_{71} = (7, 6, 1) = a_1 \cap a_5 \cap c_{14} \cap c_{45} \)
49 : \( P_{72} = (8, 6, 1) = a_1 \cap a_6 \cap c_{15} \cap c_{56} \)
The points off the curve are: 
\[22, 23, 26, 27, 28, 29, 31, 33, 35, 36, 37, 39, 40, 42, 43, 45, 47, 48, 49, 50, 52, 54, 55, 56, 59, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 79, 80, 81, 85, 86, 87, 88, 89, 90, 6\]

The points off the curve are:

\[0 : P_6 = (3, 1, 0) \]
\[1 : P_7 = (4, 1, 0) \]
\[2 : P_8 = (5, 1, 0) \]
\[3 : P_9 = (6, 1, 0) \]
\[4 : P_{10} = (7, 1, 0) \]
\[5 : P_{11} = (8, 1, 0) \]
\[6 : P_{12} = (1, 0, 1) \]
\[7 : P_{13} = (2, 0, 1) \]
\[8 : P_{15} = (4, 0, 1) \]
\[9 : P_{16} = (5, 0, 1) \]
\[10 : P_{18} = (7, 0, 1) \]
\[11 : P_{19} = (8, 0, 1) \]
\[12 : P_{20} = (0, 1, 1) \]
\[13 : P_{21} = (2, 1, 1) \]
\[14 : P_{22} = (3, 1, 1) \]
\[15 : P_{23} = (4, 1, 1) \]
\[16 : P_{26} = (7, 1, 1) \]
\[17 : P_{27} = (8, 1, 1) \]
\[18 : P_{28} = (0, 2, 1) \]
\[19 : P_{29} = (1, 2, 1) \]
\[20 : P_{31} = (3, 2, 1) \]
\[21 : P_{33} = (5, 2, 1) \]
The lines and their points of contact are:

\[ a_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha^3 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 8 \end{bmatrix} \quad P_0 = P(1,0,0) 4\times \]

\[ a_2 = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 1 \end{bmatrix} \quad P_{83} = P(1,8,1) 4\times \]

\[ a_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^6 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 5 \end{bmatrix} \quad P_{57} = P(2,5,1) 4\times \]

\[ a_4 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^2 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 7 \end{bmatrix} \quad P_{53} = P(7,4,1) 4\times \]

\[ a_5 = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^7 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 4 \end{bmatrix} \quad P_{30} = P(2,2,1) 4\times \]

\[ a_6 = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^6 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 7 \\ 0 & 1 & 7 \end{bmatrix} \quad P_5 = P(2,1,0) 4\times \]

\[ b_1 = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^7 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \end{bmatrix} \quad P_{58} = P(3,5,1) 4\times \]

\[ b_2 = \begin{bmatrix} 1 & 0 & \alpha^7 \\ 0 & 1 & \alpha^6 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 5 \end{bmatrix} \quad P_{14} = P(3,0,1) 4\times \]

\[ b_3 = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \end{bmatrix} \quad P_{62} = P(7,5,1) 4\times \]

\[ b_4 = \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & \alpha^2 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 7 \end{bmatrix} \quad P_{77} = P(4,7,1) 4\times \]

\[ b_5 = \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & \alpha^3 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 8 \end{bmatrix} \quad P_{41} = P(4,3,1) 4\times \]

\[ b_6 = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^2 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 7 \end{bmatrix} \quad P_3 = P(1,1,1) 4\times \]

\[ c_{12} = \begin{bmatrix} 1 & 0 & \alpha^3 \\ 0 & 1 & \alpha^4 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 8 \\ 0 & 1 & 2 \end{bmatrix} \quad P_{17} = P(6,0,1) 4\times \]

\[ c_{13} = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^4 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \end{bmatrix} \quad P_{84} = P(2,8,1) 4\times \]

\[ c_{14} = \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 1 \end{bmatrix} \quad P_{32} = P(4,2,1) 4\times \]
\[ c_{15} = \begin{bmatrix} 1 & 0 & \alpha^5 \\ 0 & 1 & 0 \end{bmatrix} \]
\[ c_{16} = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha^6 \end{bmatrix} \]
\[ c_{23} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^5 \end{bmatrix} \]
\[ c_{24} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^7 \end{bmatrix} \]
\[ c_{25} = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^4 \end{bmatrix} \]
\[ c_{26} = \begin{bmatrix} 1 & 0 & \alpha^2 \\ 0 & 1 & \alpha^3 \end{bmatrix} \]
\[ c_{34} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^3 \end{bmatrix} \]
\[ c_{35} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha^4 \end{bmatrix} \]
\[ c_{36} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^7 \end{bmatrix} \]
\[ c_{45} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & \alpha^6 \end{bmatrix} \]
\[ c_{46} = \begin{bmatrix} 1 & 0 & \alpha^6 \\ 0 & 1 & \alpha \end{bmatrix} \]
\[ c_{56} = \begin{bmatrix} 1 & 0 & \alpha^4 \\ 0 & 1 & 1 \end{bmatrix} \]
\[ d = \begin{bmatrix} 1 & \alpha^6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\[ P_1 = \mathbf{P}(0, 1, 0) 4 \times \]
\[ P_{51} = \mathbf{P}(5, 4, 1) 4 \times \]
\[ P_{82} = \mathbf{P}(0, 8, 1) 4 \times \]
\[ P_{25} = \mathbf{P}(6, 1, 1) 4 \times \]
\[ P_{76} = \mathbf{P}(3, 7, 1) 4 \times \]
\[ P_{44} = \mathbf{P}(7, 3, 1) 4 \times \]
\[ P_{38} = \mathbf{P}(1, 3, 1) 4 \times \]
\[ P_{24} = \mathbf{P}(5, 1, 1) 4 \times \]
\[ P_{78} = \mathbf{P}(5, 7, 1) 4 \times \]
\[ P_{34} = \mathbf{P}(6, 2, 1) 4 \times \]
\[ P_{46} = \mathbf{P}(0, 4, 1) 4 \times \]
\[ P_4 = \mathbf{P}(1, 1, 0) 4 \times \]
\[ P_2 = \mathbf{P}(0, 0, 1) 4 \times \]

Rank of lines: ( 8, 51, 15, 17, 74, 77, 54, 45, 31, 67, 68, 37, 82, 32, 61, 60, 35, 16, 14, 72, 78, 28, 52, 24, 25, 53, 21, 59 )

Line type: 1^{28}

Point types: 1^{28}
point types for points off the curve: \(4^{63}\)

<table>
<thead>
<tr>
<th>28</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10,</td>
<td></td>
</tr>
<tr>
<td>11, 12, 13, 14, 15, 16, 17, 18, 19, 20,</td>
<td></td>
</tr>
<tr>
<td>21, 22, 23, 24, 25, 26, 27, 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>63</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10,</td>
<td></td>
</tr>
<tr>
<td>11, 12, 13, 14, 15, 16, 17, 18, 19, 20,</td>
<td></td>
</tr>
<tr>
<td>21, 22, 23, 24, 25, 26, 27, 28, 29, 30,</td>
<td></td>
</tr>
<tr>
<td>31, 32, 33, 34, 35, 36, 37, 38, 39, 40,</td>
<td></td>
</tr>
<tr>
<td>41, 42, 43, 44, 45, 46, 47, 48, 49, 50,</td>
<td></td>
</tr>
<tr>
<td>51, 52, 53, 54, 55, 56, 57, 58, 59, 60,</td>
<td></td>
</tr>
<tr>
<td>61, 62, 0</td>
<td></td>
</tr>
</tbody>
</table>

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 = \(P_7 = (4, 1, 0)\) lies on 4 bisecants : \{ 13, 14, 23, 25 \}
Off point 2 = \(P_8 = (5, 1, 0)\) lies on 4 bisecants : \{ 1, 3, 19, 24 \}
Off point 3 = \(P_9 = (6, 1, 0)\) lies on 4 bisecants : \{ 6, 11, 12, 20 \}
Off point 4 = \(P_{10} = (7, 1, 0)\) lies on 4 bisecants : \{ 2, 10, 22, 27 \}
Off point 5 = \(P_{11} = (8, 1, 0)\) lies on 4 bisecants : \{ 7, 8, 18, 21 \}
Off point 6 = \(P_{12} = (1, 0, 1)\) lies on 4 bisecants : \{ 2, 3, 17, 18 \}
Off point 7 = \(P_{13} = (2, 0, 1)\) lies on 4 bisecants : \{ 21, 23, 24, 26 \}
Off point 8 = \(P_{15} = (4, 0, 1)\) 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 \}
Off point 29 = \( P_{42} = (5, 3, 1) \) lies on 4 bisecants : \{ 3, 7, 13, 26 \}
Off point 28 = \( P_{43} = (6, 3, 1) \) lies on 4 bisecants : \{ 2, 11, 14, 19 \}
Off point 29 = \( P_{45} = (8, 3, 1) \) lies on 4 bisecants : \{ 15, 17, 25, 27 \}
Off point 30 = \( P_{47} = (1, 4, 1) \) lies on 4 bisecants : \{ 5, 7, 14, 22 \}
Off point 31 = \( P_{48} = (2, 4, 1) \) lies on 4 bisecants : \{ 8, 10, 17, 19 \}
Off point 32 = \( P_{49} = (3, 4, 1) \) lies on 4 bisecants : \{ 4, 11, 26, 27 \}
Off point 33 = \( P_{50} = (4, 4, 1) \) lies on 4 bisecants : \{ 1, 2, 20, 23 \}
Off point 34 = \( P_{52} = (6, 4, 1) \) lies on 4 bisecants : \{ 6, 9, 13, 21 \}
Off point 35 = \( P_{54} = (8, 4, 1) \) lies on 4 bisecants : \{ 12, 15, 18, 24 \}
Off point 36 = \( P_{55} = (0, 5, 1) \) lies on 4 bisecants : \{ 3, 5, 9, 11 \}
Off point 37 = \( P_{56} = (1, 5, 1) \) lies on 4 bisecants : \{ 13, 20, 24, 27 \}
Off point 38 = \( P_{59} = (4, 5, 1) \) lies on 4 bisecants : \{ 18, 19, 25, 26 \}
Off point 39 = \( P_{60} = (5, 5, 1) \) lies on 4 bisecants : \{ 12, 14, 17, 21 \}
Off point 40 = \( P_{61} = (6, 5, 1) \) lies on 4 bisecants : \{ 1, 4, 7, 10 \}
Off point 41 = \( P_{63} = (8, 5, 1) \) lies on 4 bisecants : \{ 15, 16, 22, 23 \}
Off point 42 = \( P_{64} = (0, 6, 1) \) lies on 4 bisecants : \{ 0, 10, 20, 21 \}
Off point 43 = \( P_{65} = (1, 6, 1) \) lies on 4 bisecants : \{ 0, 9, 19, 23 \}
Off point 44 = \( P_{66} = (2, 6, 1) \) lies on 4 bisecants : \{ 0, 11, 18, 22 \}
Off point 45 = \( P_{67} = (3, 6, 1) \) lies on 4 bisecants : \{ 0, 1, 13, 17 \}
Off point 46 = \( P_{68} = (4, 6, 1) \) lies on 4 bisecants : \{ 0, 7, 12, 27 \}
Off point 47 = \( P_{69} = (5, 6, 1) \) lies on 4 bisecants : \{ 0, 2, 6, 8 \}
Off point 48 = \( P_{70} = (6, 6, 1) \) lies on 4 bisecants : \{ 0, 3, 16, 25 \}
Off point 49 = \( P_{71} = (7, 6, 1) \) lies on 4 bisecants : \{ 0, 4, 14, 24 \}
Off point 50 = \( P_{72} = (8, 6, 1) \) lies on 4 bisecants : \{ 0, 5, 15, 26 \}
Off point 51 = \( P_{73} = (0, 7, 1) \) lies on 4 bisecants : \{ 2, 7, 16, 24 \}
Off point 52 = \( P_{74} = (1, 7, 1) \) lies on 4 bisecants : \{ 4, 8, 12, 25 \}
Off point 53 = \( P_{75} = (2, 7, 1) \) lies on 4 bisecants : \{ 3, 6, 14, 27 \}
Off point 54 = \( P_{76} = (3, 7, 1) \) lies on 4 bisecants : \{ 17, 20, 22, 26 \}
Off point 55 = \( P_{80} = (7, 7, 1) \) lies on 4 bisecants : \{ 5, 10, 13, 18 \}
Off point 56 = \( P_{81} = (8, 7, 1) \) lies on 4 bisecants : \{ 1, 11, 15, 21 \}
Off point 57 = \( P_{85} = (3, 8, 1) \) lies on 4 bisecants : \{ 14, 16, 18, 20 \}
Off point 58 = \( P_{86} = (4, 8, 1) \) lies on 4 bisecants : \{ 3, 4, 21, 22 \}
Off point 59 = \( P_{87} = (5, 8, 1) \) lies on 4 bisecants : \{ 10, 11, 24, 25 \}
Off point 60 = \( P_{88} = (6, 8, 1) \) lies on 4 bisecants : \{ 5, 8, 23, 27 \}
Off point 61 = \( P_{89} = (7, 8, 1) \) lies on 4 bisecants : \{ 2, 9, 12, 26 \}
Off point 62 = \( P_{90} = (8, 8, 1) \) lies on 4 bisecants : \{ 6, 7, 15, 19 \}

187
7.3 Classification

There are several different approaches to classify cubic surfaces with 27 lines over finite fields $\mathbb{F}_q$ in Orbiter. Classification means to determine the non-equivalent surfaces under the action of the collineation group $\text{PGL}(4, q)$ of $\text{PG}(3, q)$. The approach described in [11] relies on Schlaefli's notion of a double six as a substructure [54]. The approach described in [33] utilizes the relation to non-conical six-arcs in a plane. A third approach is described in [34]. All three approaches are available in Orbiter.

In $\text{PG}(3, 4)$, there is only one type of cubic surfaces with 27 lines. It is a member of the Hirschfeld family, described in [29]. 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 [11] to do so:

```bash
surface classify_q4:
> $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot5\cdot$
> > -define F:-finite_field-q4-end-
> > -define P:-projective_space-3-F:-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-
> > > > -report -end-
> > > > -end-
> > > > -print_symbols
> > > pdflatex Surfaces_q4.tex
> > > open Surfaces_q4.pdf
```

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

**The semilinear group**

**The Action**

Group action $\text{PGL}(4, 4)$ of degree 85
The group is a matrix group.

The base action is on projective space $\text{PG}(3, 4)$
$q = 4$
The orthogonal group

The Action

Group action $\Gamma L(4, 4)\text{OnWedge}$ of degree 1365
The group is a matrix group.
The base action is on projective space $\text{PG}(3, 4)$

The group stabilizing the fixed line

The Action

Group action $\Gamma L(4, 4)\text{OnWedgeres100}$ of degree 100

The classification of five-plus-ones

Poset classification up to depth 5
The Orbits

Number of Orbits By Level

<table>
<thead>
<tr>
<th>Depth</th>
<th>Nb of orbits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Summary of Orbit Representatives

N = node
D = depth or level
O = orbit with a level
Rep = orbit representative
(S,O) = (order of stabilizer, orbit length)
L = number of live points
F = number of flags
Gen = number of generators for the stabilizer of the orbit rep.

Table 7.3: Orbit Representatives

<table>
<thead>
<tr>
<th>N</th>
<th>D</th>
<th>O</th>
<th>Rep</th>
<th>(S,O)</th>
<th>L</th>
<th>F</th>
<th>Gen</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>{}</td>
<td>(5529600, 1)</td>
<td>100</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>{ 0 }</td>
<td>(55296, 100)</td>
<td>64</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>{ 0, 3 }</td>
<td>(1728, 3200)</td>
<td>36</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>{ 0, 3, 56 }</td>
<td>(144, 38400)</td>
<td>16</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>{ 0, 3, 56, 76 }</td>
<td>(288, 19200)</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>{ 0, 3, 56, 77 }</td>
<td>(96, 57600)</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
<td>{ 0, 3, 56, 80 }</td>
<td>(72, 76800)</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>0</td>
<td>{ 0, 3, 56, 76, 96 }</td>
<td>(1440, 3840)</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1</td>
<td>{ 0, 3, 56, 76, 97 }</td>
<td>(96, 57600)</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>2</td>
<td>{ 0, 3, 56, 80, 92 }</td>
<td>(360, 15360)</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<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>
</tbody>
</table>
Poset of Orbits in Detail

Classification of $5 + 1$ Configurations in $PG(3, 4)$

The order of the group is 1974067200
The group has 4 orbits on five plus one configurations in $PG(3, 4)$.

Of these, 1 impose 19 conditions.
Of these, 1 are associated with double sixes. They are:
0/1 is orbit $3/4 \{0, 3, 56, 80, 93\}_{120}$ orbit length 46080
The overall number of five plus one configurations associated with double sixes in $PG(3, 4)$ is: 46080

Flag orbits for double sixes

The number of primary orbits below is 4
The number of primary orbits above is 1
The number of flag orbits is 1
The flag orbits are:

(1) Flag orbit $0 / 1$ down=$\{3,0\}$ up=$\{0,-1\}$ is $\{0, 3, 56, 80, 93, 16, 340, 38, 61, 156, 0, 16, 340, 38, 61, 156, 165, 72, 54, 25, 356, 0 \}$ with a stabilizer of order 120

Strong generators for a group of order 120:

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

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

The order of the group is 1974067200
The group has 1 orbits:

The orbits are:

(1) 0/1 \{16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0\}_{1440} orbit length 1370880
Strong generators for a group of order 1440:

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

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

nb received = 0

193
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 \(0 \downarrow 1\) down\((0,0)\) up\((-1,0)\) is \((16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0, 71, 55, 26, 100, 1, 138, 109, 345, 84, 85, 122, 110, 145, 139, 81)\) with a stabilizer of order 1440

Strong generators for a group of order 1440:

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

nb received = 0

**Surfaces**

The order of the group is 1974067200
The group has 1 orbits:

The orbits are:

(1) \(0/1 \{16, 340, 38, 61, 156, 165, 155, 72, 54, 25, 356, 0, 71, 55, 26, 100, 1, 138, 109, 345, 84, 85, 122, 110, 145, 139, 81\}_{51840}\) orbit length 38080
Strong generators for a group of order 51840:

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

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

The overall number of objects is: 38080

**The Group** $\text{PGL}(4, 4)$

The order of the group is 1974067200

**Cubic Surfaces with 27 Lines in** $\text{PG}(3, 4)$

The order of the group is 1974067200

The group has 1 orbits:

The orbits are:
Strong generators for a group of order 51840:
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
\omega^2 & \omega & \omega^2 & 1 \\
\omega^2 & 0 & 1 & 0 \\
\omega & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 0 & \omega & 0 \\
\omega^2 & \omega & \omega^2 & 0 \\
\omega & 0 & 1 & 0 \\
\end{bmatrix}
\]
1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,
1,0,0,0,2,0,0,0,0,2,0,0,0,0,1,0,
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, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 )
Number of points on the surface 45
The automorphism group of the surface has order 51840
The automorphism group is the following group

Strong generators for a group of order 51840:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_1,
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_0.
\]

General information

Points on lines:

\[5^{27}\]

Lines on points:

\[3^{45}\]

The 27 Lines

\[\ell_0 = a_1 = \begin{bmatrix}
1 & 0 & \omega^2 \\
0 & 1 & 1 \\
0 & 1 & \omega
\end{bmatrix}_{72} = \begin{bmatrix}
1 & 0 & 3 \\
0 & 1 & 1 \\
0 & 1 & 0
\end{bmatrix}_{72} = \text{Pl}(3, 2, 3, 0, 3, 1)_{308}\]

\[\ell_1 = a_2 = \begin{bmatrix}
1 & 0 & \omega \\
0 & 1 & 0 \\
0 & 1 & \omega^2
\end{bmatrix}_{54} = \begin{bmatrix}
1 & 0 & 2 \\
0 & 1 & 0 \\
0 & 1 & 3
\end{bmatrix}_{54} = \text{Pl}(2, 3, 0, 0, 2, 1)_{238}\]
\[ \ell_2 = a_3 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{25} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{25} = \Pi(1, 1, 0, 0, 1, 1)_{177} \]

\[ \ell_3 = a_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{356} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{356} = \Pi(0, 1, 0, 0, 0, 0)_{1} \]

\[ \ell_4 = a_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}_{0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}_{0} = \Pi(1, 0, 0, 0, 0, 0)_{0} \]

\[ \ell_5 = a_6 = \begin{bmatrix} 1 & 0 & \omega^2 & 1 \\ 0 & 1 & 0 & \omega \end{bmatrix}_{155} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 0 & 1 & 0 & 2 \end{bmatrix}_{155} = \Pi(3, 2, 0, 2, 3, 1)_{314} \]

\[ \ell_6 = b_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{340} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{340} = \Pi(0, 0, 0, 1, 0, 0)_{9} \]

\[ \ell_7 = b_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{38} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{38} = \Pi(0, 0, 1, 1, 1)_{198} \]

\[ \ell_8 = b_3 = \begin{bmatrix} 1 & \omega & 0 & 0 \\ 0 & 0 & 1 & \omega^2 \end{bmatrix}_{61} = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix}_{61} = \Pi(0, 0, 2, 3, 2, 1)_{265} \]

\[ \ell_9 = b_4 = \begin{bmatrix} 1 & 0 & \omega^2 & 1 \\ 0 & 1 & 1 & \omega \end{bmatrix}_{156} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}_{156} = \Pi(3, 0, 3, 2, 3, 1)_{335} \]

\[ \ell_{10} = b_5 = \begin{bmatrix} 1 & \omega & 0 & 1 \\ 0 & 0 & 1 & \omega \end{bmatrix}_{165} = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}_{165} = \Pi(0, 2, 3, 2, 3, 1)_{337} \]

\[ \ell_{11} = b_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{16} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{16} = \Pi(0, 0, 1, 0, 0, 0)_{2} \]

\[ \ell_{12} = c_{12} = \begin{bmatrix} 1 & 0 & \omega & 1 \\ 0 & 1 & 0 & \omega^2 \end{bmatrix}_{138} = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 3 \end{bmatrix}_{138} = \Pi(2, 3, 0, 3, 2, 1)_{256} \]

\[ \ell_{13} = c_{13} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{109} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{109} = \Pi(1, 1, 0, 1, 1, 1)_{189} \]

\[ \ell_{14} = c_{14} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{345} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{345} = \Pi(0, 1, 0, 1, 0, 0)_{13} \]

\[ \ell_{15} = c_{15} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}_{84} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}_{84} = \Pi(1, 0, 0, 1, 0, 0)_{10} \]
\[ \ell_{16} = c_{16} = \begin{bmatrix} 1 & 0 & \omega^2 & 0 \\ 0 & 1 & 0 & \omega \end{bmatrix}_{71} = \begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}_{71} = \text{Pl}(3, 2, 0, 0, 3, 1)_{299} \]

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

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

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

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

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

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

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

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

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

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

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:

0 : $E_{56} = a_5 \cap b_6 \cap c_{56} = P_0 = P_0 = P(1,0,0,0) = P(1,0,0,0), T = 0$
1 : $E_{51} = a_5 \cap b_1 \cap c_{15} = P_1 = P_1 = P(0,1,0,0) = P(0,1,0,0), T = 4$
2 : $E_{46} = a_4 \cap b_6 \cap c_{46} = P_2 = P_2 = P(0,0,1,0) = P(0,0,1,0), T = 20$
3 : $E_{41} = a_4 \cap b_1 \cap c_{41} = P_3 = P_3 = P(0,0,0,1) = P(0,0,0,1), T = 84$
4 : $E_{32} = a_3 \cap b_2 \cap c_{32} = P_4 = P_4 = P(1,1,1,1) = P(1,1,1,1), T = 27$
5 : $E_{36} = a_5 \cap b_2 \cap c_{36} = P_5 = P_5 = P(1,1,0,0) = P(1,1,0,0), T = 1$
6 : $E_{54} = a_5 \cap b_4 \cap c_{45} = P_6 = P_6 = P(\omega,1,0,0) = P(2,1,0,0), T = 2$
7 : $E_{33} = a_5 \cap b_3 \cap c_{35} = P_7 = P_7 = P(\omega^2,1,0,0) = P(3,1,0,0), T = 3$
8 : $E_{36} = a_3 \cap b_6 \cap c_{36} = P_8 = P_8 = P(1,0,1,0) = P(1,0,1,0), T = 5$
9 : $E_{16} = a_1 \cap b_6 \cap c_{16} = P_9 = P_9 = P(\omega,0,1,0) = P(2,0,1,0), T = 10$
10 : $E_{26} = a_2 \cap b_6 \cap c_{26} = P_{10} = P_{10} = P(\omega^2,0,1,0) = P(3,0,1,0), T = 15$
11 : $E_{14,23,56} = c_{14} \cap c_{23} \cap c_{56} = P_{11} = P_{11} = P(0,1,1,0) = P(0,1,1,0), T = 9$
12 : $E_{13,24,56} = c_{13} \cap c_{24} \cap c_{56} = P_{12} = P_{12} = P(1,1,1,0) = P(1,1,1,0), T = 6$
13 : $E_{66} = a_6 \cap b_5 \cap c_{56} = P_{13} = P_{13} = P(\omega,1,1,0) = P(2,1,1,0), T = 12$
14 : $E_{12,34,56} = c_{12} \cap c_{34} \cap c_{56} = P_{14} = P_{14} = P(\omega^2,1,1,0) = P(3,1,1,0), T = 18$
15 : $E_{15,23,46} = c_{15} \cap c_{23} \cap c_{46} = P_{15} = P_{23} = P(1,0,1,0) = P(1,0,1,0), T = 21$
16 : $E_{31} = a_3 \cap b_1 \cap c_{13} = P_{16} = P_{26} = P(0,1,0,1) = P(0,1,0,1), T = 25$
17 : $E_{15,24,36} = c_{15} \cap c_{24} \cap c_{36} = P_{17} = P_{27} = P(1,1,0,1) = P(1,1,0,1), T = 22$
18 : $E_{21} = a_2 \cap b_1 \cap c_{12} = P_{18} = P_{30} = P(0,\omega,0,1) = P(0,2,0,1), T = 46$
19 : $E_{15,26,34} = c_{15} \cap c_{26} \cap c_{34} = P_{19} = P_{31} = P(1,\omega,0,1) = P(1,2,0,1), T = 24$
20 : $E_{61} = a_6 \cap b_1 \cap c_{16} = P_{20} = P_{34} = P(0,\omega^2,0,1) = P(0,3,0,1), T = 67$
21 : $E_{15} = a_1 \cap b_5 \cap c_{15} = P_{21} = P_{35} = P(1,\omega^2,0,1) = P(1,3,0,1), T = 23$
22 : $E_{42} = a_4 \cap b_2 \cap c_{24} = P_{22} = P_{38} = P(0,0,1,1) = P(0,0,1,1), T = 41$
23 : $E_{13,25,46} = c_{13} \cap c_{25} \cap c_{46} = P_{23} = P_{39} = P(1,0,1,1) = P(1,0,1,1), T = 26$
24 : $E_{14,25,36} = c_{14} \cap c_{25} \cap c_{36} = P_{24} = P_{42} = P(0,1,1,1) = P(0,1,1,1), T = 30$
25 : $E_{62} = a_6 \cap b_2 \cap c_{26} = P_{25} = P_{47} = P(\omega,\omega,1,1) = P(2,2,1,1), T = 53$
26 : $E_{25} = a_2 \cap b_5 \cap c_{25} = P_{26} = P_{48} = P(\omega^2,\omega,1,1) = P(3,2,1,1), T = 80$
27 : $E_{16,25,34} = c_{16} \cap c_{25} \cap c_{34} = P_{27} = P_{51} = P(\omega,\omega^2,1,1) = P(2,3,1,1), T = 55$
28 : $E_{12} = a_1 \cap b_2 \cap c_{12} = P_{28} = P_{32} = P(\omega^2,\omega^2,1,1) = P(3,3,1,1), T = 79$
29 : $E_{43} = a_4 \cap b_3 \cap c_{34} = P_{29} = P_{53} = P(0,0,\omega,1) = P(0,0,2,1), T = 62$
30 : $E_{12,35,46} = c_{12} \cap c_{35} \cap c_{46} = P_{30} = P_{54} = P(1,0,\omega,1) = P(1,0,2,1), T = 36$
31 : $E_{35} = a_3 \cap b_5 \cap c_{35} = P_{31} = P_{59} = P(\omega,\omega,1,1) = P(2,1,2,1), T = 49$
32 : $E_{63} = a_6 \cap b_3 \cap c_{36} = P_{32} = P_{60} = P(\omega^2,1,\omega,1) = P(3,1,2,1), T = 76$
33 : $E_{14,26,35} = c_{14} \cap c_{26} \cap c_{35} = P_{33} = P_{61} = P(0,\omega,\omega,1) = P(0,2,2,1), T = 51$
34 : $E_{23} = a_2 \cap b_3 \cap c_{23} = P_{34} = P_{62} = P(1,\omega,\omega,1) = P(1,2,2,1), T = 39$
35 : $E_{13} = a_1 \cap b_3 \cap c_{13} = P_{35} = P_{67} = P(\omega,\omega,\omega,1) = P(2,3,2,1), T = 50$
36 : $E_{15,24,35} = c_{15} \cap c_{24} \cap c_{35} = P_{36} = P_{68} = P(\omega^2,\omega^2,\omega,1) = P(3,3,2,1), T = 74$
37 : $E_{45} = a_4 \cap b_5 \cap c_{45} = P_{37} = P_{69} = P(0,0,\omega^2,1) = P(0,0,3,1), T = 83$
38 : $E_{64} = a_6 \cap b_4 \cap c_{46} = P_{38} = P_{70} = P(1,0,\omega^2,1) = P(1,0,3,1), T = 31$
39 : $E_{12,36,45} = c_{12} \cap c_{36} \cap c_{45} = P_{39} = P_{75} = P(\omega,1,\omega^2,1) = P(2,1,3,1), T = 59$
40 : $E_{34} = a_3 \cap b_4 \cap c_{34} = P_{40} = P_{76} = P(\omega^2,1,\omega^2,1) = P(3,1,3,1), T = 71$
41 : \( E_{24} = a_2 \cap b_4 \cap c_{24} = P_{41} = P_{79} = \mathbb{P}(\omega, \omega, \omega^2, 1) = \mathbb{P}(2, 2, 3, 1), \ T = 58 \)
42 : \( E_{13,26,45} = c_{13} \cap c_{26} \cap c_{45} = P_{42} = P_{80} = \mathbb{P}(\omega^2, \omega, \omega^2, 1) = \mathbb{P}(3, 2, 3, 1), \ T = 70 \)
43 : \( E_{14} = a_1 \cap b_4 \cap c_{14} = P_{43} = P_{81} = \mathbb{P}(0, \omega^2, \omega^2, 1) = \mathbb{P}(0, 3, 3, 1), \ T = 72 \)
44 : \( E_{16,23,45} = c_{16} \cap c_{23} \cap c_{45} = P_{44} = P_{82} = \mathbb{P}(1, \omega^2, \omega^2, 1) = \mathbb{P}(1, 3, 3, 1). \ T = 33 \)

Set of tangent planes: \( \{0, 4, 20, 84, 27, 1, 2, 3, 5, 10, 15, 9, 6, 12, 18, 21, 25, 22, 46, 24, 67, 23, 41, 26, 30, 53, 80, 55, 79, 62, 36, 49, 76, 51, 39, 50, 74, 83, 31, 59, 71, 58, 70, 72, 33\} \)
Line type of Eckardt points: \( 5^{27}, 3^{240}, 1^{90} \)
Plane type of Eckardt points: \( 13^{45}, 9^{40} \)

**Hesse planes**

Number of Hesse planes: 40
Set of Hesse planes: \( \{7, 8, 11, 13, 14, 16, 17, 19, 28, 29, 32, 34, 35, 37, 38, 40, 42, 43, 44, 45, 47, 48, 52, 54, 56, 57, 60, 61, 63, 64, 65, 66, 68, 69, 73, 75, 77, 78, 81, 82\} \)
subspace 0 / 40 is 7:

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

::
subspace 39 / 40 is 82:

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

0 : 7 : \( E_{56}, E_{31}, E_{15,24,36}, E_{16,25,34}, E_{12}, E_{14,26,35}, E_{23}, E_{45}, E_{64} \)
::
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 = \( E_{23}, E_{31}, E_{12} \)
::
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 [33] 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.-
   $\$define.F.-finite_field.-q.4.-end.-
   $\$define.P.-projective_space.3.F.-end-
   $\$with.P.-do-
   $\$with-projective_space_activity-
   $\$with.control.six_arcs.-problem_label.sixarcs_q4.-end-
   $\$with.classify_surfaces_through_arcs_and_two_lines-
   $\$with.end
   $\$pdflatex.-surfaces.arc_lifting.4.tex
   $\$open.-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 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,</td>
<td>Computes an isomorphism between two given surfaces or concludes that none exists.</td>
</tr>
<tr>
<td></td>
<td>surface-descr-2</td>
<td></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:-q:7:-end-
▷ -define:P:-projective_space-3:F:-end-
▷ -with:P:-do-
▷ -projective_space_activity-
▷ -classify_surfaces_with_double_sixes_Surf:-W:-end-
▷ -end-
▷ -with:Surf:-do-
▷ -classification_of_cubic_surfaces_with_double_sixes_activity-
▷ -recognize-
```

203
identifies the surface (cf. Table 4.6)

\[ X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 + X_1X_2X_3 = 0 \]  

(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

\texttt{surface\_isomorph\_16:}

\begin{verbatim}
$ (ORBITER\_PATH) orbiter.out -v.3 \ndefine F -finite_field -q 16 \ndefine P -projective_space -3 F \nwith P -do \nprojective_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> -print_symbols
\end{verbatim}

204
computes an isomorphism between two cubic surfaces with 27 lines

\[
X_0^2X_2 + X_1^2X_2 + X_1^3X_3 + X_0X_2^2 + X_1X_2^2 + X_2^2X_3 + \delta^{13}X_1X_3^2 + \delta^{12}X_2X_3^2 + \delta^{11}X_0X_2X_3 + X_1X_2X_3 = 0
\]

and

\[
\delta^{11}X_0^2X_3 + \delta^{12}X_1^2X_2 + \delta^{11}X_1X_2^2 + \delta^{11}X_0X_2^2 + X_1X_2X_3 = 0
\]

over the field \( \mathbb{F}_{16} \).

\[
\begin{bmatrix}
12 & 13 & 0 & 0 \\
8 & 13 & 0 & 0 \\
0 & 0 & 13 & 0 \\
12 & 13 & 11 & 1
\end{bmatrix}
\]

Orbiter can recognize the isomorphism type of a cubic surface with 27 lines. This means that Orbiter can determine the Orbiter Catalogue Number of the surface in the catalogue which is isomorphic to the given surface. For instance, the following command determines the isomorphism type of the surface

\[
X_0^2X_3 + X_1^2X_2 + X_1X_2^2 + X_0X_3^2 + X_1X_2X_3 = 0.
\]

```
surface_recognize_8:  
  > $(ORBITER_PATH)orbiter.out -v 3 \ 
  > -define F -finite_field -q 8 -end \ 
  > -define P -projective_space -3 F -end \ 
  > -with P -do \ 
  > -projective_space_activity \ 
  > -classify_surfaces_with_double_sixes Surf27 -W -end \ 
  > -end \ 
  > -with Surf27 -do \ 
  > -classification_of_cubic_surfaces_with_double_sixes_activity \ 
  > -recognize \ 
  > -q 8 \ 
  > -by_coefficients "1,6,1,8,1,11,1,13,1,19" \ 
  > -end \ 
  > -end \ 
  > -print symbols
```

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 [20]. 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 ·
  -draw_options -yout 500000 -radius 15 -nodes_empty ·
  -line_width 0.5 -y_stretch 0.25 -embedded ·
  -define G -linear_group -PGL 4 2 ·
  -with G ·
  -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_d3n3q2_F2.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 ·
  -with P ·
  -projective_space_activity ·
  -table_of_cubic_surfaces_compute_properties ·
  -orbits_on_polynomials_draw_tree 2 0 ·
  -end ·
```

Dickson_q2_analyze: poly_orbits_d3n3q2_F2.csv
```
poly_orbits_d3n3q2_F2.csv: poly_orbits_d3n3q2_F2.csv
  $(\text{ORBITER\_PATH})\text{orbiter.out} -v 4 ·
  -define F -finite_field -q 2 -end ·
  -define P -projective_space 3 F ·
  -with P ·
  -projective_space_activity ·
  -cubic_surface_properties_analyze ·
  -orbits_on_polynomials_draw_tree 2 ·
  -end ·
```

To investigate the properties of these surfaces over the extension field $\mathbb{F}_4$, the following commands can be used:

```
pdflatex poly_orbits_d3n3q2_F2_report.tex
open poly_orbits_d3n3q2_F2_report.pdf
```
poly_orbits_d3_n3_q2_F4.csv: poly_orbits_d3_n3_q2.csv

 $(\text{ORBITER\ PATH})\text{oriber.out}\ -v\ 4$

 -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

 $(\text{ORBITER\ PATH})\text{oriber.out}\ -v\ 4$

 -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 \( q = 17 \):

\[
\text{MAKE\_TABLE\_OF\_CUBIC\_SURFACES}=-\text{define}\, P\, -\text{projective\_space}\, 3\, F\, -\text{end}\, \$
\]
\[
\quad \text{-define}\, \text{P}\, \text{-projective\_space}\, \text{activity}\, \$
\]
\[
\quad \text{-table\_of\_cubic\_surfaces}\, \$
\]
\[
\quad \text{-end}
\]

\[
cubic\_surfaces\_tables\_17:\$
\]
\[
\quad \$(\text{ORBITER\_PATH})\text{orbiter\_out}\,-v\,3\, \$
\]
\[
\quad \text{-define}\, F\, -\text{finite\_field}\, -q\,17\, -\text{end}\, \$
\]
\[
\quad \$(\text{MAKE\_TABLE\_OF\_CUBIC\_SURFACES})
\]

A file `table_of_cubic_surfaces_q17_info.csv` is created. The command

\[
cubic\_surfaces\_table\_latex\_17:\$
\]
\[
\quad \$(\text{ORBITER\_PATH})\text{orbiter\_out}\,-v\,3\,-\text{csv\_file\_latex}\,1\, \$
\]
\[
\quad \text{table\_of\_cubic\_surfaces\_q17\_info.csv}
\]

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,

\[
\text{inverse}
\text{mod}
\text{a:}
\]

\[
\diamond
\text{(ORBITER PATH) orbiter.out -v 2 - inverse mod 18059241 58014043}
\]

computes the inverse of 18059241 modulo 58014043.

The Legendre symbol tells us if a number \( a \) is a square modulo an odd prime \( p \). By definition,

\[
\left( \frac{a}{p} \right) = \begin{cases} 
1 & \text{if there exists } r \text{ s.t. } r^2 \equiv a \mod p \\
-1 & \text{if there does not exist } r \text{ s.t. } r^2 \equiv a \mod p \\
0 & \text{if } p \text{ divides } a.
\end{cases}
\]

The Jacobi symbol generalizes the Legendre symbol to allow non-prime bottom arguments. By definition,

\[
\left( \frac{a}{b} \right) = \prod_{i=1}^{k} \left( \frac{a}{r_i^{e_i}} \right),
\]

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

\[
jacobi_a:
\]

\[
\diamond \ (\text{ORBITER PATH}) \ orbiter.out -v 5 - jacobi 2221 7817
\]
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:

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

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:  
\$\text{(ORBITER_PATH)}\text{orbiter.out.-v.2.}-\text{square_root_mod.2221.7817}
```

yields that 7634 is a square root. Indeed,

\[7634^2 \equiv 2221 \mod 7817.\]
<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 8.2: Combinatorial Commands

### 8.2 Combinatorics

In Table 8.2, Orbiter commands for combinatorics are shown.

For instance, the command

Char_Sym_4:

\[
\text{\$}($\text{ORBITER\_PATH})\text{orbiter.out\_v\_2\_character_table_symmetric_group\_4}$
\]

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

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

\[
\text{elementary\_symmetric\_functions\_4:}
\]

\[
\text{\$}($\text{ORBITER\_PATH})\text{orbiter.out\_make\_elementary\_symmetric\_functions\_4\_4}$
\]

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

\[
k=1:
\]

\[
x_0 + x_1 + x_2 + x_3
\]
k=2:
x0*x1 + x0*x2 + x0*x3 + x1*x2 + x1*x3 + x2*x3
k=3:
x0*x1*x2 + x0*x1*x3 + x0*x2*x3 + x1*x2*x3
k=4:
x0*x1*x2*x3

Orbiter can be used to classify incidence structures and combinatorial designs. A point-line incidence structure can be encoded as 0/1-matrix. Orbiter inc files can be used to store the 0/1 matrices. The following command creates an inc file for three configurations 24\textsubscript{3} and then invokes Orbiter to perform isomorphism testing:

```plaintext
FILE_24_3_TFC_INC="24-24-72"
```

design_test:
  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 -
  > -save_ago -
  > -end
<table>
<thead>
<tr>
<th>Command</th>
<th>Args</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-orbits_on_polynomials</td>
<td>d</td>
<td>Computes the representation of the group $G$ on homogeneous polynomials of degree $d$. This is a group theoretic activity as described in Section 5.6. The group $G$ must be constructed first.</td>
</tr>
</tbody>
</table>

Table 8.3: Representation Theory Commands

8.3 Representation Theory

Orbiter has some commands for representations of finite groups. Table 8.3 list the commands available to classify arcs. The command

```
representation_on_polynomials_of_degree_3:
  $(ORBITER_PATH)orbiter.out--v.4:\
  -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 $\text{PGL}(4, 3)$ on cubic polynomials
### Table 8.4: Cryptographic Commands

<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>$a \ n_1 \ n_2 \ n_3$</td>
<td>Computes a pseudoprime which survives $n_1$ Fermat tests, $n_2$ Miller Rabin tests, $n_3$ Solovay Strassen tests</td>
</tr>
<tr>
<td>-find_strong_pseudoprime</td>
<td>$a \ n_1 \ n_2$</td>
<td>Computes a pseudoprime which survives $n_1$ Fermat tests and $n_2$ Miller Rabin tests</td>
</tr>
<tr>
<td>-RSA_encrypt_text</td>
<td>$d \ n \ b \ \text{text}$</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>$d \ n \ \text{list-of-integers}$</td>
<td>encrypt the given sequence of integers using RSA with exponent $d$ modulo $n$</td>
</tr>
</tbody>
</table>

### 8.4 Cryptography

In Table 8.4, some cryptographic commands are shown. In Table 8.4, some cryptographic commands depending on a finite field are shown. We assume that the field $\mathbb{F}_q$ has been defined. For instance,

```plaintext
EC_add:
- $(ORBITER\_PATH)orbiter.out\cdot-v.2\cdot$
- -define\:F::finite\_field::q.11::end\$
- -with\:F::do\$
- -finite\_field\_activity\$
- -EC\_add.1.3:"1,4"::"1,4"::-end
```

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

```plaintext
EC_cyclic_subgroup:
- $(ORBITER\_PATH)orbiter.out\cdot-v.2\cdot$
- -define\:F::finite\_field::q.11::end\$
- -with\:F::do\$
- -finite\_field\_activity\$
- -EC\_cyclic\_subgroup.1.3:"1,4"::-end
```

216
<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 8.5: Finite Field Activities related to Cryptography
Figure 8.2: The elliptic curve $y^2 = x^3 + 5x + 7 \pmod{199}$

computes the cyclic subgroup generated by the point $(1, 4)$ on the curve $y^2 = x^3 + x + 3 \pmod{11}$. The command

\begin{verbatim}
EC_points_199:
  $(ORBITER_PATH)orbiter.out\ -v\ -2\$
  -define:F:\ -finite_field:q199\ -end$
  -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
\end{verbatim}

computes all points on the curve $y^2 = x^3 + 5x + 7 \pmod{199}$ 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 \texttt{-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.\$
		define\ F:\ finite\ field\ \ -q\ 199:\ end\$
		with\ F:\ do\$
		finite_field_activity\$
		EC_Koblitz_encoding\ 5\ 7\ 67.\"147,164".\"DEADBEEF".\$
		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:
	$\$(ORBITER_PATH)orbiter.out\ -v\ 2.\$
		define\ F:\ finite\ field\ \ -q\ 199:\ end\$
		with\ F:\ do\$
		finite_field_activity\$
		EC_bsgs\ 5\ 7\ "147,164".\ 212.\$
		"172,158,45,195,50,22,10,103,55,33,50,22,145,105,31,74,73,155,67,60,25,6".\$
		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.

```
EC_bsgs_decode:
	$\$(ORBITER_PATH)orbiter.out\ -v\ 2.\$
		define\ F:\ finite\ field\ \ -q\ 199:\ end\$
		with\ F:\ do\$
		finite_field_activity\$
		EC_bsgs_decode\ 5\ 7\ "129,176".\ 212.\$
		"127,188,51,141,85,29,106,90,41,105,179,71,171,2,16,197,183,72,27,129,37,10".\$
		"50,179,169,13,153,169,115,116,188,110,176".\$
		end
```
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 [31]). 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 \(-\frac{p}{2} \leq a_i \leq \frac{p}{2}\). So, in an extension to the conventions 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:
NTRU_Alice1:
> $(ORBITER_PATH)orbiter.out: -v.2\n>   -define:finite_field:q$(NTRU_Q):-end\n>   -with:do\n>   -finite_field_activity\n>   -extended_gcd_for_polynomials$(NTRU_XN1).$(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:
> $(ORBITER_PATH)orbiter.out: -v.2\n>   -define:finite_field:q$(NTRU_P):-end\n>   -with:do\n>   -finite_field_activity\n>   -extended_gcd_for_polynomials$(NTRU_XN1).$(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:
> $(ORBITER_PATH)orbiter.out: -v.2\n>   -define:finite_field:q$(NTRU_Q):-end\n>   -with:do\n>   -finite_field_activity\n>   -polynomial_mult_mod$(ALICE_PRIVATE_F)\n>   $(ALICE_PRIVATE_G).$(NTRU_XN1).\n>   -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) -

#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 -

222
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

\texttt{Hamming\_space\_4\_2\_distance\_matrix:}
\texttt{$($\texttt{ORBITER\_PATH}$)$orbiter.out\_\texttt{-Hamming\_space\_distance\_matrix\_4\_2}}
\texttt{$($\texttt{ORBITER\_PATH}$)$orbiter.out\_\texttt{-v\_2\_draw\_matrix\_}}
\texttt{$($\texttt{ORBITER\_PATH}$)$orbiter.out\_\texttt{-v\_2\_draw\_matrix\_}}
\texttt{-input\_csv\_file\_Hamming\_n4\_q2\_csv\_}}
\texttt{-box\_width\_20\_bit\_depth\_24\_partition\_4\_16\_16\_end}
\texttt{open\_Hamming\_n4\_q2\_draw\_bmp}

create the csv-file \texttt{Hamming\_n4\_q2\_csv} and produce the bitmap file

\texttt{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 \(\{0,7,25,30\}\):

\texttt{CODE\_5\_2\_3\_CODEWORDS="0\_7\_25\_30"}

\texttt{code\_5\_2\_3\_diagram:}
\texttt{$($\texttt{ORBITER\_PATH}$)$orbiter.out\_\texttt{-v\_2\_code\_diagram\_"code\_5\_2\_3\"}}
\texttt{$($\texttt{ORBITER\_PATH}$)$orbiter.out\_\texttt{-v\_2\_code\_diagram\_}}
\texttt{-input\_csv\_file\_code\_5\_2\_3\_diagram\_01\_5\_4\_csv\_}}
\texttt{-box\_width\_25\_bit\_depth\_24\_partition\_4\_8\_4\_end}

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

\texttt{Hamming\_5\_2\_graph:}
\texttt{$($\texttt{ORBITER\_PATH}$)$orbiter.out\_\texttt{-v\_2\_}}
<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>$n \ k \ r_1 \ldots \ r_k$</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.

```
dot -Tpng $ORBITER_PATH/orbiter.out -v 2 -draw_matrix -define G::graph::Hamming_5_2::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 -input_csv_file Hamming_5_2.csv -box_width 8 -bit_depth 24 -partition 4.32.32 -partition 4.32.32 -end
```

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:

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

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\$
```

\[\text{229}\]
Suppose we want to compute the weight enumerator of the Hamming code. We use the following command:

```
HAMMING_CODE_GENERATOR="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"
```

Hamming weight enumerator:

```
$(ORBITER_PATH)orbiter.out -v 2 -
define F -finite_field -q 2 -
define v -vector -field F -format 4 -
 dense $(HAMMING_CODE_GENERATOR) -
end -
with F -do -
finite_field_activity -
weight Enumerator v -end
```

We find that the weight enumerator is

$$(1, 0, 0, 7, 7, 0, 0, 1).$$
Suppose we want to establish the MacWilliams relations for the Hamming code. The following command creates the matrix of Kravtchuck numbers:

Hamming_code_macwilliams:
▷ $(\text{ORBITER PATH})$orbiter.out\-v.2\-make_macwilliams_system.7.4.2
▷ pdflatex MacWilliams_n7_k4_q2.tex
▷ open MacWilliams_n7_k4_q2.pdf

This produces the following output:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
7 & 5 & 3 & 1 & -1 & -3 & -5 & -7 \\
21 & 9 & 1 & -3 & -3 & 1 & 9 & 21 \\
35 & 5 & -5 & -3 & 3 & 5 & -5 & -35 \\
35 & -5 & -5 & 3 & 3 & -5 & -5 & 35 \\
21 & -9 & 1 & 3 & -3 & -1 & 9 & -21 \\
7 & -5 & 3 & -1 & -1 & 3 & -5 & 7 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1
\end{bmatrix}
\]

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

Hamming_singer:
▷ $(\text{ORBITER PATH})$orbiter.out\-v.3.\$
▷ \$ define G -linear_group -PGL.3.2 -singer.1 -end \$
▷ \$ with G -do \$
▷ \$ -group_theoretic_activity \$
▷ \$ -report \$
▷ \$ -orbits_on_points \$
▷ \$ -end \$
▷ pdflatex PGL.3.2_Singer.3.2.1_report.tex
▷ 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

0
1
2
5
3
4
6

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

From this, we know how to rearrange the points of PG(2, 2) to exhibit the cyclic structure. We issue the following command to recreate the Hamming code:

Hamming_cyclic_generator:

\$(\text{ORBITER\_PATH})\text{orbit}er\text{.out} -v 2 \$
\n\text{-define F \text{-finite\_field \text{-q 2 \text{-end}} \text{-define v \text{-vector \text{-format 3 \text{-field F}}}} \text{-dense $(\text{SIMPLEX\_CODE\_GENMA\_CYCLIC})$}} \text{-end}} \text{-with F -do -finite\_field\_activity}} \text{-nullspace v}} \text{-end}

\text{pdflatex nullspace 3.7.tex}
\text{open nullspace 3.7.pdf}

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, \n8,9,10,11,132,913,1460,1750,1898,2518,2787,2874, \n3320,3357,3662".
```

Suppose we want to list the code words. The following command can be used:

```
Golay23_code_words:
  $(ORBITER_PATH)orbiter.out -v 2\n  -define v-vector-dense $(GOLAY_23_COLUMN_RANKS_PROJECTIVELY) -end\n  -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:-polynomial_division_file_13.pdf
```

We decide to pick the binary polynomial $13 = X^3 + X^2 + 1$. We divide the information polynomial by the check polynomial:

```
encode_text_5bits_check:
  $(ORBITER_PATH)orbiter.out:-v.2-
  -define:F:-finite_field:-q:2:-end-
  -with:F:-do-
  -finite_field_activity-
  -polynomial_division_from_file:text_with_1error.csv:13:-end
  pdflatex:-polynomial_division_file_13.tex
  open:-polynomial_division_file_13.pdf
```

This creates the following output:

```
text.csv / 13 =
1010110100110101010111000010111100 / 1101 =
11011100000110111111010000101
==================================
1101 | 1010110100110101010111000010111100
```

235
1101
====
1111010011010110111000010111100
1101
====
101010011101010111100010111100
1101
====
1111001101010111100010111100
1101
====
10011010101110000010111100
1101
====
101101010101111000010111100
1101
====
110101010111000010111100
1101
====
100011010101111000010111100
1101
====
101101010101111000010111100
1101
====
101010111000010111100
1101
====
1011011000010111100
1101
====
100111000010111100
1101
====
10011000010111100
1101
====
1001000010111100
1101
====
100000010111100
1101
====
10100010111100
1101
====
1110010111100
1101
====
11010111100
1101
====
111001
1101
====
110101
1101
====
1010101
1101
====
11110111000010111100
1101
====
100111000010111100
1101
====
10011000010111100
1101
====
1001000010111100
1101
====
100000010111100
1101
====
10100010111100
1101
====
1110010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101
====
11010111100
1101

236
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_error:
  ▶ $(ORBITER_PATH)orbiter.out -encode_text_5bits:\
  ▶ ▶ "Hithere"."text.csv"
  ▶ ▶ $(ORBITER_PATH)orbiter.out -v.2:\
  ▶ ▶ -define:F::finite_field:-q:2:-end:\
  ▶ ▶ -with:F:-do:\
  ▶ ▶ -finite_field_activity:\
  ▶ ▶ -polynomial_division_from_file_all_1_bit_error_patterns:\
  ▶ ▶ ▶ text.csv.13.1-\end
  ▶ ▶ pdflatex:polynomial_division_file_all_1_error_patterns_13.tex
  ▶ open-polynomial_division_file_all_1_error_patterns_13.pdf
```

The following output is created:

```
| 0 | 01010110011010101011101000010111100 |
| 1 | 010101100110101010111010010111101 |
| 2 | 01010110011010101011101010111000 |
| 3 | 01010110011010101011101010111000 |
| 4 | 01010110011010101011101010111000 |
| 5 | 01010110011010101011101010111000 |
| 6 | 01010110011010101011101010111000 |
| 7 | 01010110011010101011101010111000 |
| 8 | 01010110011010101011101010111000 |
| 9 | 01010110011010101011101010111000 |
| 10| 01010110011010101011101010111000 |
| 11| 01010110011010101011101010111000 |
| 12| 01010110011010101011101010111000 |
| 13| 01010110011010101011101010111000 |
```

The following output is created:

```
/13 = 1841528453 Remainder 5
```

```
```

237
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 \( n = 128 \).

```
$ (ORBITER_PATH) orbiter.out -v 1 \
   -define F finite_field -q 2 -end \n   -with F -do finite_field_activity \n   -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.v.vector.dense $(REED_MULLER_3_1_BASIS_IN_BINARY) -end:
  linear_code_through_basis 8.v
  pdflatex code_n8_k4_q2.tex
  open-code_n8_k4_q2.pdf
```

#Codewords: (0,255,170,85,204,51,102,153,240,15,90,165,60,195,150,105)

The output is shown in Figure 9.4.

The following command produces a diagram of the characterstic 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).8:
  -metric_balls 1
  $(ORBITER_PATH)orbiter.out -v -2 draw_matrix:
```
Figure 9.5: Boolean function representation of RM$_{3,1}$ in $H(8,2)$

```
▷▷ -input_csv_file:RM_3_1_diagram_01_8_16.csv \n▷▷ -box_width:25 -bit_depth:8 -partition:4 16 16 -end 
▷▷ $(ORBITER$_PATH)orbiter.out:$v$2:-draw_matrix \n▷▷ -input_csv_file:RM_3_1_diagram_8_16.csv \n▷▷ -box_width:25 -bit_depth:8 -partition:4 16 16 -end
▷▷ open:RM_3_1_diagram_8_16_draw.bmp
```

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

$$\text{lcm}\left(m_{\beta^1, \mathbb{F}_q}, m_{\beta^2, \mathbb{F}_q}, \ldots, m_{\beta^{d-1}, \mathbb{F}_q}\right).$$

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

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

As an example, let us consider the 2-cyclotomic sets of 2 and 3 modulo 255. The following command produces a graphical representation. The 255-th roots of unity are placed on a unit circle, similar to the placement of the corresponding complex roots in the Argand diagram.

draw_mod_255_cyclotomic_1_and_3:

```verbatim
$\text{(ORBITER\_PATH)orbirer.out -v 2;}
\text{-draw.options -nodes empty -radius 10;}
\text{-line_width 0.4 -embedded -end;}
\text{-draw_mod_n -n 255 -file mod_255_cyclotomic_1_and_3;}
\text{-cycloic_sets 2,"1,3" -end}
\text{pdflatex mod_255_cyclotomic_1_and_3_draw.tex}
\text{open mod_255_cyclotomic_1_and_3_draw.pdf}
```

The drawing is shown in Figure 9.6.
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:

```plaintext
F256_roots_771:
  ▶ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot3\backslash
  ▶ ▶ -\text{define}\cdot\text{F\_finite_field}\cdot-q\cdot256\cdot-\text{end}\backslash
  ▶ ▶ ▶ -\text{with}\cdot\text{F\_do}\cdot\text{finite_field_activity}\cdot-nth\_roots\cdot771\cdot-\text{end}
```

The next command creates a BCH-code of length 771 over $\mathbb{F}_{256}$ with minimum distance at least 16:

```plaintext
F256_BCH_code_d16:
  ▶ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot3\backslash
  ▶ ▶ -\text{define}\cdot\text{F\_finite_field}\cdot-q\cdot256\cdot-\text{end}\backslash
  ▶ ▶ ▶ -\text{with}\cdot\text{F\_do}\cdot\text{finite_field_activity}\cdot\text{make_BCH_code}\cdot771\cdot16\cdot-\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:

```plaintext
POLY_Q256_DEG30_DENSE="1,26,210,24,138,148,160,58,108,199,95,56,9,\n205,194,193,3,248,110,150,24,169,192,212,112,144,97,109,174,253,1"
```

```plaintext
F256_BCH_code_d16_division:
  ▶ $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot2\backslash
  ▶ ▶ -\text{define}\cdot\text{F\_finite_field}\cdot-q\cdot256\cdot-\text{end}\backslash
  ▶ ▶ ▶ -\text{define}\cdot\text{A\_vector\_field}\cdot\text{F\_sparse}\cdot772\cdot"1,771,1,0"\cdot-\text{end}\backslash
  ▶ ▶ ▶ ▶ -\text{define}\cdot\text{B\_vector\_field}\cdot\text{F\_dense}\cdot$(\text{POLY\_Q256\_DEG30\_DENSE})\cdot-\text{end}\backslash
  ▶ ▶ ▶ ▶ ▶ -\text{with}\cdot\text{F\_do}\backslash
  ▶ ▶ ▶ ▶ ▶ ▶ -\text{finite_field_activity}\backslash
  ▶ ▶ ▶ ▶ ▶ ▶ ▶ -\text{polynomial_division}\cdot\text{A\_B}\cdot-\text{end}
```

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="\n62100-\n06210-\n006210-\n000621"

RREF_RS_8_weight Enumerator:
▷ $\$(ORBITER_PATH)oriter.out--v.2-$
▷ ▷ -define:F:-finite_field-q.8-end-$
▷ ▷ -define:V:-vector:-format.5:-field:F-$
▷ ▷ ▷ -compact:$\$(CODE_RS_8)-$
▷ ▷ -end-$
▷ ▷ -with:F:-do-$
▷ ▷ -finite_field_activity-$
▷ ▷ ▷ -weight Enumerator:v-$
▷ ▷ -end
\end{verbatim}

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

$$\begin{bmatrix}
5 & 6 & 1 & 0 & 0 & 0 & 0 \\
0 & 5 & 6 & 1 & 0 & 0 & 0 \\
0 & 0 & 5 & 6 & 1 & 0 & 0 \\
0 & 0 & 0 & 5 & 6 & 1 & 0 \\
0 & 0 & 0 & 0 & 5 & 6 & 1 \\
\end{bmatrix}.$$ 

We use the makefile variable `CODE_RS_8` to hold this generator matrix. The following command computes the weight enumerator

```plaintext
RS_8 field reduction:
```

```plaintext
which turns out to be

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

Computing the automorphism group of the code is computationally infeasible. The next command performs field reduction on the code. This produces a $[21, 15]_2$ code.

```plaintext
RS_8 field reduction:
```
The reduced matrix is shown in Figure 9.7. Let us compute the weight enumerator of the reduced code. The command

```
RS_8_reduced="
010001100000000000000000
001110010000000000000000
110011010000000000000000
000010001100000000000000
000001110100000000000000
000110011100000000000000
000000011000000000000000
000000001100000000000000
000000000110000000000000
000000000011000000000000
```

Figure 9.7: Field reduction of a Reed-Solomon code
RREF_RS_8_reduced_weight Enumerator:

\[
\text{\textbackslash (ORBITER\_PATH)or} \text{\textbackslash (v.2.1)} \\
\text{\textbackslash (define\_F::finite\_field\_q=2\_end)} \\
\text{\textbackslash (define\_v::vector::format=15::field=F::)} \\
\text{\textbackslash (compact:\textbackslash (RS_8\_reduced))} \\
\text{\textbackslash (with\_F::do)} \\
\text{\textbackslash (finite\_field\_activity)} \\
\text{\textbackslash (weight\_enumerator\_v)} \\
\text{\textbackslash (end)}
\]

computes the weight enumerator of the binary code. It is

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

In particular, the field reduced Reed-Solomon code is not optimal. It has minimum distance three, but there are codes of minimum distance 4. Here is one. We store the code to a file and then draw the generator matrix as bitmap.

CODE_21_15_4="\n1110001000000000000000\n1101000100000000000000\n1011000100000000000000\n0111000100000000000000\n1100100000000000000000\n1010100000000000000000\n0011000000000000000000\n1001100000000000000000\n0101100000000000000000\n0011000000000000000000\n1111000000000000000000\n1100010000000000000000\n1010010000000000000000\n0110010000000000000000\n10010100000000000000001"

246
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

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

This shows that this code is a $[21, 15, 4]_2$. It is optimal.
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

\begin{verbatim}
bounds_for_d_given_n15_k6_q2:
  > $(ORBITER_PATH)orbiter.out --v 2\n  > -make_bounds_for_d_given_n_and_k_and_q 15 6 2
\end{verbatim}

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:

\begin{align*}
d_{\text{GV}} &= 5 \\
d_{\text{singleton}} &= 10 \\
d_{\text{hamming}} &= 6 \\
d_{\text{plotkin}} &= 7 \\
d_{\text{griesmer}} &= 6
\end{align*}

This shows that $5 \leq d_{\text{max}} \leq 6$. The command

\begin{verbatim}
coding_theory_bounds_q2:
  > $(ORBITER_PATH)orbiter.out --v 2 --table_of_bounds 20 2
\end{verbatim}

produces a table of bounds for binary codes with $n,k \leq 20$. A file

\begin{verbatim}
table_of_bounds_n20_q2.csv
\end{verbatim}

is computed. The command

\begin{verbatim}
GV_n15_k6_d5:
  > $(ORBITER_PATH)orbiter.out --v 2\n  > -define F : finite_field : q 2 : end\n  > -define P : projective_space : 8 F : end\n  > -with P : do\n  > -projective_space_activity : make_gilbert_varshamov_code 15 5 : end
\end{verbatim}

creates a $[16, 6, d]_2$ 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:

\begin{verbatim}
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 1 0 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
\end{verbatim}
To compute the minimum distance of the code, we do:

CODE

<table>
<thead>
<tr>
<th>G V</th>
<th>N15 K6</th>
</tr>
</thead>
<tbody>
<tr>
<td>111111111100000</td>
<td></td>
</tr>
<tr>
<td>111110000010000</td>
<td></td>
</tr>
<tr>
<td>111001100010000</td>
<td></td>
</tr>
<tr>
<td>101010100001000</td>
<td></td>
</tr>
<tr>
<td>101010100000010</td>
<td></td>
</tr>
<tr>
<td>101101001000001</td>
<td></td>
</tr>
</tbody>
</table>

GV

n15

k6
d5

weight
enumerator:

⊿ \$(ORBITER\_PATH)orbiter.out\:-v\:2\:\$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$

⊿ \$
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:
  > $(ORBITER_PATH)orbiter.out -v 6\
  > -orbiter_path=$(ORBITER_PATH)\
  > -define:G:\
  > -linear_group:-PGL:4:2:-end:\
  > -with:G:-do:\
  > -group_theoretic_activity:\
  > -poset_classification_control:\
  >  -problem_label:codes_8_4_4:\
  >  -draw:poset:\
  >  -draw_options: -embedded: -radius:250\n  >  -line_width:1.0: -spanning_tree:-end:\
  >  -report:-end:\
  > -end:\
  > -linear_codes:3:8:\
  > -end
```

classifies linear codes with redundancy 4 and minimum distance at least 4. Orbiter confirms that there is exactly one such code, and it computes the code together with the projective stabilizer. Orbiter creates the action of the group \( \text{PGL}(4,2) \) on the poset \( \Lambda_{3,3}(2) \). Using poset classification, Orbiter then produces the poset of orbits shown in Figure 9.8. 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 9.8: Orbits of $\text{PGL}(4,2)$ 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 \(\text{PG}(3, 2)\). 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 \(\text{AGL}(3, 2)\), a group of order 1344.
Chapter 10
Incidence Geometry

10.1 Diophantine Systems

Diophantine systems of equations and inequalities arise frequently in Combinatorics. In Table 10.1, Orbiter commands for creating diophantine systems are shown. In Table 10.2, Orbiter activities for diophantine systems are shown.

Suppose we want all partitions of an integer $n$ as

$$n = a_1 + a_2 + \ldots + a_k, \quad a_1 \geq a_2 \geq \cdots \geq a_k \geq 1,$$

with $a_i \in \mathbb{Z}_{>0}$. For $1 \leq j \leq n$, let

$$c_j = \# \{ i \mid a_i = j \}.$$

The following diophantine equation holds for any partition

$$\sum_{j=1}^{n} j c_j = n \quad (10.1)$$

Conversely, any partition is uniquely determined by a solution of this equation. Therefore, counting partitions of $n$ is the same as counting nonnegative integer solutions of (10.1). Let $p_n$ be the number of partitions of $n$. Suppose we wish to compute $p_{10}$. In this case, the extended coefficient matrix of the system is

$$[10 \ 9 \ 8 \ 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \mid 10].$$

Orbiter creates and solves this diophantine system using the command

```
part10:
  $(ORBITER_PATH)orbiter.out\-v\-4\$
  \-define D\-diophant\-label part10\$
  \-coefficient_matrix 1\-10\"10,9,8,7,6,5,4,3,2,1\"$
  \-RHS 10,10,1\"x_min_global 0\-x_max_global 10\$
  \-end$
  \-with D\-do$
```
<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>m n list-of-integers</td>
<td>Set the $m \times n$ coefficient matrix.</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 $\text{PG}(2,q)$. 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 $\text{PG}(2,q)$ for maximal arcs.</td>
</tr>
<tr>
<td>-override_polynomial</td>
<td>a</td>
<td>Use polynomial numerically coded as $a$ for creating $\mathbb{F}_q$.</td>
</tr>
</tbody>
</table>

Table 10.1: Orbiter Commands to create Diophantine systems
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-print</td>
<td></td>
<td>Print the system</td>
</tr>
<tr>
<td>-solve_mckay</td>
<td></td>
<td>Solve the system using McKay’s pos-solve</td>
</tr>
<tr>
<td>-solve_standard</td>
<td></td>
<td>Solve the system using the standard solver</td>
</tr>
<tr>
<td>-draw</td>
<td></td>
<td>Produce a drawing of the coefficient matrix of the system</td>
</tr>
<tr>
<td>-draw_as_bitmap</td>
<td>$w \ b$</td>
<td>Produce a bitmap drawing of the coefficient matrix of the system, using boxes of $w$ pixels with. Set the color bit-depth to $b$ ($b = 8$ or $b = 24$). The output is a bmp-file.</td>
</tr>
<tr>
<td>-perform_column_reductions</td>
<td></td>
<td>Eliminate variables which must be zero and write a reduced system</td>
</tr>
<tr>
<td>-test_single_equation</td>
<td></td>
<td>For each row of the system, compute the number of solutions of the system restricted to the nonzero coefficients.</td>
</tr>
<tr>
<td>-project_to_single_equation_and_solve</td>
<td>$i \ j$</td>
<td>Solve the system assuming the $j$th solution to the restricted system consisting of the $i$th row.</td>
</tr>
<tr>
<td>-project_to_two_equations_and_solve</td>
<td>$i \ j \ r \ m$</td>
<td>Solve the system assuming any solution to the restricted system consisting of the $i$th and the $j$-th row whose number is congruent to $r$ mod $m$.</td>
</tr>
</tbody>
</table>

Table 10.2: Orbiter activities for Diophantine systems
It also creates three csv-files: one for the coefficient matrix, one for the RHS information, and one for the bounds on the variables. After solving, we know that the number of partitions of the number 10 is 42, so $p_{10} = 42$. The sequence $p_n$ is recorded under the key A000041 in Sloane’s Handbook of integer sequences [57].
10.2 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

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

is satisfied. The equation can be used to generate all possible line types of a putative linear space. Here is an example. For \(|S| = 6\), \((10.2)\) 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

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

The Orbiter command

```
linsp6:
  > $(ORBITER_PATH)orbiter.out -v 4;
  > > -define D -diophant -label linsp6 -
  > > -coefficient_matrix 1 5 "15,10,6,3,1"
  > > -RHS "15,15,1"
  > > -x_min_global 0
  > > -x_max_global 15
  > > -end
  > > -with D -do
  > > > -diophant_activity -solve mckay
  > > > -end
```

solves the system using McKay’s program possolve [46]. The program finds 15 solutions, written to the file linsp6.sol.

Let us consider a problem from [10]. Suppose we are interested in linear spaces on 30 points with line type \((7, 5^{27}, 4^{24})\). This notation means that we assume one 7-lines, 27 5-lines and 24 4-lines. The type of a point \(P\) is the set of integers

\[
p_j = \#j\text{-lines through } P.
\]

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·-define·D·
▷ ▷ -diophant·-label·linsp30_pt_types·
▷ ▷ -coefficient_matrix·1·3·"6,4,3"·
▷ ▷ -RHS·"29,29,1"·-x_bounds·"0,1,0,27,0,24"·
▷ ▷ -end·
▷ ▷ -with·D·-do·
▷ ▷ ▷ -diophant_activity·-solve_mckay·
▷ ▷ -end
```

we determine the possibilities

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

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

```
linsp30_pt_distribution:
▷ $(ORBITER_PATH)orbiter.out·-v·4·-define·D·
▷ ▷ -diophant·-label·linsp30_pt_distribution·
▷ ▷ -coefficient_matrix·6·4·
▷ ▷ ▷ "1,1,1,1,1,0,0,5,2,5,2,1,5,3,7,10,1,10,1,0,10,3,21"·
```

258
we determine the possibilities

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

A partial linear space is a set system on a fixed set $V$. We write $\mathcal{L} = (V, \mathcal{B})$, where $\mathcal{B}$ is a set of distinct subsets of $V$, called lines. The members of $V \cup \mathcal{B}$ are called elements. For two elements $x, y$, we say that $x$ is incident with $y$, written $xIy$, if either $x \in y$ or $y \in x$. We require that any line has at least two points and any two points are contained in at most one line. A decomposition of a linear space is a partition $\Pi = (C_1, \ldots, C_n)$ of $V \cup \mathcal{B}$ such that each $C_i$ either is a subset of $V$ or a subset of $\mathcal{B}$. A decomposition is called tactical if for all $i$, the incidence number

$$\iota(C_i, C_j) = \# \{ y \in C_j, xIy \}$$

does not depend on the choice of $x \in C_i$. Any linear space has a tactical decomposition, as the discrete partition (every element is in its own class) is tactical. Let $\text{Aut}(\mathcal{L})$ be the automorphism group of the linear space, which is the subgroup of $\text{Sym}(V)$ which preserves incidence. For $\alpha \in \text{Aut}(\mathcal{L})$ we say that the decomposition $\Pi$ preserves $\alpha$ if $\alpha$ fixes every class of $\Pi$. For $A \leq \text{Aut}(\mathcal{L})$, we say that $\Pi$ preserves $A$ if $\Pi$ preserves every element $\alpha \in A$. Mostly, we are interested in those decompositions $\Pi$ which preserve $\text{Aut}(\mathcal{L})$. In light of this, the discrete decomposition is not that interesting.

Any linear space has a coarsest tactical decomposition that preserves its automorphism group: The orbit partition of the automorphism group acting on $V \cup \mathcal{B}$ will do. Up to ordering of the classes, the coarsest tactical refinement is unique. Computing the orbit decomposition is challenging as it involves computing the automorphism group. Computationally, there are easier ways to get to admissible decompositions. One is by means of successive refinements. If a class $C_i$ does not have the property that $\iota(C_i, C_j)$ is well-defined for all $x \in C_i$, then a refinement of $C_i$ will do. The coarsest refinement of $C_i$ has the property that if $C_i$ preserves some group $A$ then the refinement will do, too. This shows that there is an algorithm to compute a tactical decomposition of any given linear space $\mathcal{P}$. Simply start with the decomposition of two classes, one the set of points and one the set of blocks, and refine. The output may or may not be equal to the decomposition arising from the orbit partition of $\text{Aut}(\mathcal{L})$.

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 -geometry_builder option can answer these kinds of questions.

The command

```
geo_10.3:
  ▶ $(ORBITER_PATH) orbiter.out -v 2 \n  ▶ ▶ -define Test_lines -set -loop 4 11 1 -end \n  ▶ ▶ -geometry Builder -V 10 -B 10 -TDO 3 -fuse 1 \n  ▶ ▶ ▶ -fname GEO-10.3 \n  ▶ ▶ ▶ -test Test_lines \n  ▶ ▶ -end
```

260
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 starts 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 52 56 57 62 66 68 73 77 79 84 86 89 95 98 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 68 73 77 79 84 86 89 95 98 99
0 1 2 10 13 14 20 25 26 31 33 35 41 47 48 52 54 57 62 66 68 73 76 79 85 88 89 96 97 98
0 1 2 10 13 14 20 25 26 31 33 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
0 1 2 10 13 14 20 25 26 31 33 33 35 41 47 48 52 54 57 62 66 69 73 78 79 84 86 88 95 97 99
120, 24, 12, 10, 6, 4^2, 3^2, 2
```

Two further files are written, containing the lines of the incidence geometry. The file $10_3$.blocks:

```
10 10 3
0 15 26 44 51 68 81 109 114 116
0 15 26 46 49 68 81 109 114 116
0 15 26 46 49 68 83 106 115 116
0 15 26 46 52 69 77 106 114 116
0 15 26 46 56 69 80 101 106 119
0 15 26 46 56 69 80 103 104 119
0 15 26 46 56 69 80 103 107 117
0 15 26 46 56 72 80 93 106 119
0 15 26 46 56 72 81 93 105 119
0 15 26 46 56 74 79 93 105 119
-1 10
120, 24, 12, 10, 6, 4^2, 3^2, 2
```

contains the blocks as ranked 3-subsets of a 10-element set. The file $10_3$.blocks_long contains the list of blocks written out.

It is possible to create graphical representations of the search tree. The command below does so for the example that we just did. Note the additional option -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.
In a second invocation of Orbiter, the `-tree_draw` command is used to draw a tree from
the file `10_3_tree.txt` that was just created. The green nodes are nodes that are accepted.
The red nodes are nodes that are rejected. This means they represent geometries that have
been seen before. The 10 green nodes at the very bottom of the diagram represent the 10
$10^3$ configurations.

```
geo_10_3_tree:
  ➞ $(ORBITER_PATH)orbiter.out-v.2\n  ➞ -define:Test_lines--set:-loop-0.11.1--end\n  ➞ -geometry_builder:-V-10-B-10-TDO-3--fuse-1\n  ➞ -fname:GEO-10_3\n  ➞ -search_tree\n  ➞ -test:Test_lines\n  ➞ -end
  ➞ $(ORBITER_PATH)orbiter.out-v.2\n  ➞ -draw_options:-embedded--radius-50\n  ➞ -xin:10000,-yin:10000\n  ➞ -xout:1000000,-yout:500000\n  ➞ -nodes_empty\n  ➞ -scale-0.5--line_width-0.3--end\n  ➞ -tree_draw:10_3_tree.txt
  ➞ pdflatex:10_3_tree_draw.tex
  ➞ open:10_3_tree_draw.pdf
```

The resulting tree is shown in Figure 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
```
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 $15_3$, only one of them has girth 4. This is the Cremona Richmond configuration. It is associated to a cubic surface. The following command classifies all configurations $15_3$:

```
geo_15_3:
  $(ORBITER_PATH)orbiter.out --v.2 \n  -define Test_lines --set --loop 4 16 1 --end \n  -geometry builder \n  -define 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 $15_3$ 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 \n  -define Test_lines --set --loop 4 16 1 --end
```
-geometry_builder:\
  -V:15:-B:15:-TDO:3:\
  -fuse:1:-fname:GEO:15_3.g4:\
  -girth:4:\
  -test:Test_lines:\
  -end
### 10.4 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

```plaintext
design PG 2 3:
$\text{(ORBITER PATH)orbiter.out}\cdot-v\cdot-8\cdot$
$\triangleright \quad -create_design\cdot-q\cdot3\cdot-family\cdot PG\cdot2\cdotq\cdot-end$
```

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:

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

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

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

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. [36]). For each pair of $t$-orbit and $k$-orbit, for instance with representatives $T$ and $K$, say, we count the number of elements in the orbit of $K$ which contain $T$. The rows of the matrix are in correspondence to the $t$-orbits, while the columns are in correspondence to the $k$-orbits. The matrix entry $a_{ij}$ is the number just defined where $T$ is the representative of the $i$-th orbit on $t$-sets, and where $K$ is the representative of the $j$-th orbit on $k$-sets. Let $M_{t,k}(G)$ be the Kramer-Mesner matrix for the group $G \leq \text{Sym}(V)$ defined in this way. The $t-(v,k,\lambda)$ designs invariant under $G$ are in one-to-one correspondence to the solutions of

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

where $x$ is a column vector of zeros and ones and $1$ is the column vector of all ones. The length of $x$ is the number of $k$-orbits of $G$ on $V$, while the length of $1$ is the number of $t$-orbits of $G$ on $V$. Any vector $x$ satisfying the matrix equation corresponds to a design.
invariant under $G$. Simply take the blocks of the design to be the union of those orbits of $G$ on $k$-subsets whose associated entry in $x$ is one. We assume the group $\text{PGL}(2,32)$ in the action on points of the projective line $\text{PG}(1,32)$ over the field $\mathbb{F}_{32}$. The parameters of the design are $7-(33,8,10)$, that is, each 7-subset of $\text{PG}(1,32)$ is covered exactly 10 times by the chosen 8-subsets comprising the design. The first orbiter command creates the group $\text{PGL}(2,32)$ and computes the Kramer-Mesner matrix

$$M_{7,8}(\text{PGL}(2,32)).$$

The number of 7-orbits is 32. The number of 8-orbits is 97. Correspondingly, the Kramer-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:
  $(\text{ORBITER\_PATH})$oribiter.out--v.3\$
  \text{define\_G\_linear\_group\_PGGL\_2\_32\_end}\$
  \text{with\_G\_do}\$
  \text{group\_theoretic\_activity}\$
  \text{poset\_classification\_control}\$
  \text{problem\_label\_KM\_PGGL\_2\_32\_W\_depth\_8}\$
  \text{Kramer\_Mesner\_matrix\_7\_8}\$
  \text{draw\_poset}\$
  \text{draw\_options\_embedded\_sideways\_radius\_50}\$
  \text{scale\_0.5\_line\_width\_0.3\_end}\$
  \text{end}\$
```

Figure 10.2: Kramer-Mesner matrix $M_{7,8}(\text{PGL}(2,32))$
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.
10.5 Design Theory – Large Sets

Fix a set of size $v$ and an integer $k$ with $1 < k < v$. Is it possible to partition the set of $k$-subsets of $v$ into designs, all with the same parameters? If so, the resulting set of designs is called a large set (of designs). So, a large set of designs is a set of designs, all of the same types, on a fixed $v$-element set whose block sets are pairwise disjoint and partition the set of $k$-subsets. Let us see how Orbiter can help construct and classify small large sets.

Suppose we consider $AG(2, 3)$, the affine plane of order 3. It is a configuration with 9 points, 12 lines, 4 lines on each point and 3 points on each line. To see if it is unique, we use the following command:

```
geo_9.4_12.3:
  > $(ORBITER_PATH)orbiter.out:-v.2:\
  >  -geometry_builder:\
  >  > -V.9-B.12:\
  >  > -TDO.4-fuse.1:\
  >  > -fname_GEO_AG_2.3:\
  >  > -test.3,4,5,6,7,8,9.\n  >  > -end
```

The command produces the file $AG_2.3.inc$, which contains the following lines:

```
9 12 36
0 1 2 3 12 16 18 24 31 32 33 37 40 43 46 49 53 56 59 62 64 69 71 74 78 80 82 87 89 93 94 99 102 103 107
-1 1
432
```

This shows that the design is unique, and has an automorphism group of order 432. We can use the following command to check the automorphism group:

```
geo_9.4_12.3_c:
  > $(ORBITER_PATH)orbiter.out:-v.2:\
  >  -define.C:-combinatorial_objects:\
  >  > -file_of_incidence_geometries:\
  >  >  AG_2.3.inc:9.12.36:\
  >  >  -end:\
  >  >  -with.C:-do:\
  >  >  -combinatorial_object_activity:\
  >  >  -canonical_form:\
  >  >  -classification_prefix_AG_2.3:\
  >  >  -save_ago:\
  >  >  -end
```

This command writes several files: $AG_2.3.blocks$ contains the list of blocks as ranks of $k$-subsets. $AG_2.3.blocks_long$ contains the list of blocks as $k$-subsets. $AG_2.3_ago.csv$ contains the automorphism group order of the design. For the following commands, we will
treat blocks of the design as sets of ranks of $k$-subsets. We can now create a table of all
designs $AG(2, 3)$, as orbit under the group $Sym(9)$. The following command does that:

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

The following command creates the design table:

```
LS_AG_2_3_design_table_create:
  > $(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
```

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
  > -define Gamma_graph
  > -disjoint_sets_graph
  > AG_2_3_design_table.csv
  > -end
  > -with Gamma -do
  > -graph_theoretic_activity
  > -save
  > -end
  > -with Gamma -do
  > -graph_theoretic_activity
  > -find_cliques -target_size 7 -end
  > -end
  > -print_symbols
```

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
  > -define D_design_list_of_blocks 9 3
  > -define $(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
```
The final step to classify the large sets up to isomorphism will be discussed in Section 12.4.
10.6 Design Theory – Delandtsheer-Doyen

Delandtsheer and Doyen in [19] 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. Orter can help to classify small examples. Let us consider an example. Suppose we want to classify all designs on 21 points with blocks of size $k = 5$ invariant under a cyclic group of order 21 preserving a grid of type $3 \times 7$. To this end, we consider the group $\text{AGL}(1,3) \times \text{AGL}(1,7)$. The subgroup is generated by the map

$$(\tau_1, \tau_2), \mathbb{Z}_3 \times \mathbb{Z}_7 \to \mathbb{Z}_3 \times \mathbb{Z}_7,$$

where

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

With blocks of size 5, we cover 10 pairs each. The group of order 21 allows to cover each of the $210 = \binom{21}{2}$ pairs exactly once using a single orbit of a block. The question remains to construct all blocks and to classify the resulting designs. The Desarguesian plane $\text{PG}(2,4)$ provides a solution. The question is to decide whether there are any other, nonisomorphic designs. The following Orbiter commands can be used:

```
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\n   -depth.5\n   -mask_label."no_mask"

DD_PP4:
   $(\text{ORBITER_PATH})orbiter.out.-v.6.\n   -Delandtsheer_Doyen.$(PP4).$(PP4_GROUP1).$(PP4_MASK1).\n   -end.\n
DD_PP4_system:
   $(\text{ORBITER_PATH})orbiter.out.-v.4.\n   -define.D-diophant.-label.PP4.\n   -problem_of_Steiner_type.10.PP4_pair_covering.csv.\n   -has_sum.1.\n   -end.\n   -with.D-do.\n   -diophant_activity.-solve_mckay.\n   -end
```

272
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.
10.7 Tactical Decompositions

Table 10.3 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><code>-lambda3</code></td>
<td>$\lambda_3 s$</td>
<td>Refine as 3-design with $\lambda_3$ and with block-size $s$.</td>
</tr>
<tr>
<td><code>-solution</code></td>
<td>$i$ <code>fname</code></td>
<td>Use solutions to system $i$ from file <code>fname</code>.</td>
</tr>
<tr>
<td><code>-range</code></td>
<td>$f l$</td>
<td>Refine cases $i$ with $f \leq i &lt; f + l$ only.</td>
</tr>
<tr>
<td><code>-select</code></td>
<td><code>label</code></td>
<td>Select the case for refinement by label.</td>
</tr>
<tr>
<td><code>-o1</code></td>
<td>$s$</td>
<td>Omit $s$ variables from the first refinement system.</td>
</tr>
<tr>
<td><code>-o2</code></td>
<td>$s$</td>
<td>Omit $s$ variables from the second refinement system.</td>
</tr>
<tr>
<td><code>-D1_upper_bound_x0</code></td>
<td>$b$</td>
<td>Add the bound $x_0 \leq b$ in the first refinement.</td>
</tr>
<tr>
<td><code>-reverse</code></td>
<td></td>
<td>Sort the distributions in reverse order.</td>
</tr>
<tr>
<td><code>-reverse_inverse</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>-nopacking</code></td>
<td></td>
<td>Do not use packing inequalities.</td>
</tr>
<tr>
<td><code>-dual_is_linear_space</code></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><code>-geometric_test</code></td>
<td></td>
<td>Subject the distributions to the geometric test.</td>
</tr>
<tr>
<td><code>-once</code></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><code>-mckay</code></td>
<td></td>
<td>Use McKay’s solver instead (by default, a lexicographic solver is used).</td>
</tr>
<tr>
<td><code>-input_file</code></td>
<td><code>fname</code></td>
<td>Specify the input TDO-file for refinement.</td>
</tr>
</tbody>
</table>

Table 10.3: 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 -v -4 -tdo_refinement \
$ -dual -input_file max.arc_q16_r4.tdo -dual_is_linear_space -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 -v -4 -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
```

276
col_scheme at level 4:
is 2 x 2

<table>
<thead>
<tr>
<th>221_{1}</th>
<th>52_{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>52_{0}</td>
<td>4 0</td>
</tr>
<tr>
<td>221_{3}</td>
<td>13 17</td>
</tr>
</tbody>
</table>

extra_col_scheme at level 3:
is 1 x 2

<table>
<thead>
<tr>
<th>221_{1}</th>
<th>52_{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>273_{0}</td>
<td>17 17</td>
</tr>
</tbody>
</table>
10.8 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 $Q = 4$):

```
desarguesian_spread_in_PG_3_2:
▷ $\$(\text{ORBITER\_PATH})\text{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
```

The cheat sheet contains the following spread:

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

Spread element 1 is $(0, 1) =$ \[
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix}
\]
34

Spread element 2 is $(1, 1) =$ \[
\begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1
\end{bmatrix}
\]
9

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

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

The following command creates the Desarguesian plane-spread in $\text{PG}(5, 2)$:

```
desarguesian_spread_in_PG_5_2:
▷ $\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3:\$
```

278
Spread element 0 is $(1,0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

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

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

Spread element 3 is $(2,1) = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 671 & 1 & 394 & 0 & 189 & 1 \end{bmatrix}$

Spread element 4 is $(3,1) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 562 & 1 & 394 & 0 & 189 & 1 \end{bmatrix}$

Spread element 5 is $(4,1) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1040 & 1 & 394 & 0 & 189 & 1 \end{bmatrix}$

Spread element 6 is $(5,1) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 792 & 1 & 394 & 0 & 189 & 1 \end{bmatrix}$

Spread element 7 is $(6,1) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1161 & 1 & 394 & 0 & 189 & 1 \end{bmatrix}$

Spread element 8 is $(7,1) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 373 & 1 & 394 & 0 & 189 & 1 \end{bmatrix}$
Apart from the first spread element, the left halves of the generator matrices of the subspaces in the Desarguesian spread are the elements of $\mathbb{F}_8$ in a matrix representation over $\mathbb{F}_2$.

Two $t$-spreads are isomorphic if there is a collineation which maps one to the other. The classification problem for $t$-spreads is the problem of determining a complete set of pairwise non-isomorphic $t$-spreads. Orbiter can be used to classify spreads for small parameters. For instance, the command

```
spreads16_4:
  $(ORBITER_PATH)orbiter.out -v 6 \n  -orbiter_path$(ORBITER_PATH) \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  -spreadclassify 2 -problem_label spreads 4_2 \n  -W -depth 17 -drawPoSet \n  -draw_options -radius 20 \n  -nodes_empty -line_width 0.2 -embedded \n  -end \n  -report \n  -end
```

classifies the line-spreads of PG(3, 4) under the action of PΓL(4, 4). Under the André, Bruck-Bose construction [3, 14], 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:

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:

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

There are 0 extensions
Number of generators 3

**Orbit 1 / 3 at Level 17**

Node number: 1127

\{0, 25, 50, 75, 90, 107, 140, 157, 179, 204, 213, 238, 265, 282, 299, 316, 356\}_{81600}

Strong generators for a group of order 81600:

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

\{1,0,0,0,0,1,0,0,0,0,0,3,0,0,0,0,0,3,0, \\
1,0,0,0,0,1,0,0,0,0,0,2,3,0,0,0,1,1,0, \\
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,
\}


281
There are 0 extensions
Number of generators 7

**Orbit 2 / 3 at Level 17**

Node number: 1128

\(\{0, 25, 50, 75, 90, 108, 122, 140, 158, 183, 199, 217, 233, 250, 268, 312, 345\}_{576}\)

Strong generators for a group of order 576:

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

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

There are 0 extensions
Number of generators 6
The three spreads in $\text{PG}(3, 4)$ can be distinguished by their stabilizer orders. Table 10.4 lists the line spreads in $\text{PG}(3, 4)$ according to their orbiter catalogue number (OCN). Table 10.5 lists the solid spreads in $\text{PG}(7, 2)$ according to their orbiter catalogue number (OCN).

| OCN | |Aut| |Name                     |
|-----|-----|---|-------------------------|
| 0   | 1200|   | Hall spread             |
| 1   | 81600|  | Desarguesian spread     |
| 2   | 576 |   | Semifield spread        |

Table 10.4: Spreads in $\text{PG}(3, 4)$ in the Orbiter Catalogue

<table>
<thead>
<tr>
<th>OCN</th>
<th></th>
<th>Aut</th>
<th></th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1728</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>288</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>244800</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10.5: Spreads in $\text{PG}(7, 2)$ in the Orbiter Catalogue
10.9 Translation Planes

Via the André, Bruck, Bose construction (cf. [3, 14]), 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:
```

produces the Hall plane of order 16. Remember from Table 10.4 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:

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

subspace 1 / 17 is 356:

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

subspace 2 / 17 is 25:

creates the Hall plane of order 16. Remember from Table 10.4 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:
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:
\[
\begin{bmatrix}
\omega & 0 & 0 & 0 \\
0 & \omega & 0 & 0 \\
0 & 0 & \omega & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \omega
\end{bmatrix},
\begin{bmatrix}
\omega^2 & \omega & \omega & 1 \\
0 & 0 & \omega^2 & 1 \\
0 & 0 & 0 & \omega
\end{bmatrix}
\]
\[
\begin{bmatrix}
\omega^2 & 0 & 0 & 0 \\
\omega^2 & 0 & 0 & 0 \\
0 & \omega^2 & \omega & 0 \\
\omega^2 & 0 & 0 & \omega^2
\end{bmatrix},
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\omega & \omega & 0 & 0 \\
1 & 0 & \omega^2 & 0 \\
\omega & \omega & 1 & 1
\end{bmatrix},
\begin{bmatrix}
1 & \omega^2 & \omega & 1 & 0 \\
0 & \omega^2 & 0 & 1 & 0 \\
\omega & 1 & \omega^2 & 0 \\
0 & 1 & 0 & \omega^2 & 0
\end{bmatrix}
\]

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

Tactical decomposition schemes:

\[
\begin{array}{c|ccc}
\rightarrow & 80_1 & 192_5 & 1_4 \\
256_0 & 5 & 12 & 0 \\
5_3 & 16 & 0 & 1 \\
12_2 & 0 & 16 & 1 \\
\end{array}
\quad
\begin{array}{c|ccc}
\downarrow & 80_1 & 192_5 & 1_4 \\
256_0 & 16 & 16 & 0 \\
5_3 & 1 & 0 & 5 \\
12_2 & 0 & 1 & 12 \\
\end{array}
\]

287
10.10 Packings

A packing of $\text{PG}(3, q)$ is a set of pairwise line-disjoint spreads of $\text{PG}(3, q)$ of size $q^2 + q + 1$. Each spread contains $q^2 + 1$ lines. A simple counting argument shows that every line is contained in exactly one spread of the packing. The classification problem for packings is the problem of determining a complete set of pairwise non-isomorphic packings. Orbiter can be used to classify packings for small parameters. It is sometimes useful to make a symmetry assumption. This means that only those packings will be found that satisfy the symmetry assumption. The reason for making such an assumption is that the problem becomes easier and hence more tractable. Often, an assumption is made that the packings are invariant under a (nontrivial) group $H$. This section describes various ways in which Orbiter can help find and classify packings, with or without symmetry assumption.

Table 10.6 list Orbiter commands related to the construction of packings with assumed symmetry.

Table 10.7 list Orbiter commands related to the construction of packings with assumed symmetry by picking long orbits.

A packing is regular if it consists solely of regular spreads. The smallest regular packings exist in $\text{PG}(3, 5)$. They were first described by Prince [53] and later placed into an infinite family by Penttila and Williams [50]. Up to isomorphism, there are exactly two regular packings in $\text{PG}(3, 5)$. Let us construct these packings. We start by making a table of all regular packings:

```
spread_table_PG_3_5_regular:
▷ -mkdir SPREAD_TABLES_5_REG
▷ $(\text{ORBITER_PATH})\text{orbiter.out}-v.6\backslash
▷ ▷ -define F:\text{finite_field}-q.5-end\backslash
▷ ▷ -define P:\text{projective_space}-3-F:-end\backslash
▷ ▷ -define T:-spread_table-P.2."12"."SPREAD_TABLES_5_REG/".\backslash
▷ ▷ -print_symbols
```

There are 155,000 packings. In the command, we rely on the classification of spreads in $\text{PG}(3, 5)$ which is built into Orbiter. The spread with orbiter catalogue number 12 is the regular spread.

We consider the projectivity of order 31 given by the matrix

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

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 10.7.</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 10.6: 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 10.7: Orbiter commands related to the construction of packings with assumed symmetry related to picking long orbits
The normalizer is a group of order 372. We encode the group and its normalizer as makefile variables:

PGL_4_5_SUBGROUP_31_ME=-PGL.4:5-\n  > -subgroup_by_generators."31"."31.1-\n  > "1,0,0,0,0,3,4,3,0,3,4,-0,3,2,3"\nPGL_4_5_SUBGROUP_31_ME_NORMALIZER=-PGL.4:5-\n  > -subgroup_by_generators."normalizer.31"."372".4-\n  > "1,0,0,0,0,4,0,0,0,0,0,0,4,".-\n  > "1,0,0,0,3,0,0,0,3,0,0,0,0,3,".-\n  > "1,0,0,0,4,0,0,0,0,2,1,0,3,2,4,".-\n  > "1,0,0,0,0,1,0,0,0,0,1,0,1,1,3",".-\n
Let $H$ be the subgroup of order 31 and let $N$ be its normalizer. Then we compute the orbits of $H$ on the regular spreads:

PG_3_5_assume_31_graph:  
  > open.H31_reduced_spread_orbits_orbits_report.pdf  
  > pdflatex.H31_line_orbits_orbits_report.tex
The command produces reports about the orbits of both $H$ and $N$ on points, lines and spreads. The following command searches all cliques of size 1 in the graph on long orbits. This is not very difficult!

```
PG_3.5_assume_31_fpc0_lo_cliques:
  $(ORBITER_PATH)orbiter.out-v.2.\n  \> -define::G::graph::load_from_file::H31_fpc0_lo.graph::end\n  \> -with::G::do\n  \>   -graph_theoretic_activity\n  \>   -find_cliques::target_size:1::end::end\n  \> -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:
  $(ORBITER_PATH)orbiter.out-v.5.\n  \> -define::F::finite_field::q:5::end\n  \> -define::P::projective_space::3:F::end\n  \> -define::T::spread_table::P:2:"12"::"SPREAD_TABLES_5_REG/".\n  \> -define::PW::packing_with_symmetry_assumption::T.\n  \>   \>    -H:"H31"::$(PGL_4.5_SUBGROUP_31_ME)::end\n  \>   \>    -N:"H31"::$(PGL_4.5_SUBGROUP_31_ME)::end\n  \>   \> -end\n  \> -define::PWF::packing_choose_fixed_points::PW:0::end\n  \> -define::L::packing_long_orbits::PWF\n  \> -orbit_length:31::clique_size:1\n  \> -read_solutions\n  \> -end\n```

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

```
PG_3.5_assume_31_classify:
  $(ORBITER_PATH)orbiter.out-v.2\n```

292
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.
### 10.11 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 [38], Penttila-Royle [49], Penttila-Law [39, 40], Betten [8], 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 10.8 shows options to create known BLT-sets. Table 10.9 shows options for known families or sporadic sets. For instance, the command

\begin{verbatim}
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
\end{verbatim}

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

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

<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 10.9 for possibilities for $F$.</td>
</tr>
</tbody>
</table>

Table 10.8: Commands for creating BLT-sets
<table>
<thead>
<tr>
<th>$F$</th>
<th>Condition</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td></td>
<td>Linear BLT-set.</td>
</tr>
<tr>
<td>Fisher</td>
<td></td>
<td>Fisher BLT-set [23].</td>
</tr>
<tr>
<td>Mondello</td>
<td>$q \equiv \pm 1 \mod 10$</td>
<td>Mondello BLT-set due to Penttila [48].</td>
</tr>
<tr>
<td>FTWKB</td>
<td>$q \equiv \pm 2 \mod 3$</td>
<td>Fisher, Thas, Walker [62], Kantor, Betten [12] BLT-set.</td>
</tr>
<tr>
<td>Kantor1</td>
<td></td>
<td>$q = p^{e}, e &gt; 1$ Kantor’s first family.</td>
</tr>
<tr>
<td>Kantor2</td>
<td></td>
<td>$q \equiv \pm 2 \mod 5$ 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 [40].</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 [40].</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 [40].</td>
</tr>
<tr>
<td>LP_71</td>
<td>$q = 71$</td>
<td>BLT-set for $q = 71$ due to Law and Penttila [40].</td>
</tr>
</tbody>
</table>

Table 10.9: Families of BLT-sets

$\triangleright \triangleright \triangleright$ -create_BLT_set:family:"Mondello"-end$
$\triangleright \triangleright \triangleright$ -end
$\triangleright \text{pdflatex:BLT_Mondello_q11.tex}$
$\triangleright \text{open:BLT_Mondello_q11.pdf}$

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)

$\begin{array}{c|c}
\rightarrow & 18_1 \\
\downarrow & 18_1 \\
\hline
12_0 & 6 \\
\hline
\end{array}$

$\begin{array}{c|c}
\rightarrow & 18_1 \\
\downarrow & 18_1 \\
\hline
12_0 & 6 \\
\hline
\end{array}$

$C_0 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}_{12}$
$C_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}_{18}$

The classification of BLT-sets proceeds via the poset of partial BLT-sets. For more details, see [1, 8, 38]. The following command classifies the BLT-sets in $\mathbb{Q}(4, 13)$:

```
BLT_13_deep_14:
▷ $(\text{ORBITER\_PATH})\text{or} \cdot t.\text{out} -v-2 \cdot \$
▷ ▷ -define:\text{F} -finite\text{\_field} -q\_13 -end\$
▷ ▷ -define:\text{O} -orthogonal\text{\_space} -0\_5\text{\_F} -end\$
▷ ▷ -with:\text{O} -do-orthogonal\text{\_space}_activity\$
▷ ▷ ▷ -BLT\_set\_starter\_14 -problem\_label\_BLT\_q13 -W -depth\_14 -end\$
```

296
-end
Chapter 11

Graph Theory

11.1 Creating Graphs

Table 11.1 shows some Orbiter commands to create graphs.

For instance, the command

```
Cycle_13:
  $(ORBITER_PATH)orbiter.out-v.2-
  -define Gamma -graph-
  -cycle 13-
  -end
```

creates the cycle graph of degree 13.

There are many ways to read graphs from file. One way is by means of an adjacency matrix stored as a csv file. Consider an example. Suppose we have the file `triangle_graph.csv` which contains the following adjacency matrix of a graph with 3 vertices

```
0,1,1
1,0,1
1,1,0
```

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:
  $(ORBITER_PATH)orbiter.out-v.6-
  -define G -graph-
  -load_csv_no_border-
  triangle_graph.csv-
  -end
```
<table>
<thead>
<tr>
<th>Key</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-load</td>
<td>filename</td>
<td>Read a graph from file in Orbiter format.</td>
</tr>
<tr>
<td>-load_csv_no_border</td>
<td>filename</td>
<td>Read a graph from a csv file. Ignore the first row and first column.</td>
</tr>
<tr>
<td>-load_dimacs</td>
<td>filename</td>
<td>Read a graph from file in dimacs format.</td>
</tr>
<tr>
<td>-edge_list</td>
<td>n list-of-edges</td>
<td>Create a graph on n vertices from a list of edges as ranked pairs.</td>
</tr>
<tr>
<td>-edges_as_pairs</td>
<td>n edges-as-pairs</td>
<td>Create a graph on n vertices from a list of edges as pairs.</td>
</tr>
<tr>
<td>-cycle</td>
<td>n</td>
<td>Cycle graph on n vertices.</td>
</tr>
<tr>
<td>-Hamming</td>
<td>n q</td>
<td>Hamming graph $H(n,q)$</td>
</tr>
<tr>
<td>-Johnson</td>
<td>n k s</td>
<td>Johnson graph</td>
</tr>
<tr>
<td>-Paley</td>
<td>q</td>
<td>Paley graph</td>
</tr>
<tr>
<td>-Sarnak</td>
<td>p q</td>
<td>Lubotzky-Phillips-Sarnak graph [42]</td>
</tr>
<tr>
<td>-Schlaefli</td>
<td>q</td>
<td>Schlaefli graph</td>
</tr>
<tr>
<td>-Shrikhande</td>
<td></td>
<td>Shrikhande graph</td>
</tr>
<tr>
<td>-Winnie_Li</td>
<td>q i</td>
<td>Winnie-Li graph [41]</td>
</tr>
<tr>
<td>-Grassmann</td>
<td>n k q r</td>
<td>Grassmann graph</td>
</tr>
<tr>
<td>-coll_orthogonal</td>
<td>$\epsilon$ d q</td>
<td>Collinearity graph of $O^{\epsilon}(d,q)$</td>
</tr>
<tr>
<td>-triheral_pair_disjointness_graph</td>
<td></td>
<td>Triheral pair disjointness graph</td>
</tr>
<tr>
<td>-non_attacking_queens_graph</td>
<td>n</td>
<td>Create the graph for non-attacking queens on a $n \times n$ chess board.</td>
</tr>
<tr>
<td>-subset</td>
<td>label labeltex subset</td>
<td>Define vertex coloring with two colors based on a subset of vertices.</td>
</tr>
</tbody>
</table>
| -orbital_graph             | $G i$           | Define orbital graph from the $i$-th orbit of the group $G$ acting on pairs. See Section ??.
| -collinearity_graph        | inc-matrix      | Collinearity graph of the given incidence matrix.                       |
| -chain_graph               | P1 P2           | Chain graph with respect to the partitions P1 and P2.                    |

Table 11.1: Orbiter commands to define graphs
This will create the three-cycle graph.

The command

```
Chain_232:
  $(ORBITER_PATH)orbiter.out -v 2 \n  -define P1 -vector -dense 2,3,2 -end \n  -define P2 -vector -dense 2,3,2 -end \n  -define Gamma -graph \n  -chain_graph P1 P2 \n  -end \n  -with Gamma -do \n  -graph_theoretic_activity -export_csv \n  -end \n  -with Gamma -do \n  -graph_theoretic_activity -properties \n  -end
```

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

301
<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 11.4.</td>
</tr>
<tr>
<td>-export_magma</td>
<td></td>
<td>Export to Magma [13].</td>
</tr>
<tr>
<td>-export_maple</td>
<td></td>
<td>Export to Maple [45].</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>file</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 12.7.</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>
</tbody>
</table>

Table 11.2: Graph Theoretic Activities

11.2 Graph Theoretic Activities

Graph theoretic activities allow us to perform tasks on graphs. Table 11.2 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:
\$ (ORBITER_PATH) orbiter.out -v.6 \n\$ -define G -graph \n\$ -load_csv_no_border \n\$ -triangle_graph.csv \n\$ -end \n\$ -with G -do \n\$ -graph_theoretic_activity -properties \n\$ -end
```

computes the degree type of the graph. This is the distribution of degrees in the degree sequence of the graph. It prints the distribution of degree values in exponential notation.
Multiplicities are indicated as exponent. Since there are three vertices of degree 2, the degree sequence is printed as $2^3$.

We can export the adjacency matrix and create a bitmap drawing like so:

```
Cycle_13_draw:
  ▶ $(ORBITER_PATH)orbiter.out:-v.2.-define:Gamma:-graph:-cycle.13.-end\n  ▶ ▶ -with:Gamma:-do\n  ▶ ▶ -graph_theoretic_activity:-export_csv:-end\n  ▶ ▶ -with:Gamma:-do\n  ▶ ▶ -graph_theoretic_activity:-export_graphviz:-end
  ▶ $(ORBITER_PATH)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
  ▶ #dot:-Tpng:Cycle.13.gv>Cycle.13.png
  ▶ #twopi:-Tpng:Cycle.13.gv>Cycle.13.png
  ▶ #open:Cycle.13_draw.bmp
  ▶ #pdflatex:Cycle.13_report.tex
  ▶ #open:Cycle.13_report.pdf
```

The command first creates the cycle graph of order 13, and then exports the adjacency matrix as csv file. It then draws the adjacency matrix as a bitmap graphics.

Suppose we want to compute the eigenvalues of the adjacency matrix of a graph. The command `-eigenvalues` can help:

```
Cycle_9_eigenvalues:
  ▶ $(ORBITER_PATH)orbiter.out:-v.2\n  ▶ ▶ -define:Gamma:-graph\n  ▶ ▶ ▶ -cycle:9\n  ▶ ▶ ▶ -end\n  ▶ ▶ -with:Gamma:-do\n  ▶ ▶ -graph_theoretic_activity:-eigenvalues:-end
```

computes the eigenvalues of the 9-cycle.

The command

```
petersen:
  ▶ $(ORBITER_PATH)orbiter.out:-v.2\n  ▶ ▶ -define:G:-graph:-Johnson:5:2:0:-end\n  ▶ ▶ -with:G:-do\n  ▶ ▶ -graph_theoretic_activity:-export_csv:-end\n  ▶ ▶ -with:G:-do\n  ▶ ▶ -graph_theoretic_activity:-export_graphviz:-end\n```
creates the Johnson graph $J(5, 2, 0)$ also known as the Petersen graph.

Small graphs can be encoded manually. This means that the edges are specified individually. For instance, the graph

![Johnson graph](image)

can be created using the command

```bash
small_graph:
$ (ORBITER_PATH)orbiter.out -v 2 -draw_matrix -with-G -do -graph_theoretic_activity -save -end
$ (ORBITER_PATH)orbiter.out -v 2 -draw_matrix -with-G -do -input_csv_file Johnson_5_2_0.csv -box_width 40 -bit_depth 24 -partition 4 "10", "10" -end
dot -Tpng Johnson_5_2_0.gv > Johnson_5_2_0.png
```

The graph is stored as file `graph_v5_e7.colored_graph`.

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
Table 11.3: Orbiter commands to modify graphs

<table>
<thead>
<tr>
<th>Key</th>
<th>Arguments</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-complement</td>
<td></td>
<td>Complementary graph.</td>
</tr>
<tr>
<td>-distance_2</td>
<td></td>
<td>Distance two graph: Two vertices are adjacent if and only if they have distance two in the original graph.</td>
</tr>
</tbody>
</table>

The incidence matrix to file. Here, we read the incidence matrix from file and create the collinearity graph of it:

```
PGO_5_2_collinearity_graph::~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 \ 
>   -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.

Table 11.3 shows some Orbiter commands to modify graphs. The commands replace the given graph by the graph obtained after applying the specified modification.
<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 11.4: Options for classifying graphs

11.3 Classification of Graphs and Tournaments

Table 11.4 lists the Orbiter commands to classify graphs and tournaments. The following command classifies all graphs on 4 vertices:

```
graph_classify_4:
  $(ORBITER_PATH)orbiter.out -v 2 \n  -define GC -graph_classification \n  -n 4 \n  -poset_classification_control \n  -draw_options -radius 250 -embedded -end \n  -end \n  -end \n  -end \n  -with GC -do \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 \n  pdflatex graphs_v4_rep_3_2.tex \n  open graphs_v4_rep_3_2.pdf \n  #pdflatex graphs_v4_poset_detailed_lvl_6.tex \n  #open graphs_v4_poset_detailed_lvl_6.pdf \n  #pdflatex graphs_v4_poset_lvl_6.tex \n  #open graphs_v4_poset_lvl_6.pdf
```

The next command classifies all tournaments on 4 vertices:

```
tournament_classify_4:
  $(ORBITER_PATH)orbiter.out -v 2 \n  -define GC -graph_classification \n  -n 4 -tournament \n```
Figure 11.1: The four isomorphism types of tournaments on 4 vertices

```
> > > -poset_classification_control.\
> > > -problem_label:tournament_4:-depth:6:-draw_poset.\
> > > -draw_options:-radius:250:-embedded:-end.\
> > > -end.\
> > -with:GC:-do-\
> > -graph_classification_activity.\
> > > -draw_options:-embedded:-radius:400-\
> > > -line_width:2:-scale:0.15:-end-\
> > > -draw_graphs_at_level:6-\
> > > -end.\
> > -print_symbols
```

Figure 11.1 shows the resulting list of 4 tournaments.

The next command classifies all cubic graphs on 8 vertices:

```
graph_classify_8_r3:
> $(ORBITER_PATH)orbiter.out:-v:3-\n> -define:GC:-graph_classification-\n> > -n:8:-regular:3-\n> > -poset_classification_control-\n> > > -problem_label:graphs.v8_r3:-depth:12:-draw_poset-\n> > > -draw_options:-radius:250-\n> > > > -line_width:0.2:-embedded:-end-\n> > > -end-\n> > -with:GC:-do-\
> > -graph_classification_activity-\n> > > -draw_options:-embedded:-radius:400-\
> > > -line_width:2:-end-\
> > > -draw_graphs_at_level:12-\
> > > -end-\n> > -print_symbols
```
### Table 11.5: Clique Finding Options

<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>

### 11.4 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 11.2 can be used to find all cliques in a graph. Additional options for this command are shown in Table 11.5. For instance, the cliques of size 3 in the graph `graph_v5_e7.colored_graph` from Section 11.1 can be found using

**small_graph_cliques:**

```bash
$ (ORBITER_PATH) orbiter.out -v.10 \n   -define G: -graph -load_from_file graph_v5_e7.colored_graph -end \n   -with G: -do \n   -graph_theoretic_activity -find_cliques -target_size 3 -end
```

This command finds three cliques of size 3.

It is also possible to classify all cliques under the automorphism group of the graph. This is a multi-step process, though. At first, the automorphism group of the graph has to be computed. Then, poset classification can be invoked to classify the cliques of a certain size. Here is an example. We wish to classify the cliques in the Paley graph of order 13. The first command creates the graph and computes the automorphism group:

**Paley_13_aut:**

```bash
$ (ORBITER_PATH) orbiter.out -v.2 \n   -define Gamma: -graph -Paley.13 -end \n   -with Gamma: -do \n   -graph_theoretic_activity -automorphism_group
```

308
The command writes a file `Paley_13_group.makefile`, shown below:

```
Paley_13:
  $(ORBITER_PATH)orbiter.out-v.2\n  -define.gens.vector-file:Paley_13.gens.csv-end\n  -define.G:permutation_group\n  -bsgs:Paley_13:"Paley\_13".13-78:"0,1".3-gens-end\n```

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\n  -define.gens.vector-file:Paley_13.gens.csv-end\n  -define.G:permutation_group\n  -bsgs:Paley_13:"Paley\_13".13-78:"0,1".3-gens-end\n  -define.Gamma:graph-Paley.13-end\n  -with.G-do\n  -group_theoretic_activity\n  -poset_classification_control\n  -problem_label:Paley13_cliques\n  -clique_test.Gamma\n  -depth.5\n  -end\n  -orbits_on_subsets.5\n  -report\n```

For comparison, the command

```
Paley_13_cliques:
  $(ORBITER_PATH)orbiter.out-v.10\n  -define.Gamma:graph-Paley.13-end\n  -with.Gamma-do\n  -graph_theoretic_activity-find_cliques-target_size.3-end\n```

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 ???. 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:

\[
\text{HJ64_cliques5:}
\]\[
\text{define Gamma - graph} \\
\text{load Group_Perm315_Orbital_3.colored_graph} \\
\text{-end} \\
\text{-with Gamma - do} \\
\text{-graph_theoretic_activity} \\
\text{-find_cliques - target size 5 - end} \\
\text{-end}
\]

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:

\[
\text{HJ64_cliques5_classify:}
\]\[
\text{define Gamma - graph} \\
\text{load Group_Perm315_Orbital_3.colored_graph} \\
\text{-end} \\
\text{define gens - vector} \\
\text{-file halljanko315_gens.csv} \\
\text{-end} \\
\text{-define G - permutation group} \\
\text{-bsgs halljanko315 "File\_halljanko315"} \\
\text{315:1209600 "0,1,42,95" - gens - end} \\
\text{with G - do} \\
\text{-group_theoretic_activity} \\
\text{-poset_classification_control} \\
\text{-problem_label HJ64_cliques} \\
\text{-clique_test Gamma} \\
\text{-depth 5} \\
\text{-end} \\
\text{-orbits_on_subsets 5} \\
\text{-report} \\
\text{-end}
\]

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:

```plaintext
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 12

Canonical Forms with Nauty

12.1 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 care isomorphic if and only if the associated colored Levi graphs are isomorphic in the sense of graph isomorphism. This allows to settle many questions associated with combinatorial object, such as isomorphism testing and determining the automorphism group.

A canonical labeling of a graph is a bijection of the vertices. The property is that if two graphs are isomorphic, then the graphs become identical if the canonical labeling permutation is applied (each graph has its own canonical labeling). It is therefore important to compute canonical forms. If there is a vertex coloring, we implicitly assume that the canonical labeling preserves the coloring.
<table>
<thead>
<tr>
<th>Modifier</th>
<th>Args</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-max_TDO_depth</td>
<td>$d$</td>
<td>Limit TDO depth to $d$ in the report.</td>
</tr>
<tr>
<td>-classification_prefix</td>
<td>prefix</td>
<td>Use the given prefix when writing files related to the classification.</td>
</tr>
<tr>
<td>-save_ago</td>
<td></td>
<td>Save the automorphism group orders to file.</td>
</tr>
<tr>
<td>-save_transversal</td>
<td></td>
<td>Save the indices of the elements chosen for the transversal.</td>
</tr>
</tbody>
</table>

Table 12.1: Orbiter commands related to canonical labelings

The graph theory package Nauty [47] provides canonical form algorithms 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 12.1 list Orbiter commands related to canonical labelings of combinatorial objects.
12.2 Canonical Forms of Objects in Projective Space

Suppose we want to compute the stabilizer of an elliptic curve. In Section 4.1, we have created an elliptic curve over $\mathbb{F}_{11}$ and stored the set of $\mathbb{F}_q$-points in the file `elliptic_curve_b1_c3_q11.txt`. The following example computes the set stabilizer of the curve. This means it computes the set stabilizer of the points on the curve. In order to do so, an input stream is created which referst to the file containing the Orbiter point ranks of points on the curve.

```
EC_canon:elliptic_curve_b1_c3_q11.txt
  $(ORBITER_PATH)orbiter.out--v4-
  -define-C--combinatorial_objects-
  -file_of_points-elliptic_curve_b1_c3_q11.txt-
  -end-
  -define-F--finite_field-q11-end-
  -define-P--projective_space-2F-end-
  -with-C--do-
  -combinatorial_object_activity-
  -canonical_form_PG-P-
  -classification_prefix-EC-
  -label-EC-
  -saveago-
  -max_TDO_depth-4-
  -end-
  -report-
  -prefix-EC-
  -export_flag_orbits-
  -show_TDO-
  -show_TDA-
  -dont_show_incidence_matrices-
  -export_group-
  -end
```

Orbiter shows that the curve has a collineation stabilizer of order 6, generated by
The following example computes the canonical form and the automorphism group of the Hirschfeld surface in PG(3, 4). Using indexing of points in PG(3, 4), we encode the surface as a set of points using Orbiter ranks. We use a makefile variable to provide these ranks as input for the canonical form procedure.

```
HIRSCHFELD_SURFACE_Q4_SET_OF_POINTS="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,23,\n26,27,30,31,34,35,38,39,42,47,48,51,52,53,54,59,60,61,62,67,68,69,70,75,76,\n79,80,81,82"
```

```
Hirschfeld_q4.c:Hirschfeld_surface_q4.txt
pdflatex:Hirschfeld_surface_q4_classification.tex
open:Hirschfeld_surface_q4_classification.pdf
```

In the next example, we look at the two hyperovals in PG(2, 16).

```
hyperoval_16.c:
  $(ORBITER_PATH)orbiter.out:-v.2:\n  -define C:combinatorial_objects:\n  -set_of_points:$(HYPEROVAL_16_16320):\n```
\begin{verbatim}
▷▷▷ -set_of_points\$(HYPEROVAL\_16\_144)\$
▷▷▷ -end\$
▷▷▷ -define:F:-finite_field\_q16\_end\$
▷▷▷ -define:P:-projective_space\_2\_F\_end\$
▷▷▷ -with:G:-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
$(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_2\_draw\_matrix$
▷▷▷ -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
$(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_2\_draw\_matrix$
▷▷▷ -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
\end{verbatim}

317
12.3 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 is there is a bijection between the underlying ground sets which takes one to the other. The incidence matrix is the 0/1 matrix whose rows correspond to the elements of the group set, and whose columns correspond to the chosen subsets. An entry 1 indicates that the corresponding point belongs to the corresponding set.

An incidence geometry is a set system with the following properties: No set appears twice, and no pair of elements in the ground set appear in two different sets. The elements of the ground set are called points. The sets are called lines (or sometimes planes). A flag is an incident point-line pair. An anti-flag is a non-incident point-line pair. Two points are said to be collinear of there is a line in the geometry containing both points.

It is interesting to study the action of the automorphism group on the elements of a geometry. Properties of interest are various levels of transitivity on the elements of the geometry. For instance, a geometry is line-transitive if the automorphism group is transitive on lines. Likewise, it is flag transitive if the automorphism group is transitive on flags. The collinearity graph of a geometry is the graph whose vertices correspond to the points, with two vertices adjacent of the associated points are collinear. The girth of the incidence geometry is the girth of the associated collineation graph. A geometry is triangle free if its girth is at least 4.

A configuration \( v_r b_k \) is an incidence geometry with a ground set of size \( v \) and with \( b \) lines such that each line has size \( k \) and each point is contained in exactly \( r \) lines. In the special case where \( b = v \) and \( k = r \), the name symmetric configuration \( v_r \) is used (the term symmetric is somewhat misleading because the incidence matrix of a symmetric configuration need not be symmetric). Orbiter can be used to classify incidence geometries. One of the important steps in this process is computing a canonical form of the incidence geometry.

We will also be producing drawings of the incidence matrices of geometries. In these drawings, flags are indicated as heavy squares while anti-flags are drawn as small squares. The coloring will indicate the orbits of the automorphism group on flags and anti-flags. Objects with the same color belong to the same orbit. For a flag-transitive geometry, there is only one color for the incidences.

The following command computes the canonical form and a report of the projective plane \( \text{PG}(2, 2) \), which is a configuration \( 7_3 \).

```
geo_7_3.c:
▷ $(ORBITER_PATH)orbiter.out:-v.10:\
▷ ▷ -draw.incidence_structure_description:\
▷ ▷ ▷ -width.60.-with.10.6.-end:\
```
A bitmap drawing is produced, as shown in Figure 12.1. The command also produces the following report of the geometry:

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

Ago : 168
Isomorphism type 0 / 1

Isomorphism type 0 / 1 is original object 0 and appears 1 times:
This isomorphism type appears 1 times, namely for the following 1 input objects:
{0}

incidence structure:
( 0, 1, 2, 7, 10, 11, 14, 19, 20, 22, 24, 26, 29, 32, 34, 37, 38, 41, 44, 46, 47 )

Column sets of the encoded object:
{ 0, 1, 2 }
{ 0, 3, 4 }
{ 0, 5, 6 }
{ 1, 3, 5 }
{ 1, 4, 6 }
{ 2, 3, 6 }
{ 2, 4, 5 }

Row sets of the encoded object:
{ 0, 1, 2 } = 0
{ 0, 3, 4 } = 9
{ 0, 5, 6 } = 14
{ 1, 3, 5 } = 20
{ 1, 4, 6 } = 23
{ 2, 3, 6 } = 27
{ 2, 4, 5 } = 28
Generators for the automorphism group:
The stabilizer of order 168 is generated by:
\[ g_1 = (3, 5)(4, 6)(8, 9)(12, 13) \] of order 2 and with 6 fixed points.
\[ g_2 = (3, 4)(5, 6)(10, 11)(12, 13) \] of order 2 and with 6 fixed points.
\[ g_3 = (1, 2)(5, 6)(10, 12)(11, 13) \] of order 2 and with 6 fixed points.
\[ g_4 = (1, 3)(2, 4)(7, 8)(11, 12) \] of order 2 and with 6 fixed points.
\[ g_5 = (0, 1)(4, 5)(8, 10)(9, 11) \] of order 2 and with 6 fixed points.

Canonical labeling:
canonical row = 6
canonical orbit number = 0
Flags : ( 0, 1, 2, 7, 10, 11, 14, 19, 20, 22, 24, 26, 30, 31, 34, 36, 39, 41, 44, 46, 47 )

Flag orbits:
orbit length : number of orbits of that length:
\[ 21 \quad 1 \]

Anti-Flag orbits:
orbit length : number of orbits of that length:
\[ 28 \quad 1 \]

The following command computes the canonical form and a report of the affine plane \( AG(2, 3) \), which is a configuration \( 9_4 12_3 \).

\[ \text{AG_2_3.c:AG_2_3.inc} \]
\[ \quad \text{\$ORBITER_PATH} \text{orbiter.out} -v -2 \]
\[ \quad \quad \text{-define C:-combinatorial_objects} \]
\[ \quad \quad \text{-file_of_incidence_geometries} \]
\[ \quad \quad \text{-AG_2_3.inc-9-12-36} \]
\[ \quad \quad \text{-end} \]

321
Figure 12.2: The affine plane $AG(2, 3)$ is a configuration $9_412_3$.

A bitmap drawing is produced, shown in Figure 12.2. 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:
\[ \begin{array}{c|c}
\rightarrow & 12_1 \\
9_0 & 4 \\
\downarrow & 12_1 \\
9_0 & 3 \\
\end{array} \]
Decomposition by automorphism group:

Canonical labeling:
- canonical row = 6
- canonical orbit number = 0
- Flags: ( 0, 1, 2, 3, 12, 16, 17, 18, 24, 31, 32, 33, 37, 40, 43, 46, 49, 53, 56, 59, 62, 64, 69, 71, 75, 78, 79, 83, 87, 89, 93, 94, 98, 102, 104, 106 )

Flag orbits:
- orbit length: number of orbits of that length:
  - 36: 1

Anti-Flag orbits:
- orbit length: number of orbits of that length:
  - 72: 1

It is possible to perform isomorph classification for configurations based on incidence files. Suppose we want to check that the configurations in $10_3$ are in fact all nonisomorphic. We apply the canonical form algorithm given by Nauty. This produces a transversal of the isomorphism types of incidence geometries from the given list of input objects. The objects are specified by means of the `combinatorial_objects` command. The classification algorithm can print a report which lists the transversal and all elements in it in latex form.

```bash
geo_10_3.c:
g $(ORBITER_PATH)orbiter.out -v.2 
g \> define C -combinatorial_objects 
g \> \> -file_of_incidence_geometries.10_3.inc.10.10.30 
g \> \> -end 
g \> \> -with C -do 
g \> \> -combinatorial_object_activity 
g \> \> \> -canonical_form 
g \> \> \> -classification_prefix.10_3 
g \> \> \> -save_ago 
```

324
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, 3², 4², 6, 10, 12, 24, 120

### Isomorphism type 0 / 10

Isomorphism type 0 / 10 is original object 9 and appears 1 times:
This isomorphism type appears 1 times, namely for the following 1 input objects:
{9}
incidence structure:
( 0, 1, 2, 10, 13, 14, 20, 25, 26, 31, 33, 35, 41, 44, 47, 52, 53, 58, 62, 66, 69, 74, 78, 79, 85, 87, 89, 96, 97, 98 )
Generators for the automorphism group:
The stabilizer of order 3 is generated by:
\( g_1 = (0, 1, 3)(2, 5, 4)(6, 7, 8)(10, 13, 11)(12, 14, 15)(16, 18, 17) \) of order 3 and with
2 fixed points.

Decomposition by automorphism group:

10131112141516181719

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,

16181719151214101311

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:

Canonical labeling:
canonical row = 0
canonical orbit number = 0
Flags : 0,1,2,15,18,19,24,26,29,33,37,39,40,43,44,50,55,56,61,67,68,72,75,77,82,84,88,91,93,96,

The following command computes the canonical form for the three triangle free configurations
$24_3$ found by Abdullah Alazemi. These configurations have 24 points, 24 lines, each line consists of 3 points and each point is on 3 lines.

TFC\_24\_3\_c:
\begin{verbatim}
\textgreater \textgreater \textgreater \textgreater echo\$\text{(FILE\_24\_3\_TFC\_INC)\textgreater 24\_3\_TFC\_inc}
\textgreater $(\text{ORBITER\_PATH})\text{or} \text{bit.\_out} -v \_6 \textless \textless \textless \textless
define \text{C\_combinatorial\_objects}\textless \textless \textless \textless
\textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgreater \textgre...
Figure 12.3: A flag transitive $24_3$ configuration
12.4 Canonical Forms of Objects from Design Theory

In Secton 10.5, some large sets of $AG(2, 3)$ were constructed. The final isomorphism classification is performed using the Nauty interface. A list of combinatorial objects is created, and the `-canonical_form` command is applied as activity. This will produce a list of pairwise non-isomorphic designs. The size of this list is the number of isomorphism types of large sets of $AG(2, 3)$.

```
LS_AG_2_3_solutions_classify:
  $(ORBITER_PATH)orbiter.out-v.2\$
  -draw_incidence_structure_description\$
  -width.30.-with.10.3.-end\$
  -define.C.-combinatorial_objects\$
  -file_of_designs\$
  solutions.csv.9.84.3.12\$
  -end\$
  -with.C.-do.\$
  -combinatorial_object_activity\$
  -canonical_form\$
  -save Ago\$
  -classification_prefix.large_sets_of_AG_2.3\$
  -end\$
  -report.\"classification\".\$
  -end
  pdflatex.large_sets_of_AG_2.3_classification.tex
  open.large_sets_of_AG_2.3_classification.pdf
```

It turns out that there are exactly two isomorphism types, with automorphism groups of order 54 and 42, respectively.
12.5 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:

```latex
\begin{verbatim}
code_3_2_aut:
\$\text{(ORBITER\_PATH)}\text{orbiter.out\_v.20}\$
\$\text{\_define\_F\_finite\_field\_q.2\_end}\$
\$\text{\_define\_genma\_vector\_field\_F\_format.2}\$
\$\text{\_dense}\$(\text{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{\_classification\_prefix}\text{\_3_2}\$
\$\text{\_end}\$
\end{verbatim}
```

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>
```

331
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:
\( g_1 = (1, 2) \) of order 2 and with 4 fixed points.
\( g_2 = (0, 1) \) of order 2 and with 4 fixed points.

Generators for the automorphism group as matrix group:
The stabilizer of order 6 is generated by:
\[ g_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \end{bmatrix} \] of order 2 and with 1 fixed points.
\[ g_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 01 \\ 10 \end{bmatrix} \] of order 2 and with 1 fixed points.

Decomposition by combinatorial refinement:
Decomposition by automorphism group:

\[
\begin{array}{c|c}
\rightarrow & 2_1 \\
4_0 & 2 \\
\downarrow & 2_1 \\
4_0 & 3 \\
\end{array}
\]

Canonical labeling:
- canonical row = 3
- canonical orbit number = 1

Flags: (0, 1, 2, 3, 4, 5, 7)

Flag orbits:
- orbit length : number of orbits of that length:
  - 1 1
  - 3 2

Anti-Flag orbits:
- orbit length : number of orbits of that length:
  - 1 1

We distinguish the 4 codewords of the $[5,2,3]_2$ code amongst the vertices of the Hamming graph $H(5,2)$ and compute the set stabilizer in the automorphism group of the graph.

Hamming_5_2_with_5_2_3_code:
```bash
> $(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 \ 
> -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 \ 
```
The group has order 32. For the graph theoretic commands, see Section 11.1.

The command

CODE_RM_3_1_GENMA="\n1111111\n01010101\n00110011\n00001111"

RM_3_1_group:
$\$(ORBITER\_PATH)orbiter.out\-v\-2\$
$\$-define\::finite\_field\:-q\:2\:-end\$
$\$-define\::genma\:-vector\:-field\:F\:-format\:4\$
$\$-compact\::(CODE\_RM\_3\_1\_GENMA)\$
$\$-end\$
$\$-define\::P\:-projective\_space\:3\:F\:-end\$
$\$-with\::P\:-do\$
$\$-projective\_space\_activity\$
$\$-canonical\_form\_of\_code\$
$\$-\$ classification\_prefix\:'RM\_3\_1''-end\$
$\$-end\$
pdflatex\::RM\_3\_1\_classification.tex
open\::RM\_3\_1\_classification.pdf

computes the automorphism group of the Reed-Muller code, of order 1344. It is the affine group $AGL(3, 2)$. A report is created, showing the automorphism group and the action on $PG(3, 2)$, with the Reed-Muller code distinguished.

The following command creates a drawing of the incidence matrix between points and lines in $PG(3, 2)$, with the Reed-Muller code distinguished:

REED_MULLER_3_1_CODEWORDS="0,255,170,85,204,51,102,\n153,240,15,90,165,60,195,150,105"

RM_3_1_group_and_diagram:
$\$(ORBITER\_PATH)orbiter.out\-v\-2\$
$\$-define\::finite\_field\:-q\:2\:-end\$
$\$-define\::genma\:-vector\:-field\:F\:-format\:4\$
$\$-compact\::(CODE\_RM\_3\_1\_GENMA)\$
$\$
The drawing in Figure 12.4 is created.

The command

\texttt{RS\_6\_4\_7\_group:}

\begin{verbatim}
  $(ORBITER\_PATH)orbiter.out.-v.20/ 
  -define.F.-finite_field.-q.7.-end/
  -define.genma.-vector.-field.F.-format.4/ 
  -compact:$(CODE\_RS\_6\_4\_7)/
  -end/
\end{verbatim}

335
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

\begin{verbatim}
GV_n15_k6_d5_group:
  \$ (ORBITER_PATH) orbiter.out \-v.20 \-
  \-define F \-finite_field \-q.2 \-end \-
  \-define genma \-vector \-field F \-format 6 \-
  \-define \-compact $(CODE GV_N15_K6) \-
  \-end \-
  \-define P \-projective_space.5 F \-end \-
  \-with P \-do \-
  \-projective_space_activity \-
  \-canonical_form_of_code \-
  \-classif\ation_prefix "GV_n15_k6_d5" \-save ago \-l\abel "GV_n15_k6_d5" \-
  \-classification_prefix "GV_n15_k6_d5" \-end \-
  \-end pdflatex GV_n15_k6_d5_classification.tex
  open GV_n15_k6_d5_classification.pdf
\end{verbatim}

computes the automorphism group of the Gilbert-Varshamov code from Section 9.8. It has order 12.
12.6 Canonical Forms of General Codes

The command

$\text{HAMMING\_CODE\_CODEWORDS}=\"0,\cdot67,\cdot37,\cdot102,\cdot22,\cdot85,\cdot51,\cdot112,\cdot15,\cdot76,\cdot42,\cdot105,\cdot25,\cdot90,\cdot60,\cdot127\"$

Hamming\_graph\_7\_with\_Hamming\_code:
$\triangledown$ $(\text{ORBITER\_PATH})\text{orbiter.out}:-v.2:\$
$\triangledown$ $\triangledown$ $\text{-define} G - \text{-graph} - \text{Hamming}:7:2:$
$\triangledown$ $\triangledown$ $\triangledown$ $\text{-subset} \"\text{Hamming\_code}".\"\text{\_with}\text{\_Hamming\_code}\".$
$\triangledown$ $\triangledown$ $\text{-$(\text{HAMMING\_CODE\_CODEWORDS})}\text{-end}\$.
$\triangledown$ $\triangledown$ $\triangledown$ $\text{-with} G - \text{-do}\$
$\triangledown$ $\triangledown$ $\text{-graph\_theoretic\_activity} - \text{-export} \text{csv} - \text{-end}\$
$\triangledown$ $\triangledown$ $\text{-with} G - \text{-do}\$
$\triangledown$ $\triangledown$ $\text{-graph\_theoretic\_activity} - \text{-export} \text{graphviz} - \text{-end}\$
$\triangledown$ $\triangledown$ $\text{-with} G - \text{-do}\$
$\triangledown$ $\triangledown$ $\text{-graph\_theoretic\_activity} - \text{-save} - \text{-end}\$
$\triangledown$ $\triangledown$ $\text{-with} G - \text{-do}\$
$\triangledown$ $\triangledown$ $\text{-graph\_theoretic\_activity} - \text{-automorphism\_group} - \text{-end}\$
pdflatex Hamming\_7\_2 Hamming\_code\_report.tex
$\triangledown$ 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 $2688 = 16 \cdot 168$. 
12.7 Canonical Forms of Graphs

Orbiter can compute isomorphism and automorphism between graphs. orbiter relies on the canonical labelings of graphs computed by Nauty [47], 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 \n\$ -with:Gamma -do \n\$ -graph_theoretic_activity -automorphism_group -end \n```

The output is two files: The first one, `Cycle_13_group.makefile` is a makefile containing an Orbiter command to create the automorphism group:

```
Cycle_13:
\$ (ORBITER_PATH) orbiter.out -v.2 \n\$ -define:gens -vector:file:Cycle_13_gens.csv -end \n\$ -define:G -permutation_group \n\$ -bsgs:Cycle_13:"Cycle\_13".13.26."0,5".2.gens -end \n```

The second file is `Cycle_13_gens.csv`, which contains the permutation representation of the group, and which is needed for the makefile:

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

The next command computes the automorphism group of the chain graph with respect to the partition (2,3,2).

```
Chain_232.aut:
\$ (ORBITER_PATH) orbiter.out -v.2 \n\$ -define:P1 -vector:dense:2,3,2 -end \n\$ -define:P2 -vector:dense:2,3,2 -end \n\$ -define:Gamma -graph \n\$ \$ -chain_graph:P1.P2 \n\$ -end \n\$ -with:Gamma -do \n\$ -graph_theoretic_activity -automorphism_group \n\$ -end \npdflatex:chain_graph_report.tex \nopen:chain_graph_report.pdf
```

339
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:\
  -../JUNTTILA/KASKI/benchmarks/pp/pp16-1:\
  -end:\
  -with:Gamma:-do:\
  -graph_theoretic_activity:-save:-end:\
  -with:Gamma:-do:\
  -graph_theoretic_activity:-automorphism_group:-end:\
```

340
The command shows a group of order 34217164800. As a measurement of the complexity, the number of backtrack nodes in Nauty is recorded:

\[
\begin{align*}
nb_{\text{backtrack}1} &= 6 \\
nb_{\text{backtrack}2} &= 134 \\
nb_{\text{backtrack}3} &= 134 \\
nb_{\text{backtrack}4} &= 1
\end{align*}
\]

Here, \(nb_{\text{backtrack}1}\) is the number of calls to \texttt{firstpathnode}, \(nb_{\text{backtrack}2}\) is the number of calls to \texttt{othernode}, \(nb_{\text{backtrack}3}\) is the number of calls to \texttt{processnode}, \(nb_{\text{backtrack}4}\) 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:

\[
\text{JK\_graph\_pp16\_2:} \\
\text{$(\text{ORBITER\_PATH})$orbiter.out-\text{-v}\cdot2\backslash} \\
\text{define-Gamma-\text{-graph}-load_dimacs\backslash} \\
\text{../JUNTTILA\_KASKI/benchmarks/pp/pp16\_2\backslash} \\
\text{-end\backslash} \\
\text{-with-Gamma-do\backslash} \\
\text{-graph\_theoretic\_activity-save-end\backslash} \\
\text{-with-Gamma-do\backslash} \\
\text{-graph\_theoretic\_activity-automorphism\_group-end\backslash}
\]

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.

\[
\text{JK\_graph\_sts\_13:} \\
\text{$(\text{ORBITER\_PATH})$orbiter.out-\text{-v}\cdot2\backslash} \\
\text{-define-Gamma-\text{-graph}-load_dimacs\backslash} \\
\text{../JUNTTILA\_KASKI/benchmarks/srg/sts\_13\backslash} \\
\text{-end\backslash} \\
\text{-with-Gamma-do\backslash} \\
\text{-graph\_theoretic\_activity-save-end\backslash} \\
\text{-with-Gamma-do\backslash} \\
\text{-graph\_theoretic\_activity-automorphism\_group-end} \\
\text{make-ORBITER\_PATH=$(\text{ORBITER\_PATH})-f\cdot\text{sts}\_13\_group\_makefile\_\text{-sts}\_13}$
\]
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\$
  \$-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\$
  \$-load\-csv\-no\-border\$
  \$-with\-G\-do\$
  \$-graph\-theoretic\-activity\$
  \$-poset\-classification\-control\$
  \$-W\$
  \$-problem\-label\-HJ\-orbits\$
  \$-depth\-2\$
  \$-end\$
  \$-orbits\-on\-subsets\-2\$
```

342
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\n> -define.gens.vector-file\n>   halljanko315.gens.end\n> -define.G-permutation_group\n>   -bsgs.halljanko315."File\halljanko315".\n>   315.1209600."0,1,2".6.gens.\n> -end\n> -define.Gamma-graph.\n>   -orbital_graph.G.3.\n> -end\n> -with.Gamma-do.\n>   -graph.theoretic_activity.\n>   -properties.\n>   -end.\n> -with.Gamma-do.\n>   -graph.theoretic_activity.\n>   -save.\n> -end
```

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.\n> $(ORBITER_PATH)orbiter.out-v.3.\n> -define.Inc-vector-file0_5_2_incidence_matrix.csv.end.\n> -define.Gamma-graph-collinearity_graph.Inc.end.\n> -with.Gamma-do.\n> -graph.theoretic_activity.\n>   -automorphism_group.\n> -end
```

The group is PGO(5, 2) of order 720. The command creates the group as a permutation group on the 15 vertices of the graph. The group is no longer treated as a matrix group.
Chapter 13

Interfaces

13.1 Graphical Output

Orbiter can produce graphical output in a variety of formats:

1. TikZ / Latex [60],
2. Metapost [30],
3. Bitmap files (.bmp) [63],
4. Povray, see Section 13.2.

Bitmaps can be created using the \texttt{-draw\_matrix} command. The input is an integer-valued matrix in csv format. The matrix entries are translated into colors. The possible commands after \texttt{-draw\_matrix} are shown in Table 13.1. 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:

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
Command & Arguments & Description \\
\hline
\texttt{-input\_csv\_file} & csv-file & Specify the input csv file \\
\hline
\texttt{-partition} & \( w \ R \ C \) & Specify a partitioning \( R \) of rows and \( C \) of columns. Use separating lines of with \( w \). \\
\hline
\texttt{-box\_width} & \( w \) & Use \( w \) pixels per matrix entry. \\
\hline
\texttt{-bit\_depth} & \( d \) & Use color bit depth of \( d \) bits (\( d = 8 \) or \( d = 24 \)). \\
\hline
\texttt{-invert\_colors} & & Use an inverted color scheme. \\
\hline
\end{tabular}
\caption{Commands to Create Bitmap Graphics}
\end{table}
F₇.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

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 13.1. The -partition command is used to define an outline of width 3 pixes. The all-in-one partition 7 is used as both row-partition and column-partition.

The planes PG(2, q) admit a cyclic automorphism group known as the Singer cycle. The command

PG_2.4_cyclic_incma:
  $\$(\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\$

346
Figure 13.1: Addition table of $\mathbb{F}_7$

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

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

produces a cyclically ordered incidence matrix of the plane $\text{PG}(2,4)$, shown in Figure 13.2. The Singer cycle is the projectivity defined by the companion matrix of an irreducible polynomial. We may pick the irreducible polynomial $X^2 + X + \omega$ over $\mathbb{F}_4$. The associated Singer cycle is the projectivity defined by the companion matrix

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

The poset classification algorithm from Sections 6.3 and 6.4 computes partially ordered sets. The posets are created using the -draw_poset option in the poset classification control command package, see Table 6.2. The posets are stored in a file with extension .layered_graph. These files can be drawn using the -draw_layered_graph command. The commands in Table 13.2 and Table 13.3 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 13.2: Drawing Options for Layered Graph Files (Part 1)
Figure 13.2: 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 13.3: Drawing Options for Layered Graph Files (Part 2)
Figure 13.3: The first basic orbit of PGL(4, 2) as a subgroup of PGO⁺(6, 2)

```
PGL_4_2_Wedge_4_0_detached_graphical_output:
▷ $(ORBITER_PATH)orbiter.out:-v.12:\
▷ ▷ -define G:linear_group:-PGL_4_2:\
▷ ▷ -wedge_detached:-end:\
▷ ▷ -with G:-do:\
▷ ▷ -group_theoretic_activity:\
▷ ▷ ▷ -report:\
▷ ▷ -end
▷ pdflatex PGL_4_2_Wedge_4_0_detached_report.tex:
▷ open PGL_4_2_Wedge_4_0_detached_report.pdf
```

produces a report about this group action. Figure 13.3 shows the first basic orbit in the stabilizer chain of the group in that action.

The command

```
schreier_tree_graphical_output:
▷ $(ORBITER_PATH)orbiter.out:-v.4:\
▷ ▷ -draw_options:\
▷ ▷ ▷ -yout:500000:\
▷ ▷ ▷ -radius:15.-nodes_empty:\
▷ ▷ ▷ -line_width:0.5.-y_stretch:0.25:\
▷ ▷ -end:\
▷ ▷ -define G:linear_group:-PGL_4_2:-end:\
▷ ▷ -with G:-do:\
▷ ▷ -group_theoretic_activity:\
▷ ▷ ▷ -orbits_on_polynomials:3:\
```

350
Figure 13.4: A Schreier tree in the action on polynomials

\begin{Verbatim}
\texttt{-orbits_on_polynomials_draw_tree.6.tex}
\end{Verbatim}

\begin{Verbatim}
\texttt{-end.}
\end{Verbatim}

\begin{Verbatim}
\texttt{pdflatex-poly_orbits_d3_n3_q2.tex}
\end{Verbatim}

\begin{Verbatim}
\texttt{open-poly_orbits_d3_n3_q2.pdf}
\end{Verbatim}

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 13.4. This particular orbit has length 420, so there are 420 nodes in the tree.
13.2 The Povray Interface

Orbiter can be used to create raytracing 3D-graphics. Orbiter serves as a front end for the raytracing software Povray [52]. 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 13.3). 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 13.4-13.5 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 13.6 and 13.7 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 13.8 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 13.7 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:

```bash
$ (ORBITER_PATH) orbiter.out -v 2 -povray \
   -round 0 -nb_frames default 30 \n   -output_mask_cube_4d_03d.pov \n   -video_options -W 1024 -H 768 \n   -global_picture_scale 0.5 \n   -default_angle 75 \n   -clipping_radius 2.7 \n   -end \n   -scene_objects \n   -obj_file cube_centered.obj \n   -edge "0,1" \n   -edge "0,2" \n   -edge "0,4" \n   -edge "1,3" \n   -edge "1,5" \n   -edge "2,3" \n   -edge "2,6" \n   -edge "3,7" \n   -edge "4,5"
```

352
<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 w</td>
<td></td>
<td>Set output dimension to w pixels wide.</td>
</tr>
<tr>
<td>-H h</td>
<td></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 r a_s a_t c_s c_t</td>
<td></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 r F T C</td>
<td></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 r F T C</td>
<td></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 r S C L</td>
<td></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 13.4: 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>$r \ a \ \text{text}$</td>
<td>In round $r$, produce running text text with sustain value $a$.</td>
</tr>
<tr>
<td>-label</td>
<td>$r \ s \ a \ g \ \text{text}$</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>$r \ s \ a \ \text{praeamble} \ g \ \text{text} \ l \ \text{fname}$</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>$r \ d \ \text{fname options}$</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 13.5: Options for Orbiter 3D-graphics (Part 2)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cubic_lex</td>
<td>coeffs</td>
<td>Cubic surface given by 20 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-cubic_orbiter</td>
<td>coeffs</td>
<td>Cubic surface given by 20 coefficients in Orbiter ordering</td>
</tr>
<tr>
<td>-cubic_Goursat</td>
<td>$A\ B\ C$</td>
<td>Cubic surface with tetrahedral symmetry given by 3 Goursat coefficients as $Axyz + B(x^2 + y^2 + z^2) + C = 0$</td>
</tr>
<tr>
<td>-quadric_lex_10</td>
<td>coeffs</td>
<td>Quadric surface given by 10 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-quartic_lex_35</td>
<td>coeffs</td>
<td>Quartic surface given by 35 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-octic_lex_165</td>
<td>coeffs</td>
<td>Octic surface given by 165 coefficients in lexicographic ordering</td>
</tr>
<tr>
<td>-point</td>
<td>coeffs</td>
<td>Point given by three coordinates</td>
</tr>
<tr>
<td>-point_list_from_csv_file</td>
<td>fname</td>
<td>List of points with coordinates given in a csv file</td>
</tr>
<tr>
<td>-line_through_two_points_recentered_from_csv_file</td>
<td>fname</td>
<td>List of lines through two points with point coordinates given in a csv file</td>
</tr>
<tr>
<td>-line_through_two_points_from_csv_file</td>
<td>fname</td>
<td>List of lines through two points with point coordinates given in a csv file</td>
</tr>
<tr>
<td>-point_as_intersection_of_two_lines</td>
<td>$i_1\ i_2$</td>
<td>Create a point from the intersection of two lines $i_1$ and $i_2$</td>
</tr>
<tr>
<td>-edge</td>
<td>$i_1\ i_2$</td>
<td>Create an edge (line segment) between points $i_1$ and $i_2$</td>
</tr>
<tr>
<td>-text</td>
<td>$i_1\ s$</td>
<td>Create a label $s$ located at the point $i$</td>
</tr>
<tr>
<td>-triangular_face_given_by_three_lines</td>
<td>$i_1\ i_2\ i_3$</td>
<td>Create a triangular face 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 13.6: Scene definition commands (part 1)
<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-line_through_two_points</td>
<td>pt-coords</td>
<td>Create a line through two points given by 6 coordinates</td>
</tr>
<tr>
<td>-line_through_two_existing_points</td>
<td>i₁ i₂</td>
<td>Create a line through two points</td>
</tr>
<tr>
<td>-line_through_point_with_direction</td>
<td>x y z uₓ uᵧ uₚ</td>
<td>Create a line through a point (x, y, z) with a given direction (uₓ, uᵧ, uₚ)</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,...,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,...,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 13.7: 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>$i \ r \ \text{prop}$</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>$i \ r \ \text{prop}$</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>$i \ d \ \text{prop}$</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>$i \ \text{prop}$</td>
<td>For each element in plane group $i$, create a plane with given Povray properties.</td>
</tr>
<tr>
<td>-lines</td>
<td>$i \ r \ \text{prop}$</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>$i \ \text{prop}$</td>
<td>For each element in group $i$ of cubics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-quadrics</td>
<td>$i \ \text{prop}$</td>
<td>For each element in group $i$ of quadrics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-quartics</td>
<td>$i \ \text{prop}$</td>
<td>For each element in group $i$ of quartics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-octics</td>
<td>$i \ \text{prop}$</td>
<td>For each element in group $i$ of octics, create a surface with given Povray properties.</td>
</tr>
<tr>
<td>-texts</td>
<td>$i \ d \ s \ \text{prop}$</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 13.8: 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:
The monkey saddle is a cubic surface, given by the equation
\[ z = x^3 - 3xy^2 \]

The next example plots the surface knowns as the monkey saddle. The tangent plane at 
\((0,0,0)\) is drawn as well. An animation is created by rotating the scene around the \(z\)-axis.

```plaintext
MONKEY_SADDLE_CUBIC="1,0,0,0,-3,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0"

monkey:
   $(ORBITER_PATH)orbiter.out
   -v.2
   -povray
   -round.0
   -nb_frames_default.30
   -output_mask_monkey_%d_%03d.pov
   -video_options
   -W.1024
   -H.768
   -global_picture_scale.0.8
   -default_angle.75
   -clipping_radius.0.8
   -camera.0
   -rotate_about_z_axis
   -end
   -scene_objects
   -cubic_lex$(MONKEY_SADDLE_CUBIC)
   -plane_by_dual_coordinates"0,0,1,0"
   -group_of_things."0"
   -group_of_things."0"
   -cubics.0
   -texture{pigment\{Gold\}.finish
   \{ambient.0.4.diffuse.0.5.roughness.0.001
   reflection.0.1.specular.8\}}
   -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
```

359
Here is one of the frames that are created:

The Eckardt surface is given by the equation

\[
\frac{5}{2} xyz - (x^2 + y^2 + z^2) + 1 = 0.
\]

We use the following code to plot the surface and the lines on it. The Schl"afli labeling of the lines is indicated.

Eckardt:

```
mv-makefile_animation-POV
```

```
$(ORBITER_PATH)orbiter.out -v -2 -povray \\
-round:0 -nb_frames_default:30 \\
output_mask:Eckardt_%d%03d.pov \\
video_options:-W 1024 -H 768 \\
global_picture_scale:0.9 \\
default_angle:75 \\
clipping_radius:2.4 \\
camera:0:"1,1,1":-3,1,3":0.12,0.12,0.12":: \\
end::
```

```
-scene_objects::
```

```
-Hilbert_Cohn_Vossen_surface::
```

```
-group_of_things:"0":
```

```
-cubics:0:"texture{pigment{White*0.5 transmit:0.5} finish{ambient:0.4 diffuse:0.5 roughness:0.001 reflection:0.1 specular:0.8}}":
```

```
-group_of_things_as_interval:0.6::
```

```
-group_of_things_as_interval:6.6::
```

```
-group_of_things_as_interval_with_exceptions:12.15::
```

```
"14,19,23":
```

```
-lines:1:0.02:"texture{pigment{color:Red}}":
```

360
Figure 13.5 shows the final product.
Figure 13.5: The Eckardt surface
The Endrass octic [22] is the algebraic surface given by the equation

\[ x_8 := 64 (-w^2 + x^2) (-w^2 + y^2) ((x+y)^2 - 2 w^2) ((x-y)^2 - 2 w^2) - \left(-4 \left(1 + \sqrt{2}\right) (x^2 + y^2)^2 + (8 (2 + \sqrt{2}) z^2 + 2 (2 + 7 \sqrt{2}) w^2) (x^2 + y^2) - 16 z^4 + 8 (1 - 2 \sqrt{2}) z^2 w^2 - (1 + 12 \sqrt{2}) w^4\right) \]

The following Orbiter command creates a povray graphics of the octic, shown in Figure 13.6:

ENDRASS_OCTIC_LEX 165 = "-93.2548,0,0,0,-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,-309.019,\n0,0,1582.59,0,1186.94,0,0,0,0,-2110.12,0,-3165.17,0,-1186.94,0,0,0,0,\n0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-256,0,-468.077,0,-789.019,0,\n-525.726,0,0,0.941125"

derass8:
  ▶ $(ORBITER_PATH)orbit.out -v -2 -povray -\n  ▶ -round 0 -nb_frames_default 30 -\n  ▶ -output_mask endrass_occitc_%d_%03d.pov -\n  ▶ -video_options -W 1024 -H 768 -\n  ▶ -global_picture_scale 0.75 -\n  ▶ -default_angle 75 -\n  ▶ -clipping_radius 3.7 -\n  ▶ -no_bottom_plane -\n  ▶ -camera 0:"1,1,1"."6,6,3"."0,0,0" -\n  ▶ -rotate about 111 -\n  ▶ -end -\n  ▶ -scene_objects -\n  ▶ ▶ -line_through_two_points_recentered_from_csv_file -\n  ▶ ▶ ▶ coordinate_grid.csv -\n  ▶ ▶ ▶ -group_of_things "0" -\n  ▶ ▶ ▶ -group_of_things "1" -\n  ▶ ▶ ▶ -group_of_things "2" -\n  ▶ ▶ ▶ -group_of_things_as_interval 3.39 -\n  ▶ ▶ ▶ -lines 0.0.15 "texture{pigment{color Red}}" finish{diffuse 0.9 phong 1}" -\n  ▶ ▶ ▶ -lines 1.0.15 "texture{pigment{color Green}}" finish{diffuse 0.9 phong 1}" -\n  ▶ ▶ ▶ -lines 2.0.15 "texture{pigment{color Blue}}" finish{diffuse 0.9 phong 1}" -\n  ▶ ▶ ▶ -lines 3.0.05 "texture{pigment{color Black}}" finish{diffuse 0.9 phong 1}" -\n  ▶ ▶ ▶ -octic_lex 165 $(ENDRASS_OCTIC_LEX 165) -\n
363
Figure 13.6: The Endrass Octic

This illustration includes coordinate axes and the $x,y$-plane.
### 13.3 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 13.9. For instance, the command

```bash
monkey_video:
▷ -rm -r FRAMES
▷ -mkdir FRAMES
▷ -rm monkey.mp4
▷ $(ORBITER_PATH)orbiter.out:
▷ ▷ -prepare_frames:
▷ ▷ ▷ -i 0\-30 monkey_0_%03d.png:
▷ ▷ ▷ -output_starts_at 0:
▷ ▷ ▷ -o FRAMES/frame%04d.png:
▷ ▷ -end
▷ ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png:
▷ ▷ -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 [35].

<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 13.9: Prepare frames commands
13.4 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

\[ x = r \sin(\alpha t + c), \quad y = r \sin(\beta t), \quad \alpha, \beta, c, r \in \mathbb{R}. \]

The terms
\[ r \sin(\alpha t + c), \quad r \sin(\beta t) \]
are the expressions of the two coordinate functions. RPN means that the operator is listed after the operands. A stack data structure is used to hold temporary values. Operators are pushed to the top of the stack using the push commands. A binary operator pops the two elements from the stack, performs the operation, and pushes the resulting value back onto the stack. For a unary operator, only one element is popped and replaced by the result. Here are some examples of expressions rewritten in RPN:

\[ \sin(x) \mapsto \text{push } x \sin, \]
\[ a + b \mapsto \text{push } a \text{ push } b \text{ add}, \]
\[ a \cdot b \mapsto \text{push } a \text{ push } b \text{ mult}. \]

The coordinate functions are enclosed between `-code` and `-code_end` commands. Each coordinate function is described in RPN and terminated using a `return` keyword. By the time the `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 `-const` and `-const_end` keywords. Likewise, variables are declared between the `-var` and `-var_end` keywords. Picking \( a = 3, b = 2, c = \pi/2 \) and \( r = 7 \), the function is computed using

\begin{verbatim}
liassajous:
  ▶ $(ORBITER_PATH)orbiter.out-\-v.2\-
  ▶ ▶ -smooth_curve:"liassajous"-0.07\-2000\-15\-0.18.85\-
  ▶ ▶ -const:a\-3\-b\-2\-c\-1.57\-r\-7\-\-const_end\-
  ▶ ▶ -var:t\-\-var_end\-
  ▶ ▶ -code:\n  ▶ ▶ ▶ push\-t\-push\-a\-mult\-push\-c\-add\-sin\-push\-r\-mult\-return\-
  ▶ ▶ ▶ push\-t\-push\-b\-mult\-sin\-push\-r\-mult\-return\-
  ▶ ▶ ▶ -code_end\-
\end{verbatim}

The sequence

```
push t push a mult push c add sin push r mult
```

is \( r \sin(at + c) \) expressed in RPN. The constants are defined in the line

\[-\text{const a 3 b 2 c 1.57 r 7 -const_end}\]
The input variable is defined using the line

```plaintext
-var t -var_end
```

The sequence

```plaintext
-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

```plaintext
lissajous_plot:
  $(ORBITER_PATH)orbiter.out -v.2 -povray -
  -round:0 -nb_frames_default:1 -
  -output_mask=lissajous_%d.%03d.pov -
  -video_options=-W1024 -H768 -
  -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 -
  -point_list_from_csv_file -
  function_lissajous_N2000_points.csv -
  -group_of_things_as_interval:0.6524 -
  -spheres:4.09 "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}}" -
finish{"diffuse:0.9:phong:1}" -
  -scene_objects_end -
```

367
The plot is shown in Figure 13.7.

We can turn it into a 3D plot by using the $t$ value for the $z$ coordinate. The function is computed using the command

\begin{verbatim}
lissajous_3d:
$(ORBITER_PATH)orbiter.out -v.2 -povray -smooth_curve:"lissajous_3d"-0.07:2000:50:0.18.85 -const:a.3:b.2:c.1.57:r.7:-const_end -var:t -var_end -code
push:t:return
-code_end
\end{verbatim}

The code to produce the 3D plot is

\begin{verbatim}
lissajous_3d_plot:
$(ORBITER_PATH)orbiter.out -v.2 -povray -round:0 -nb_frames_default:30
\end{verbatim}
The 3D curve is shown in Figure 13.8.
Figure 13.8: Lissajous Spacecurve
Chapter 14

Miscellaneous

14.1 Miscellaneous

Table 14.1 lists miscellaneous Orbiter commands. The command `-csv_file_select_rows` can be used to select rows from a csv file. The command `-csv_file_select_cols` can be used to select columns from a csv file. The command `-csv_file_select_rows_and_cols` selects rows and columns. Here is an example. We create the multiplication table of the finite field $\mathbb{F}_7$, ordered according to the powers of a primitive element:

$$\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5.$$ 

After that, we pull the rows and columns corresponding to even powers $\alpha^0, \alpha^2, \alpha^4$.

misc_select:
> $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot3\cdot$
> $\cdot$-define\_F\_finite\_field\_q\_7\_end$
> $\cdot$-with\_F\_do\_finite\_field\_activity\_cheat\_sheet\_GF\_end$
> $(\text{ORBITER\_PATH})\text{orbiter.out}\cdot-v\cdot4\cdot-csv\_file\_select\_rows\_and\_cols$
> $\cdot$-GF\_q7\_multiplication\_table\_reordered.csv$
> $\cdot$"0\_2\_4"."0\_2\_4".

The even powers of $\alpha$ create a multiplicative subgroup. Figure 14.1 shows the table of the multiplicative group $\mathbb{F}_7^*$ and the subgroup of squares (compare with Figure 3.4 in Section 3.2). Here is the file GF_q7_multiplication_table_reordered.csv

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

371
<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 14.1: Miscellaneous Orbiter Commands

Figure 14.1: 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:

<table>
<thead>
<tr>
<th>Row, &quot;C0&quot;, &quot;C2&quot;, &quot;C4&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, &quot;1&quot;, &quot;2&quot;, &quot;4&quot;</td>
</tr>
<tr>
<td>1, &quot;2&quot;, &quot;4&quot;, &quot;1&quot;</td>
</tr>
<tr>
<td>2, &quot;4&quot;, &quot;1&quot;, &quot;2&quot;</td>
</tr>
</tbody>
</table>

END
14.2 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 $s = \text{sizeof}(\text{int})$.

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 $s = \text{sizeof}(\text{long int})$.

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 than 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 15

The Makefile

15.1 The Makefile

1 #MY_PATH=../orbiter
2 MY_PATH="/DEV.21/GITHUB/orbiter"
3 #MY_PATH=/scratch/betten/COMPILE/orbiter
4
5 #uncomment exactly one of the following two lines.
6 #uncomment the first if you want to run orbiter through docker.
7 #uncomment the second if you have an installed copy of orbiter and you want to run it directly.
8 ORBITER_PATH=docker run -it --volume ${PWD}:/mnt -w /mnt abetten/orbiter
9 ORBITER_PATH=$(MY_PATH)/src/apps/orbiter/
10
11 #End-of-configuration-part
12
13
14 GINAC_PATH=$(MY_PATH)/src/apps/ginac
15 SANDBOX_PATH=$(MY_PATH)/src/apps/sandbox
16
17 update:
18   ▶ cd $(ORBITER_PATH); make clean;
19   ▶ cd $(MY_PATH); make cleana; git pull; make-
20
21 update_all:
22   ▶ cd $(MY_PATH); make clean; git pull; make-
sandbox:
> $(SANDBOX_PATH)/sandbox.out

# Makefile Variables

MAGMA_PATH=/usr/local/magma

#Co3 from Conway et al., 1985 (ATLAS)
# order = 495766656000
#Co3 from the paper by Suleiman and Wilson 1997

CONWAY_GEN1="\n1101110001000001010000\n11110111110100001011\n0000000000100010101\n11110011011001001110\n0101010000001001110\n0000100000001000101\n0010000000001000101\n0001000110000001111\n1110100100110100010\n0000000000110010101\n0000000001000100010\n0000000000000100010\n0000000001000100010\n0000000000000100011\n0000000000100100010\n0000000000000100001\n1110100111100100010\n0000000000101110101\n0000000001000100010\n0000000000000100010\n0000000000000100011\n0000000000000100010"

CONWAY_GEN2="\n01010000101110101111\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001\n0101000011110100001"
HIRSCHFELD_SURFACE_EQUATION="0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0"
ENDRASS_SPARSE="
6,0,4,4,2,7,5,9,6,20,6,23,1,25,3,30,1,32,3,34,4,56,6,59,1,61,6,66,\
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"
EDGE_CURVE_Q23_AS_POINTS="4,·25,·26,·47,·48,·71,·92,·95,·114,·119,·136,·143,·158,\
·167,·180,·191,·202,·215,·224,·239,·246,·263,·268,·287,·290,·311,·312,·334,·335,\
·356,·359,·378,·383,·400,·407,·422,·431,·444,·455,·466,·479,·488,·503,·510,·527,·530,·532,·551"
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="\"
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,\n0,0,0,0,1,0,0,0,\n0,0,0,0,0,1,0,0,\n0,0,0,0,0,0,1,0,\n0,0,0,0,0,0,0,1,\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"

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,\n9,7,2,15,17,5,11,-3,3,6,21,4,24,16,25,8,6,24,21,12,7,\n24,15,2,13,11,14,24"

HIRSCHFELD_SURFACE_Q4_SET_OF_POINTS="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,23,\n26,27,30,31,34,35,38,39,42,47,48,51,52,53,54,59,60,61,62,67,68,69,70,75,76,\n79,80,81,82"

HYPEROVAL_16_144="0,1,2,3,52,67,89,106,126,141,159,176,184,199,220,\n235,245,262"

HYPEROVAL_16_16320="0,1,2,3,52,70,83,109,127,139,156,174,186,199,217,229,256,264"


378
358-367·379·381·392·398·400·417·428·429·442·443·450·466·471·479·492·497·502·517·
519·521·542·548·551·571·574·575·
\n0·1·2·24·27·28·48·53·54·73·79·80·97·105·106·122·131·132·146·157·158·171·175·183
\n195·203·208·220·225·233·244·258·259·269·272·281·293·301·308·318·324·327·342·354·
357·367·373·378·392·400·403·417·419·430·442·446·447·466·472·479·492·500·503·518·
525·526·545·549·551·571·572·574·
\n0·1·2·24·27·28·48·53·54·73·79·80·97·105·106·122·131·132·146·157·158·171·175·179
\n195·201·207·220·226·232·244·257·258·269·274·277·293·300·307·318·323·329·342·352·
356·367·374·381·392·397·406·416·423·431·441·450·454·468·476·477·494·499·503·519·
521·525·544·547·550·570·572·575·
\n0·1·3" ELEMENTARY_SYMERIC_3_1="x0+x1+x2" ELEMENTARY_SYMERIC_3_2="x0*x1+x0*x2+x1*x2"
ELEMENTARY_SYMERIC_3_3="x0*x1*x2"
ELEMENTARY_SYMERIC_4_1="x0+x1+x2+x3" ELEMENTARY_SYMERIC_4_2="x0*x1+x0*x2+x0*x3+x1*x2+x1*x3+x2*x3"
ELEMENTARY_SYMERIC_4_3="x0*x1*x2+x0*x1*x3+x0*x2*x3+x1*x2*x3"
ELEMENTARY_SYMERIC_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,
53,58,59,60,61,62,63,64,67,77,80,90,93,103,107,115,118,122,125,127,142,147,155,15,
7,162,165,170,172,204,208,219,229,240,244,246,251,253,259,277,278,281,286,298,300,
302,303,310,312,316,340,343,351,354,358,365,369,372,373,379,384,386,388,393,399"
SURFACE_F7_15LINES_MCKEAN_POINTS="0,1,2,3,4,5,6,7,8,9,10,16,17,28,35,36,59,60,61,
62,63,64,65,76,80,91,95,106,107,111,119,121,122,130,138,139,141,146,150,154,155,1
58,170,175,184,186,199,203,204,206,226,231,234,239,240,252,253,254,278,279,282,28,
392,394,399"
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="\n621000\n062100\n006210\n000621"

CODE_RS_10_8_11="\n7,2,1,0,0,0,0,0,\n0,7,2,1,0,0,0,0,\n0,0,7,2,1,0,0,0,\n0,0,0,7,2,1,0,0,\n0,0,0,0,7,2,1,0,\n0,0,0,0,0,7,2,1."

#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="1,2,1,3,1,4,1,7,1,15"
D7="1,2,1,3,1,5,1,9,1,10,1,14,1,15"
D8="1,3,1,4,1,5,1,7,1,10,1,11,1,19"
D9="1,3,1,4,1,7,1,11,1,12,1,15,1,19"
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="1,4,1,7,1,3,1,15,1,12"..
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="1,4,1,7,1,6,1,8,1,11,1,14,1,12,1,16,1,17,1,18,1,19."
D20="1,4,1,5,1,6,1,7,1,8,1,10,1,13,1,14,1,15"
D21="1,5,1,6,1,8,1,10,1,13,1,14,1,15"
D22="1,4,1,5,1,7,1,9,1,10,1,11,1,13,1,14,1,15,1,18,1,19"
\[ F_{\text{ALPHA\_BETA\_GAMMA\_DELA}} = \beta \gamma (\gamma + 1) x_0 x_0 x_2^2 \]

\[ + (\alpha - \delta - \beta \gamma \gamma + \alpha - \beta - \delta - \gamma + 1) x_0 x_1 x_2 \]

\[ - 1 (\alpha - \beta + \delta - \alpha + \delta - 1) x_0 x_1 x_3 \]

\[ + (0 - \alpha - \beta + \gamma - \alpha + \beta - \delta - \gamma) x_0 x_2 x_2 \]

\[ - (\alpha - \beta + \delta - \gamma + 1) x_1 x_2 x_3 \]

\[ - (\alpha - 1) x_1 x_1 x_2 \]

\[ + (\alpha - \beta + \gamma + \alpha - \beta - \delta - \gamma + 1) x_1 x_2 x_2 \]

\[ + (\alpha + \beta + \gamma + \alpha - \beta + \delta + \gamma) x_1 x_2 x_3 \]

\[ + \alpha \beta + (\gamma + 1) x_1 x_3 x_3 \]

\[ F_{\text{abcd\_eqn}} = (a + b + c + d + a + b + c + d + a + d - b - c) (b - d) x_0 x_0 x_2 x_2 \]

\[ + (a + b + c - a + b + d - a + c + d + b + c + d + a + d - b - c) (a + b + c + d + a + d - b - c) x_0 x_1 x_2 \]

\[ + (a + a + c + a + a + c + a + c + c + c + b + c + c + a + d - b - c) (b - d) x_0 x_1 x_3 x_3 \]

\[ - (a + a + c + d - a + b + c + c + a + d - b - c) (b - d) x_0 x_2 x_2 x_2 \]

\[ - (a + a + c + d - a + b + c + c + a + d - b - c) (b - d) x_0 x_2 x_3 x_3 \]

\[ - (a - c) (a + b + c + a + b + d + b + c + d + a + d - b - c) x_1 x_1 x_2 \]
\[ -(a-b-c)(a*b*c-a*b*d+a*c*d-a*b*c)*X1*X1*X3 - (a*d-b*c)(a*b*c-a*b*d+a*c*d-a*b*c)*X1*X2*X2 - ((1+1)*a*a*b*c*d-a*a*b*d*d-a*a*c*d*d)*X1*X2*X3 - ((1+1)*a*b*b*c*c+a*b*b*c*d+a*b*c*c*d+a*b*c*d*d)*X1*X2*X3 - (b*b*c*c*d-a*a*b*c+a*a*c*d+a*b*b*c+a*b*c*c)*X1*X2*X3 - c*a*(a*d-b*c-a+b+c-d)*(b-d)*X1*X3*X3"
CODE_RS_11_RREF="\n1,0,0,0,0,0,0,0,7,2,\n0,1,0,0,0,0,0,0,8,3,\n0,0,1,0,0,0,0,0,1,2,\n0,0,0,1,0,0,0,0,8,8,\n0,0,0,0,1,0,0,0,10,3,\n0,0,0,0,0,1,0,0,1,4,\n0,0,0,0,0,0,1,0,5,4,\n0,0,0,0,0,0,0,1,5,8"

RS_8_reduced="\n010001100000000000000\n001110010000000000000\n110011001000000000000\n000100011000000000000\n000011100100000000000\n001100110010000000000\n000000100011000000000\n000000011100100000000\n000000011100110000000\n000000011100110010000\n000000000001001100000\n000000000000111001000\n000000000000010001100\n0000000000000011110010\n00000000000001110101\n00000000000001110101"

CODE_21_15_4="\n111000100000000000000-\n110100010000000000000-\n101100010000000000000-\n011100001000000000000-\n110010000100000000000-\n101010000001000000000-\n011010000001000000000-\n100110000000100000000-\n010110000000010000000-\n001110000000001000000-\n111110000000000100000-\n100000000000000000000-\n000000000000000000000-\n000000000000000000000-\n000000000000000000000-\n000000000000000000000-\n000000000000000000000-
```python
# there are 5: [15, 6, 6]

# ago=12
CODE_15_6_6_A="
111111111100000
111110000010000
111001100001000
110101010000100
101011101000001"

# ago=12
CODE_15_6_6_B="
111111111100000
111110000010000
111001100001000
110101010000100
101011101000001"

# ago=720:
CODE_15_6_6_C="
111111111100000
111110000010000
111001100001000
110101010000100
101101001000001"

# ago=96:
CODE_15_6_6_D="
111111111100000
111110000010000
111001100001000
110101010000100
101011001000010"

# ago=360
CODE_15_6_6_E="
111111111100000"
```
# Created in the Combinatorics Section:

**Elementary Symmetric 8.1:**
\[ x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \]

**Elementary Symmetric 8.2:**
\[ x_0 x_1 x_2 + x_0 x_2 x_3 + x_0 x_3 x_4 + x_0 x_4 x_5 + x_0 x_5 x_6 + x_0 x_6 x_7 + x_1 x_2 x_3 + x_1 x_3 x_4 + x_1 x_4 x_5 + x_1 x_5 x_6 + x_1 x_6 x_7 + x_2 x_3 x_4 + x_2 x_4 x_5 + x_2 x_5 x_6 + x_2 x_6 x_7 + x_3 x_4 x_5 + x_3 x_5 x_6 + x_3 x_6 x_7 + x_4 x_5 x_6 + x_4 x_6 x_7 + x_5 x_6 x_7 + x_6 x_7 \]

**Elementary Symmetric 8.3:**
\[ x_0 x_1 x_2 + x_0 x_1 x_3 + x_0 x_1 x_4 + x_0 x_1 x_5 + x_0 x_1 x_6 + x_0 x_1 x_7 + x_0 x_2 x_3 + x_0 x_2 x_4 + x_0 x_2 x_5 + x_0 x_2 x_6 + x_0 x_2 x_7 + x_0 x_3 x_4 + x_0 x_3 x_5 + x_0 x_3 x_6 + x_0 x_3 x_7 + x_0 x_4 x_5 + x_0 x_4 x_6 + x_0 x_4 x_7 + x_0 x_5 x_6 + x_0 x_5 x_7 + x_0 x_6 x_7 + x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_2 x_5 + x_1 x_2 x_6 + x_1 x_2 x_7 + x_1 x_3 x_4 + x_1 x_3 x_5 + x_1 x_3 x_6 + x_1 x_3 x_7 + x_1 x_4 x_5 + x_1 x_4 x_6 + x_1 x_4 x_7 + x_1 x_5 x_6 + x_1 x_5 x_7 + x_1 x_6 x_7 + x_2 x_3 x_4 + x_2 x_3 x_5 + x_2 x_3 x_6 + x_2 x_3 x_7 + x_2 x_4 x_5 + x_2 x_4 x_6 + x_2 x_4 x_7 + x_2 x_5 x_6 + x_2 x_5 x_7 + x_2 x_6 x_7 + x_3 x_4 x_5 + x_3 x_4 x_6 + x_3 x_4 x_7 + x_3 x_5 x_6 + x_3 x_5 x_7 + x_3 x_6 x_7 + x_4 x_5 x_6 + x_4 x_5 x_7 + x_4 x_6 x_7 + x_5 x_6 x_7 + x_5 x_7 + x_6 x_7 \]

**Elementary Symmetric 8.4:**
\[ x_0 x_1 x_2 + x_0 x_1 x_3 + x_0 x_1 x_4 + x_0 x_1 x_5 + x_0 x_1 x_6 + x_0 x_1 x_7 + x_0 x_2 x_3 + x_0 x_2 x_4 + x_0 x_2 x_5 + x_0 x_2 x_6 + x_0 x_2 x_7 + x_0 x_3 x_4 + x_0 x_3 x_5 + x_0 x_3 x_6 + x_0 x_3 x_7 + x_0 x_4 x_5 + x_0 x_4 x_6 + x_0 x_4 x_7 + x_0 x_5 x_6 + x_0 x_5 x_7 + x_0 x_6 x_7 + x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_2 x_5 + x_1 x_2 x_6 + x_1 x_2 x_7 + x_1 x_3 x_4 + x_1 x_3 x_5 + x_1 x_3 x_6 + x_1 x_3 x_7 + x_1 x_4 x_5 + x_1 x_4 x_6 + x_1 x_4 x_7 + x_1 x_5 x_6 + x_1 x_5 x_7 + x_1 x_6 x_7 + x_2 x_3 x_4 + x_2 x_3 x_5 + x_2 x_3 x_6 + x_2 x_3 x_7 + x_2 x_4 x_5 + x_2 x_4 x_6 + x_2 x_4 x_7 + x_2 x_5 x_6 + x_2 x_5 x_7 + x_2 x_6 x_7 + x_3 x_4 x_5 + x_3 x_4 x_6 + x_3 x_4 x_7 + x_3 x_5 x_6 + x_3 x_5 x_7 + x_3 x_6 x_7 + x_4 x_5 x_6 + x_4 x_5 x_7 + x_4 x_6 x_7 + x_5 x_6 x_7 + x_5 x_7 + x_6 x_7 + x_7 \]

**Elementary Symmetric 8.5:**
\[ x_0 x_1 x_2 + x_0 x_1 x_3 + x_0 x_1 x_4 + x_0 x_1 x_5 + x_0 x_1 x_6 + x_0 x_1 x_7 + x_0 x_2 x_3 + x_0 x_2 x_4 + x_0 x_2 x_5 + x_0 x_2 x_6 + x_0 x_2 x_7 + x_0 x_3 x_4 + x_0 x_3 x_5 + x_0 x_3 x_6 + x_0 x_3 x_7 + x_0 x_4 x_5 + x_0 x_4 x_6 + x_0 x_4 x_7 + x_0 x_5 x_6 + x_0 x_5 x_7 + x_0 x_6 x_7 + x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_2 x_5 + x_1 x_2 x_6 + x_1 x_2 x_7 + x_1 x_3 x_4 + x_1 x_3 x_5 + x_1 x_3 x_6 + x_1 x_3 x_7 + x_1 x_4 x_5 + x_1 x_4 x_6 + x_1 x_4 x_7 + x_1 x_5 x_6 + x_1 x_5 x_7 + x_1 x_6 x_7 + x_2 x_3 x_4 + x_2 x_3 x_5 + x_2 x_3 x_6 + x_2 x_3 x_7 + x_2 x_4 x_5 + x_2 x_4 x_6 + x_2 x_4 x_7 + x_2 x_5 x_6 + x_2 x_5 x_7 + x_2 x_6 x_7 + x_3 x_4 x_5 + x_3 x_4 x_6 + x_3 x_4 x_7 + x_3 x_5 x_6 + x_3 x_5 x_7 + x_3 x_6 x_7 + x_4 x_5 x_6 + x_4 x_5 x_7 + x_4 x_6 x_7 + x_5 x_6 x_7 + x_5 x_7 + x_6 x_7 + x_7 + x_7 + x_7 + x_7 + x_7 + x_7 \]
2 \times 3 \times 4 \times 5 \times 6 \times 7

\text{ELEMENTARY SYMMETRIC}_8 = x_0 \times x_1 \times x_2 \times x_3 \times x_4 \times x_5 \times x_6 \times x_7 + x_0 \times x_1 \times x_2 \times x_3 \times x_4 \times x_5 \times x_6 \times x_7 + x_0 \times x_1 \times x_2 \times x_3 \times x_4 \times x_5 \times x_6 \times x_7 + x_0 \times x_1 \times x_2 \times x_3 \times x_4 \times x_5 \times x_6 \times x_7

\text{ELEMENTARY SYMMETRIC}_8 = x_0 \times x_1 \times x_2 \times x_3 \times x_4 \times x_5 \times x_6 \times x_7

E_0 = -1,55,1,45,-1,46,1,57,1,61,-1,62
E_1 = 1,39,-1,40,-1,49,1,51,1,61,-1,62
E_2 = -1,67,1,69,1,59,-1,64,-1,22,1,24
E_3 = -1,38,1,47,1,40,-1,67,-1,51,1,69
E_4 = 1,61,-1,62,-1,67,1,69
E_5 = 1,38,-1,47,-1,39,2,66,-1,67,1,49,-2,68,1,69,1,61,-1,62
E_6 = 1,62,-1,63,-1,64,1,24
E_7 = 1,35,-1,36,-1,38,2,65,-1,29,1,40,-2,66,1,45,1,51,-1,69
E_8 = 1,61,-1,63,-1,59,1,22
E_9 = 1,27,1,47,-2,65,1,48,1,50,-1,40,1,67,2,68,-1,55,-1,51
E_{10} = 1,66,-1,68,-1,59,1,64,1,21,-1,23
E_{11} = -1,53,1,44,1,66,-1,68,1,55,-1,45
E_{12} = 1,61,-1,62,1,66,-1,68
E_{13} = 1,53,-1,44,1,66,-2,67,-1,68,1,46,2,69,-1,57,1,61,-1,62
E_{14} = 1,60,-1,61,1,64,-1,23
E_{15} = 1,27,1,41,1,42,-2,65,1,44,-1,40,2,67,1,68,-1,55,-1,45
E_{16} = 1,60,-1,62,1,59,-1,21
E_{17} = 2,65,-1,53,-1,54,-1,29,1,56,-1,66,1,55,1,45,1,51,-2,69
E_{18} = 1,-1,67,1,69,1,66,-1,68
E_{19} = -1,66,1,67,1,68,-1,69,-2,61,2,62,1,21,-1,22,-1,23,1,24
E_{20} = -1,68,1,61,1,69,-1,62
E_{21} = 1,36,-1,41,-1,37,1,43,2,66,-2,67,-1,68,1,69,1,61,-1,62
E_{22} = -1,66,1,62,1,67,-1,61
E_{23} = 1,54,-1,50,1,66,-1,67,-2,68,2,69,1,52,-1,58,1,61,-1,62
E_{24} = 1,68,-1,69,-1,60,1,63,1,23,-1,24
E_{25} = 1,36,-1,41,-1,40,1,68,1,45,-1,69
E_{26} = 1,66,-1,67,-1,60,1,63,1,21,-1,22
E_{27} = -1,54,1,50,1,66,-1,67,1,55,-1,51
E_{28} = 1,37,-1,40,-1,43,1,45,1,61,-1,62
E_{29} = 1,55,-1,51,1,52,-1,58,-1,61,1,62
E_{30} = 1,21,-1,23,-1,22,1,24
E_{31} = 1,61,-1,21,-1,62,1,22
E_{32} = 1,61,-1,24,-1,62,1,23
E_{33} = 1,61,-1,24,-1,62,1,22
E_{34} = 1,61,-1,21,-1,24,1,23
E_{35} = 1,62,-1,22,-1,23,1,21
E_{36} = 1,61,-1,21,-1,62,1,23
E_{37} = 1,61,-1,23
E_{38} = 1,21,-1,62
E_{39} = 1,62,-1,22,-1,23,1,24
506  E40="1,62,-1,24"
507  E41="1,22,-1,23"
508  E42="1,61,-1,21,-1,24,1,22"
509  E43="1,61,-1,22"
510  E44="1,21,-1,24"
511  H0="1,21"
512  H1="1,21,-1,4"
513  H2="1,22"
514  H3="1,22,-1,4"
515  H4="1,23"
516  H5="1,23,-1,4"
517  H6="1,24"
518  H7="1,24,-1,4"
519  H8="1,21,-1,22"
520  H9="1,21,-1,23"
521  H10="1,22,-1,24"
522  H11="1,23,-1,24"
523  H12="1,61,-1,62"
524  H13="1,61,-1,21,-1,24,-1,62,1,22,1,23"
525  H14="-1,67,-1,68,1,61,1,66,1,69,-1,62"
526
527
528  Orb0="0"
529  Orb1="4"
530  Orb2="16"
531  Orb3="20"
532  Orb4="21"
533  Orb5="24"
534  Orb6="31"
535  Orb7="34"
536  Orb8="35"
537  Orb9="38"
538  Orb10="43"
539  Orb11="46"
540  Orb12="68"
541  Orb13="69"
542  Orb14="76"
543  Orb15="77"
544  Orb16="139"
545  Orb17="140"
546  Orb18="156"
547  Orb19="192"
548  Orb20="264"
549  Orb21="265"
550  Orb22="331"
551  Orb23="337"
# elements of order 2:
# conjugacy class reps:
# elt order, class size, centralizer order.
#2A: 2^4 8960 40320 Baer involution
#2B: 2^4 5355 368640 one block of 10,11
#2C: 2^1 64260 30720 two blocks of 10,11 (problem group)

-> pdflatex-PGGL_4_4_classes.out.tex
-> open-PGGL_4_4_classes_out.pdf
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736

#2·48960·40320·Baer·involution
#2·5355·368640··one·block·of·10,11
#2·64260·30720·two·blocks·of·10,11·(problem·group)

CLASS 2A=-centralizer of element·"1,0,0,0,·0,1,0,0,·0,0,1,0,·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,·0,0,0,1,·0"·-label·"
2B"
CLASS 2C=-centralizer of element·"1,0,0,0,·1,1,0,0,·0,0,1,0,·0,0,1,1,·0"·-label·"
2C"··
#·problem·group

#·3·classes·of·elements·of·order·3
#·4·classes·of·elements·of·order·4

#·Baer·involution:
PGGL 4 4 SUBGROUP 2A=-PGGL·4·4·\
. -subgroup by generators·"2A"·2·1·"1,0,0,0,·0,1,0,0,·0,0,1,0,·0,0,0,1,·1"
PGGL 4 4 SUBGROUP 2A NORMALIZER=-PGGL·4·4·\
. -subgroup by generators·"centralizer 2A"·"40320"·10·\
. "1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1,0,·\
1,0,0,0,0,1,0,0,1,1,0,1,1,1,1,0,0,·\
1,0,0,0,0,1,0,0,1,0,1,0,0,1,0,1,1,·\
1,0,0,0,0,1,0,0,0,0,1,1,1,0,1,0,0,·\
1,0,0,0,1,1,1,1,1,0,1,0,1,1,1,0,0,·\
1,0,0,0,0,1,1,0,0,0,1,1,1,0,1,0,0,·\
0,1,0,0,0,1,0,1,1,1,0,1,0,1,1,1,1

#·the·problem·group,·two·blocks·of·10,11:
PGGL 4 4 SUBGROUP 2C=-PGGL·4·4·\
. -subgroup by generators·"2C"·2·1·\
. "1,0,0,0,·1,1,0,0,·0,0,1,0,·0,0,1,1,·0"

392


PGGL_4_4_SUBGROUP_2C_NORMALIZER=-PGGL_4_4\

- subgroup_by_generators:"centralizer_2C"."30720".9\
1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,1,1,\n1,0,0,0,0,1,0,0,0,0,2,0,0,0,0,2,1,\n1,0,0,0,0,1,0,0,0,0,1,0,2,0,3,1,0,\n1,0,0,0,0,1,0,0,0,0,1,0,1,0,3,1,0,\n1,0,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0,\n1,0,0,0,0,1,0,0,0,0,1,0,1,0,1,1,1,\n1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,1,0,\n1,0,0,0,2,1,0,0,0,0,1,0,1,0,0,1,0,\n1,0,0,0,1,1,2,0,0,0,0,1,0,0,0,1,0,\n1,0,3,0,1,1,1,3,0,0,2,0,0,0,0,2,1,"

PGGL_4_4_SUBGROUP_5A=-PGGL_4_4\

- subgroup_by_generators:"5A"."5-1"\
"0,2,0,0,-1,1,0,0,-0,0,3,0,-0,0,3,0"
PGGL_4_4_SUBGROUP_5A_NORMALIZER=-PGGL_4_4\

- subgroup_by_generators:"normalizer_5A"."3600".6\
"1,0,0,0,0,1,0,0,0,0,2,0,0,0,0,2,0,\
1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,1,0,\n1,0,0,0,0,1,0,0,0,0,0,2,0,0,2,0,0,\n1,0,0,0,2,2,0,0,0,0,1,0,0,0,0,1,1,\n0,1,0,0,3,3,0,0,0,0,2,0,0,0,0,2,0,  

PGGL_4_4_SUBGROUP_5B=-PGGL_4_4\

- subgroup_by_generators:"5B"."5-1"\
"0,2,0,0,1,1,0,0,0,0,0,2,0,0,1,1,0,  
PGGL_4_4_SUBGROUP_5B_NORMALIZER=-PGGL_4_4\

- subgroup_by_generators:"normalizer_5B"."81600".6\
"1,0,0,0,0,1,0,0,0,0,2,0,0,0,0,2,0,\
1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,2,0,0,\n1,0,0,0,2,2,0,0,0,0,1,0,0,0,0,1,1,\n0,1,0,0,3,3,0,0,0,0,2,0,0,0,0,2,0,  
0,0,1,0,0,0,1,2,2,0,0,0,2,3,0,0,1  

PGGL_4_4_SUBGROUP_2Cx2_0=-PGGL_4_4\

- subgroup_by_generators:"2Cx2_0"."4-2"\
"1,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0,  
1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,1,0,  
PGGL_4_4_SUBGROUP_2Cx2_0_NORMALIZER=-PGGL_4_4\

- subgroup_by_generators:"normalizer_2Cx2_0"."768".8\
"1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,  
1,0,0,0,0,1,0,0,0,0,1,0,2,0,0,1,0,  

393
#PGL_4_5_SUBGROUP_3B=-PGL_4_5:\n
#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:\n
PGL_4_5_SUBGROUP_3B_ME_NORMALIZER=-PGL_4_5:\n
#nice-generators, from Michael Epstein:
PGL 4.5 SUBGROUP 31 ME = -PGL 4.5 \ 
- subgroup_by_generators "31" 31 1 \ 
"1,0,0,0,0,3,4,3,0,3,3,4,0,3,2,3" \ 
PGL 4.5 SUBGROUP 31 ME_NORMALIZER = -PGL 4.5 \ 
- subgroup_by_generators "normalizer_31" "372" 4 \ 
"1,0,0,0,0,4,0,0,0,0,0,4,0,0,0,0,4," \ 
"1,0,0,0,0,4,0,0,0,0,0,4,0,0,0,0,4," \ 
"1,0,0,0,0,3,0,0,0,0,3,0,0,0,0,3," \ 
"1,0,0,0,0,4,0,0,0,2,1,0,3,2,4," \ 
"1,0,0,0,0,1,0,0,0,1,0,1,1,3," 

#subgroup-of-order-31-for-the-construction-of-regular-packings-in-PG_3_5: 

PGL 4.5 SUBGROUP 31 = -PGL 4.5 \ 
- subgroup_by_generators "31" 31 1 \ 
"2,0,0,0,0,0,1,0,0,0,1,0,3,0,4" \ 
PGL 4.5 SUBGROUP 31 NORMALIZER = -PGL 4.5 \ 
- subgroup_by_generators "normalizer_31" "372" 4 \ 
"1,0,0,0,0,4,0,0,0,0,0,4," \ 
"1,0,0,0,0,3,0,0,0,0,3,0,0,0,0,3," \ 
"1,0,0,0,0,4,0,0,0,2,1,0,3,2,4," \ 
"1,0,0,0,0,1,0,0,0,1,0,1,1,3," 

#372: 
"1,0,0,0,0,4,0,0,0,4,0,0,0,0,4," \ 
"1,0,0,0,0,3,0,0,0,3,0,0,0,0,3," \ 
"1,0,0,0,0,4,0,0,0,2,1,0,3,2,4," \ 
"1,0,0,0,0,1,0,0,0,1,0,1,1,3," 

#Exterior-square-roots: 

#elt-of-order-3: 

#the-exterior-square-root-of-f-is-X = 

# [1.0.0.0] 

# [0.1.0.0]
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,\n0,1,0,0,0,1,1,\n0,0,0,1,1,1,1"

HAMMING_CODE_GENERATOR="\n1,0,0,0,0,0,1,\n0,1,0,0,0,1,1,\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="\n1,0,0,1,1,0,\n0,1,0,0,1,1,\n0,0,1,1,1,0,1"

CODE_GV_N15_K6="\n1111111110000\n1111000001000\n110011000000\n101010100000\n1010101000001"

CODE_GV_N15_K6_CHECK="\n"
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_PARITY_CHECK="1,3,5,7,9,11,13,  
15,17,19,21,23,25,27,29,31"  
#-nearest_codeword: "8,16,32,24,40,48,56,1,2,4,3,5,6,7,9,18,36,27,45,54,63,33,42,2  
0,11,53,62,31"
RM_6_GENERATOR_1="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,  
22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,  
46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63"
RM_6_GENERATOR_2="1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,  
41,43,45,47,49,51,53,55,57,59,61,63"
RM_6_GENERATOR_3="2,3,6,7,18,19,22,23,10,11,14,15,26,27,30,31,34,35,38,  
39,42,43,46,47,50,51,54,55,58,59,62,63"
RM_6_GENERATOR_4="4,6,12,14,36,38,52,54,5,7,13,15,37,39,53,55,20,22,28,  
30,44,46,60,62,21,23,29,31,45,47,58,59,62,63"
RM_6_GENERATOR_5="8,9,12,13,24,25,28,29,10,11,14,15,26,27,30,31,40,41,  
44,45,56,57,60,61,42,43,46,47,58,59,62,63"
RM_6_GENERATOR_6="16,18,24,26,48,50,56,58,17,19,25,27,49,51,57,59,20,22,  
28,30,52,54,60,62,21,23,29,31,53,55,61,63"
RM_6_GENERATOR_7="32,34,48,50,33,35,49,51,36,38,52,54,37,39,53,55,40,42,  
56,58,41,43,57,59,44,46,60,62,45,47,61,63"
AG_2_3_BLOCKS="0,13,22,27,35,41,47,53,55,59,71,76"
LARGE_SET_AG_2_3_NEIGHBOR_SET="129,130,133,134,136,139,141,142,153,154,156,160,16  
5,166,178,179,183,184,185,190,192,194,197,203,204,206,218,221,222,225,227,231,248  
,251,252,255,256,259,261,262,272,277,279,283,285,287,299,301,303,305,306,309,313,  
315,319,320,323,325,341,342,343,344,345,349,368,371,375,378,381,383,392,393,397,4  
397
# Consider the binary code with generator matrix:

```python
# 1-0-1
# 0-1-1
CODE_N3_K2_Q2_GENMA = "1,0,1,-0,1,1"
```

```python
CODE_N6_K3_Q2_GENMA = "
```

```python
# q=17:
# 3-is-p.e. mod 17.
# so-we-pick f=3.
# then, 2f^2 = 18 = 1
# 4f = 12
# X^4-Y^4-Z^4+2f^2Y^2Z^2+4fX^2YZ:
# (1,-1,-1,0,0,0,0,0,0,0,0,2f^2,4f,0,0)
```
FILE_Q17="orbit,curve,pts_on_curve,bitangents,go\n"n0,"$(EDGE_CURVE_Q17_EQUATION)\"","$(EDGE_CURVE_Q17_AS_POINTS)\"\",-1\n\nFILE_Q23="orbit,curve,pts_on_curve,bitangents,go\n"n0,"$(EDGE_CURVE_Q23_EQUATION)\"","$(EDGE_CURVE_Q23_AS_POINTS)\"\",-1\n\n\n#q=25
#5-is-p.e.-mod-25.
#so-we-pick-f=5.
#then,2f^2=2*23=16
#4f=-20

#X^4-Y^4-Z^4+2f^2Y^2Z^2+4fX^2YZ:

#(1,-1,0,0,0,0,0,0,0,0,2f^2,4f,0,0)
EDGE_CURVE_Q25_EQUATION="1,4,4,0,0,0,0,0,0,0,0,16,20,0,0"
EDGE_CURVE_Q25_AS_POINTS="4,5,6,7,8,28,29,30,31,190,192,209,223,256,27
1,284,298,412,420,561,566,612,620,633,649"
FILE_Q25="orbit,curve,pts_on_curve,bitangents,go\n"n0,"$(EDGE_CURVE_Q25_EQUATION)\"","$(EDGE_CURVE_Q25_AS_POINTS)\"\",-1\n\nOCN12_PTS_Q25="0,1,2,3,4,5,28,37,87,147,213,217,220,269,303,335,356,377,412,468,5
58,562,565,629"
OCN41_PTS_Q25="0,1,2,3,4,5,76,137,182,201,247,265,292,303,358,360,373,382,420,472
,505,547,564,607"

OCN12_PTS_Q25="0,1,2,3,4,5,28,37,87,147,213,217,220,269,303,335,356,377,412,468,5
58,562,565,629"
OCN41_PTS_Q25="0,1,2,3,4,5,76,137,182,201,247,265,292,303,358,360,373,382,420,472
,505,547,564,607"
FILE_Q29="orbit,curve,pts_on_curve,bitangents.go\n\n\nFILE_Q41="orbit,curve,pts_on_curve,bitangents.go\n\n\nFILE_Q61="orbit,curve,pts_on_curve,bitangents.go\n\n\n
\n
EDGE_CURVE_Q61_EQUATION="1,2,0,0,0,0,0,0,0,0,0,0,2,1,0,0"

EDGE_CURVE_Q61_AS_POINTS="4,5,-50,67,84,85,130,-147,164,210,-227,244,245,
-246,330,-332,364,403,-409,412,423,-428,491,496,521,546,-572,576,585,
-590,-652,655,688,-727,735,-737,764,789,-822,823,830,-835,901,-910,924,
953,-983,-993,1017,1049,1068,1079,-1091,-1128,-1147,1159,1162,-1189,1226,
1229,-1236,-1242,-1314,-1319,-1346,1368,1383,1385,-1393,1399,1475,-1482,-151
0,1525,1550,1560,1572,1601,1630,-1633,1639,1645,1713,1721,1746,1778,
0,2053,2080,2119,2131,2144,2181,-2198,2202,-2208,2213,2298,2300,2320,
2331,2377,2392,2404,2425,2445,2456,2459,2487,-2540,2544,2564,2573,260
4,2612,2623,2653,2705,2710,-2736,2746,-2758,2761,-2790,2813,2869,2889,
2893,2903,-2951,2967,-2976,2987,-3011,3024,-3036,3071,3115,3122,-3151,-315
6,-3190,3197,-3220,3234,-3282,3289,-3310,3320,-3361,3374,3390,3400,-3407,
3411,-3443,-3483,-3522,-3582,-3575,3560,3574,-3583,3607,3640,3675,-3689,-370
1,-3723,3772,-3774,3790,-3791,3809,3810,-3853,3871,3935,3954,3986,-3993,
4020,4040,-4084,-4099,-4108,-4120,-4172,4181,-4203,4212,-4254,-4266,-4280,-429
2,-4329,4345,-4352,-4360,-4390,4396,-4427,-4443,-4484,-4498,4512,4531,-4564,
4568,-4591,-4599,-4632,-4643,-4647,-4673,-4733,-4752,-4758,-4766,-4783,-486,
481,4837,-4896,-4911,-4919,-4931,-4953,-4961,-4973,-5004,-5051,-5079,-5093,
5083,-5136,-5140,-5165,-5176,-5186,-5217,-5231,-5250,-5311,-5329,-5393,-5394,
-5411,-5412,-5463,-5480,-5486,-5496,-5540,-5560,-5575,-5588,-5605,-5611,-5643,
5657,-5704,-5720,-5728,-5742,-5768,-5775,-5800,-5821,-5873,-5883,5903,-5913,-5951,
-5957,-5988,5991,6008,6021,6031,-6067,-6106,-6114,-6136,-6149,6166,-6175,
6200,6234,-6276,-6280,-6311,-6313,-6348,-6363,-6374,-6395,-6440,-6452,-6468,
-6480,-6520,-6529,-6550,-6559,-6566,-6570,-6603,6641"

MONKEY_SADDLE_CUBIC="1,0,0,0,-3,0,0,0,0,0,0,0,0,0,0,0,0,-1,0"

ECKARDT_CUBIC_DEFORM1_LEX="0,-10,0,-8,10,-25,2,0,-20,-8,-20,-10,-24,-10
,-2,-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="-2,0,0,0,2,0,0,2,0,0,0,0,0,-2,0,2,0,0,0,-2,0,2,0,0,-2"

BEAUVILLE_QUINTIC_LEX_56="-44,-228,400,315,-396,-852,
-512,-553,-1050,-354,284,504,-62,-707,-1390,-1010,\n281,-167,-1644,-1024,-72,-196,192,373,322,78,150,\n966,1540,348,-475,-492,1063,1550,390,0,96,3,-337,\n-426,-66,425,-673,-156,-216,-223,-60,1543,1998,618,"
# Chapter 2: Getting Started

### SECTION ORBITER SESSION:

```
$ (ORBITER_PATH) orbiter.out
```

# Section 2.3: Makefiles and Shell Scripts

### SECTION_MAKEFILES_AND_Shell_SCRIPTS:

# Section 2.4: Objects and Activities
SECTION_OBJECTS_AND_ACTIVITIES:

example_set: $(ORBITER_PATH)orbiter.out -v.2 -define S -set -here "2,3,5,7,11,13" -end

object_F_2: $(ORBITER_PATH)orbiter.out -v.3 -define F -finite_field -q 2 -end

object_PG_3_2: $(ORBITER_PATH)orbiter.out

vector_ex: $(ORBITER_PATH)orbiter.out -v.2

create_BLT_5_1: $(ORBITER_PATH)orbiter.out -v.2
create_surface_4.0:

```bash
$ (ORBITER_PATH) orbiter.out -v 3.1
```

```bash
define F finite_field -q 4 -end
```

```bash
define P projective_space 3 F -end
```

```bash
with P -do
```

```bash
projective_space_activity
```

```bash
define_surface S4 0 -q 4 -catalogue 0 -end
```

```bash
end
```

```bash
define_surface -projective_space activity
```

```bash
define P projective_space -do
```

```bash
end
```

```bash
with S4 0 -do
```

```bash
end
```

```bash
end
```

### Section 2.6: Set-Builder

SECTION_SET_BUILDER:

```bash
set_of_primes:
```

```bash
define S -set here "2,3,5,7,11,13" -end
```

```bash
print_symbols
```

```bash
set_interval:
```

```bash
define S -set loop 0 64 1 -end
```

```bash
print_symbols
```

```bash
set_builder_examples:
```

```bash
define S -set loop 0 64 1 -end
```

```bash
define S -set loop 0 32 1 -affine_function 2 1 -end
```

```bash
define S -set loop 0 16 1 -affine_function 4 2 -clone_with_affine_function 4 3 -end
```

```bash
set_builder -set_builder -set_builder -set_builder -loop 0 4 1 -affine_function 1 4
```

```bash
set_builder -loop 0 32 1 -affine_function 1 4 -end
```

```bash
set_builder -loop 0 16 1 -affine_function 1 4 -end
```

```bash
clone_with_affine_function 1 12 -end -clone_with_affine_function 1 16
```

```bash
set_builder -set_builder -loop 0 8 1 -affine_function 1 8
```

```bash
clone_with_affine_function 1 24 -end -clone_with_affine_function 1 32 -end
```

```bash
set_builder -loop 0 16 1 -affine_function 1 16 -clone_with_affine_function 1 16
```

```bash
clone_with_affine_function 1 32 -end
```

```bash
set_builder -set_builder -loop 0 4 1 -affine_function 1 4
```

```bash
clone_with_affine_function 1 12 -end -clone_with_affine_function 1 16
```

```bash
set_builder -set_builder -loop 0 8 1 -affine_function 1 8
```

```bash
clone_with_affine_function 1 24 -end -clone_with_affine_function 1 32 -end
```

```bash
set_builder -loop 0 16 1 -affine_function 1 16 -clone_with_affine_function 1 16
```

```bash
clone_with_affine_function 1 32 -end
```

```bash
set_builder -set_builder -loop 0 4 1 -affine_function 1 4
```

```bash
clone_with_affine_function 1 12 -end -clone_with_affine_function 1 16
```

```bash
set_builder -set_builder -loop 0 8 1 -affine_function 1 8
```

```bash
clone_with_affine_function 1 24 -end -clone_with_affine_function 1 32 -end
```

```bash
set_builder -loop 0 16 1 -affine_function 1 16 -clone_with_affine_function 1 16
```
Section 2.7: Vector Builder

SECTION_VECTOR BUILDER:

vector_example1:

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

\$-define F-finite_field-q5-end\$

\$-define v-vector-field F-dense"0,1,2,3,4"-end\$

\$-print_symbols\$

vector_example2:

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

\$-define F-finite_field-q5-end\$

\$-define v-vector-field F-format-2-dense"0,1,2,3,4,0"-end\$

\$-print_symbols\$

vector_example_sparse:

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

\$-define F-finite_field-q5-end\$

\$-define v-vector-field F-format-4-sparse-20"1,0,1,19"-end\$

\$-print_symbols\$

vector_example_repeat:

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

\$-define F-finite_field-q5-end\$

\$-define v-vector-repeat"0,1,2,3"-11-end\$

\$-print_symbols\$

vector_example_all_one_11:

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

\$-define v-vector-repeat-1-11-end\$

\$-print_symbols\$

matrix_example1:

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

\$-define F-finite_field-q2-end\$

\$-define v-vector-field F-format-4-dense$(HAMMING_CODE_GENERATOR)-end\$

\$-print_symbols\$

406
matrix_example_co_1:

$ (ORBITER_PATH) orbiter.out -v.2 \ 
$define F -finite_field -q.2 -end \ 
$define v -vector -field F -format 22 -compact $(CONWAY_GEN1) -end \ 
$print_symbols

# Section 2.8: Input Streams

SECTION_INPUT_STREAMS:

# Chapter 3: Basic Algebra

SECTION_BASIC_NUMBER_THEORY:

PR7:  
$ (ORBITER_PATH) orbiter.out -v.1 -smallest_primitive_root.7

PR11:  
$ (ORBITER_PATH) orbiter.out -v.1 -smallest_primitive_root.11

PR13:  
$ (ORBITER_PATH) orbiter.out -v.1 -smallest_primitive_root.13

PR17:  
$ (ORBITER_PATH) orbiter.out -v.1 -smallest_primitive_root.17

PR19:  
$ (ORBITER_PATH) orbiter.out -v.1 -smallest_primitive_root.19
PR23:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{smallest.primitive.root:23}

PR29:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{smallest.primitive.root:29}

PR31:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{smallest.primitive.root:31}

PR37:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{smallest.primitive.root:37}

PR100:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{primitive.root.interval:2.100}

PR1000:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{primitive.root.interval:2.1000}

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{csv_file_latex-primitive_element_table_2.1000.csv}

\text{pdflatex-primitive_element_table_2.1000.tex}

\text{open-primitive_element_table_2.1000.pdf}

PE\_number\_1000:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{number_ofPrimitive Roots.interval:2.1000}

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{csv_file_latex-table_number_of_pe_2.1000.csv}

\text{pdflatex-table_number_of_pe_2.1000.tex}

\text{open-table_number_of_pe_2.1000.pdf}

Eulerfunction\_10000:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.1.-\text{number_ofPrimitive Roots.interval:10000.100}

#-randomized-algo:

PR2:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.5.-\text{primitive.root:915839}

a-primitive.root.modulo:915839-is:43085

PM2:

\$(\text{ORBITER PATH})\text{orbiter.out}-v.5.-\text{power_mod:43085-49842:915839}

the-power-of:43085.to-the:49842-mod:915839-is:487320

DL2:
The discrete log is 49842 since 487320 = 43085^49842 mod 915839, time: 0:22

IM_723:

IM_3.19:

IM:

IM_gcd:

PM3a:

sqrt_mod:

sqrt_5_mod_11:

sqrt_5_mod_19:

draw_mod_8:

draw_mod_13:
draw_mod_3:
\begin{verbatim}
\$\langle ORBITER\_PATH\rangle\verb\orbiter.out\-v.2\$
\end{verbatim}
- draw_options::embedded::nodes_empty::end\n- draw_mod_n::n.3::file::mod_3::end.
- cyclotomic_sets::2"1"::end
- pdflatex::mod_3::draw.tex
- open::mod_3::draw.pdf

\begin{verbatim}
\$\langle ORBITER\_PATH\rangle\verb\orbiter.out\-v.2\$
\end{verbatim}
- draw_options::embedded::nodes_empty::end\n- draw_mod_n::n.3::file::mod_3::c\n- pdflatex::mod_3::c::draw.tex
- open::mod_3::c::draw.pdf

\begin{verbatim}
\$\langle ORBITER\_PATH\rangle\verb\orbiter.out\-v.2\$
\end{verbatim}
- draw_options::embedded::nodes_empty::end\n- draw_mod_n::n.4::file::mod_4::end.
- pdflatex::mod_4::draw.tex
- open::mod_4::draw.pdf

\begin{verbatim}
\$\langle ORBITER\_PATH\rangle\verb\orbiter.out\-v.2\$
\end{verbatim}
- draw_options::embedded::nodes_empty::end\n- draw_mod_n::n.6::file::mod_6::end.
- pdflatex::mod_6::draw.tex
- open::mod_6::draw.pdf

\begin{verbatim}
\$\langle ORBITER\_PATH\rangle\verb\orbiter.out\-v.2\$
\end{verbatim}
- draw_options::embedded::nodes_empty::end\n- draw_mod_n::n.7::file::mod_7::end.
- pdflatex::mod_7::draw.tex
- open::mod_7::draw.pdf

\begin{verbatim}
\$\langle ORBITER\_PATH\rangle\verb\orbiter.out\-v.2\$
\end{verbatim}
- draw_options::embedded::end\n- draw_mod_n::n.15::file::mod_15::end.
- pdflatex::mod_15::draw.tex

\end{verbatim}
draw_mod_127:
\$\text{(ORBITER\_PATH)}\text{orbiter.out-\text{-v.2\textbackslash}}$
\$\text{-draw\_options-\text{-scale0.8-\text{-embedded-\text{-end}\text\backslash}}}$
\$\text{-draw\_mod\_n-\text{-n.127-\text{-file\_mod\_127-\text{-power\_cycle3-\text{-end}}}}}$
\$\text{pdflatex\_mod\_127\_draw.tex}$
\$\text{open\_mod\_127\_draw.pdf}$

sqrt_mod_20_31:
\$\text{(ORBITER\_PATH)}\text{orbiter.out-\text{-v.2-\text{-square\_root\_mod\_20\_31}}}$

Section 3.2: Prime Fields

SECTION\_PRIME\_FIELDS:

F_2:
\$\text{(ORBITER\_PATH)}\text{orbiter.out-\text{-v.3-\text{-list\_arguments\text\backslash}}}$
\$\text{-defineF-\text{-finite\_field\_q2-\text{-end\text\backslash}}}$
\$\text{-withF-\text{-do-\text{-finite\_field\_activity-\text{-cheat\_sheet\_GF-\text{-end}}}}}$
\$\text{pdflatex\_GF\_2.tex}$
\$\text{open\_GF\_2.pdf}$

F_3:
\$\text{(ORBITER\_PATH)}\text{orbiter.out-\text{-v.3\text\backslash}}$
\$\text{-defineF-\text{-finite\_field\_q3-\text{-end\text\backslash}}}$
\$\text{-withF-\text{-do-\text{-finite\_field\_activity-\text{-cheat\_sheet\_GF-\text{-end}}}}}$
\$\text{pdflatex\_GF\_3.tex}$
\$\text{open\_GF\_3.pdf}$

F_5:
\$\text{(ORBITER\_PATH)}\text{orbiter.out-\text{-v.3\text\backslash}}$
\$\text{-defineF-\text{-finite\_field\_q5-\text{-end\text\backslash}}}$
\$\text{-withF-\text{-do-\text{-finite\_field\_activity-\text{-cheat\_sheet\_GF-\text{-end}}}}}$
\$\text{pdflatex\_GF\_5.tex}$
\$\text{open\_GF\_5.pdf}$

F_7:
\$\text{(ORBITER\_PATH)}\text{orbiter.out-\text{-v.3\text\backslash}}$
\$\text{-defineF-\text{-finite\_field\_q7-\text{-end\text\backslash}}}$
$\text{F}_{13}$:
\begin{verbatim}
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v.3\$
-define\text{-finite_field}\ -q13\-end\$
-with\text{-do}\-finite_field\text{-activity}\-cheat\_sheet GF\-end
pdflatex GF_13.tex
open GF_13.pdf
\end{verbatim}

$\text{F}_{17}$:
\begin{verbatim}
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v.3\$
-define\text{-finite_field}\ -q17\-end\$
-with\text{-do}\-finite_field\text{-activity}\-cheat\_sheet GF\-end
pdflatex GF_17.tex
open GF_17.pdf
\end{verbatim}

$\text{F}_{19}$:
\begin{verbatim}
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v.3\$
-define\text{-finite_field}\ -q19\-end\$
-with\text{-do}\-finite_field\text{-activity}\-cheat\_sheet GF\-end
pdflatex GF_19.tex
open GF_19.pdf
\end{verbatim}

$\text{F}_{31}$:
\begin{verbatim}
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v.3\$
-define\text{-finite_field}\ -q31\-end\$
-with\text{-do}\-finite_field\text{-activity}\-cheat\_sheet GF\-end
\end{verbatim}

$\text{F}_{127}$:
\begin{verbatim}
$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v.3\$
-define\text{-finite_field}\ -q127\-end\$
-with\text{-do}\-finite_field\text{-activity}\-cheat\_sheet GF\-end
\end{verbatim}

\texttt{GF}
# Section 3.3: Polynomials over Finite Fields

SECTION_POLYNOMIALS:

# Check which polynomials are irreducible and which are primitive:

sift_polynomials_3:
   $(ORBITER_PATH)orbiter.out -v 2 \n   define F - finite_field -q 2 - end \n   with F - do \n   finite_field.activity - sift_polynomials 8 16 - end

sift_polynomials_4:
   $(ORBITER_PATH)orbiter.out -v 2 \n   define F - finite_field -q 2 - end \n   with F - do \n   finite_field.activity - sift_polynomials 16 32 - end

poly_division:
   $(ORBITER_PATH)orbiter.out -v 2 \n   define F - finite_field -q 2 - end \n   with F - do \n   finite_field.activity \n   polynomial_division "1,0,0,0,0,0,0,0,1" "1,0,1,1" - end

poly_division2:
   $(ORBITER_PATH)orbiter.out -v 2 \n   define F - finite_field -q 2 - end \n   define A - vector - field F - sparse 11 "1,0,1,0" - end \n   define B - vector - field F - dense "1,0,1,1" - end \n   with F - do \n   finite_field.activity \n   polynomial_division A B - end

poly_gcd:
   $(ORBITER_PATH)orbiter.out -v 2 \n   define F - finite_field -q 2 - end \n   with F - do \n
413
poly_mult_mod1:
$(ORBITER_PATH)orbiter.out-v.2$
-define F-finite_field-q 7-end
-with F-do
-finite_field_activity
-polynomial_mult_mod."1,0,0,0,0,0,0,0,0,1"."1,0,1,1"-end

poly_mult_mod2:
$(ORBITER_PATH)orbiter.out-v.2$
-define F-finite_field-q 7-end
-with F-do
-finite_field_activity
-polynomial_mult_mod."1,0,1,1"."1,0,1,1"-end

Berlekamp_matrix_2,3:
$(ORBITER_PATH)orbiter.out-v.2$
-define F-finite_field-q 2-end
-with F-do
-finite_field_activity
-Berlekamp_matrix."1,1,0,1"-end

Berlekamp_matrix_2,4:
$(ORBITER_PATH)orbiter.out-v.2$
-define F-finite_field-q 2-end
-with F-do
-finite_field_activity
-Berlekamp_matrix."1,1,0,1"-end

Berlekamp_matrix_4,3a:
$(ORBITER_PATH)orbiter.out-v.2$
-define F-finite_field-q 4-end
-with F-do
-finite_field_activity
-Berlekamp_matrix."1,3,0,1"-end

Berlekamp_matrix_4,3b:
$(ORBITER_PATH)orbiter.out-v.2$

1655 \> \> -define F\-finite_field\-q\-4\-end\  
1656 \> \> -with F\-do\  
1657 \> \> -finite_field_activity\  
1658 \> \> -Berlekamp_matrix\"1,3,1,1\"\-end  
1659  
1660  
1661  
1662  
1663 \> find\_roots\_a:\  
1664 \> \> $(\text{ORBITER\_PATH})\text{orbiter.out\-v\-2}\  
1665 \> \> \> -define F\-finite_field\-q\-19\-end\  
1666 \> \> \> -with F\-do\  
1667 \> \> \> -finite_field_activity\  
1668 \> \> \> -polynomial\_find\_roots\"18,1,1\"\-end  
1669  
1670 \> find\_roots\_b:\  
1671 \> \> $(\text{ORBITER\_PATH})\text{orbiter.out\-v\-2}\  
1672 \> \> \> -define F\-finite_field\-q\-19\-end\  
1673 \> \> \> -with F\-do\  
1674 \> \> \> -finite_field_activity\  
1675 \> \> \> -polynomial\_find\_roots\"1,3,1\"\-end  
1676  
1677 \> find\_roots\_c:\  
1678 \> \> $(\text{ORBITER\_PATH})\text{orbiter.out\-v\-2}\  
1679 \> \> \> -define F\-finite_field\-q\-19\-end\  
1680 \> \> \> -with F\-do\  
1681 \> \> \> -finite_field_activity\  
1682 \> \> \> -polynomial\_find\_roots\"1,16,1\"\-end  
1683  
1684 \> find\_roots\_d:\  
1685 \> \> $(\text{ORBITER\_PATH})\text{orbiter.out\-v\-2}\  
1686 \> \> \> -define F\-finite_field\-q\-19\-end\  
1687 \> \> \> -with F\-do\  
1688 \> \> \> -finite_field_activity\  
1689 \> \> \> -polynomial\_find\_roots\"1,18,1\"\-end  
1690  
1691 \> find\_roots\_e:\  
1692 \> \> $(\text{ORBITER\_PATH})\text{orbiter.out\-v\-2}\  
1693 \> \> \> -define F\-finite_field\-q\-19\-end\  
1694 \> \> \> -with F\-do\  
1695 \> \> \> -finite_field_activity\  
1696 \> \> \> -polynomial\_find\_roots\"1,16,3\"\-end  
1697  
1698  
1699 \> roots\_over\_F2:\  
1700 \> \> $(\text{ORBITER\_PATH})\text{orbiter.out\-v\-2}\  
1701 \> \> \> -define F\-finite_field\-q\2\-end\  

415
roots_over_F8:

$\text{(ORBITER}\text{PATH})\text{orbiter.out}$--v.2:

$\text{define}\text{-}\text{finite}\text{-}\text{field}\text{-}q8\text{-}\text{override}\text{-}\text{polynomial}\text{-}11\text{-}\text{end}$

$\text{with}\text{-}\text{F}\text{-}\text{do}$

$\text{finite}\text{-}\text{field}\text{-}\text{activity}$

$\text{polynomial}\text{-}\text{find}\text{-}\text{roots}\text{:\{}0,1,0,1,1,1\}\text{-end}$

roots_over_F8:

$\text{(ORBITER}\text{PATH})\text{orbiter.out}$--v.2:

$\text{define}\text{-}\text{finite}\text{-}\text{field}\text{-}q8\text{-}\text{override}\text{-}\text{polynomial}\text{-}11\text{-}\text{end}$

$\text{with}\text{-}\text{F}\text{-}\text{do}$

$\text{finite}\text{-}\text{field}\text{-}\text{activity}$

$\text{polynomial}\text{-}\text{find}\text{-}\text{roots}\text{:\{}0,1,0,1,1,1\}\text{-end}$

# degree and then order of the field of coefficients:

irred_3_2:

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

$\text{define}\text{-}\text{finite}\text{-}\text{field}\text{-}q2\text{-}\text{end}$

$\text{with}\text{-}\text{F}\text{-}\text{do}$

$\text{finite}\text{-}\text{field}\text{-}\text{activity}$

$\text{make}\text{-}\text{table}\text{\{}\text{irreducible}\text{\{}polynomials}\text{\}}\text{-}3\text{-}\text{end}$

$\text{pdflatex}\text{Irred}\text{q2}\text{d3.tex}$

$\text{open}\text{Irred}\text{q2}\text{d3.pdf}$

irred_4_2:

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

$\text{define}\text{-}\text{finite}\text{-}\text{field}\text{-}q2\text{-}\text{end}$

$\text{with}\text{-}\text{F}\text{-}\text{do}$

$\text{finite}\text{-}\text{field}\text{-}\text{activity}$

$\text{make}\text{-}\text{table}\text{\{}\text{irreducible}\text{\{}polynomials\text{\}}\text{-}4\text{-}\text{end}$

$\text{pdflatex}\text{Irred}\text{q2}\text{d4.tex}$

$\text{open}\text{Irred}\text{q2}\text{d4.pdf}$

irred_5_2:

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

$\text{define}\text{-}\text{finite}\text{-}\text{field}\text{-}q2\text{-}\text{end}$

$\text{with}\text{-}\text{F}\text{-}\text{do}$

$\text{finite}\text{-}\text{field}\text{-}\text{activity}$

$\text{make}\text{-}\text{table}\text{\{}\text{irreducible}\text{\{}polynomials\text{\}}\text{-}5\text{-}\text{end}$

$\text{pdflatex}\text{Irred}\text{q2}\text{d5.tex}$

$\text{open}\text{Irred}\text{q2}\text{d5.pdf}$
1749
1750 #6 polys
1751
1752 irred_6_2:
1753 > $(ORBITER_PATH)orbir.out-v.3\ 
1754 > -define F -finite_field -q 2 -end \ 
1755 > -with F -do \ 
1756 > -finite_field_activity \ 
1757 > -make_table_of_irreducible_polynomials 6 -end \ 
1758 > pdflatex Irred_q2_d6.tex \ 
1759 > open Irred_q2_d6.pdf
1760
1761 #9 polys
1762
1763 irred_7_2:
1764 > $(ORBITER_PATH)orbir.out-v.3\ 
1765 > -define F -finite_field -q 2 -end \ 
1766 > -with F -do \ 
1767 > -finite_field_activity \ 
1768 > -make_table_of_irreducible_polynomials 7 -end \ 
1769 > pdflatex Irred_q2_d7.tex \ 
1770 > open Irred_q2_d7.pdf
1771
1772 #18 polys
1773
1774 irred_8_2:
1775 > $(ORBITER_PATH)orbir.out-v.3\ 
1776 > -define F -finite_field -q 2 -end \ 
1777 > -with F -do \ 
1778 > -finite_field_activity \ 
1779 > -make_table_of_irreducible_polynomials 8 -end \ 
1780 > pdflatex Irred_q2_d8.tex \ 
1781 > open Irred_q2_d8.pdf
1782
1783 #30 polys
1784
1785 irred_9_2:
1786 > $(ORBITER_PATH)orbir.out-v.3\ 
1787 > -define F -finite_field -q 2 -end \ 
1788 > -with F -do \ 
1789 > -finite_field_activity \ 
1790 > -make_table_of_irreducible_polynomials 9 -end \ 
1791 > pdflatex Irred_q2_d9.tex \ 
1792 > open Irred_q2_d9.pdf
1793
1794 #56 polys
1795
irred_10_2:
  $(\text{ORBITER\_PATH})\text{orbiter.out-v.3}\$
  -define F=finite_field-q2-end
  -with F-do
  -finite_field_activity
  -make_table_of_irreducible_polynomials10-end
  pdflatex-Irred_q2_d10.tex
  open-Irred_q2_d10.pdf
  #99-polys

irred_2_4:
  $(\text{ORBITER\_PATH})\text{orbiter.out-v.3}\$
  -define F=finite_field-q4-end
  -with F-do
  -finite_field_activity
  -make_table_of_irreducible_polynomials2-end
  pdflatex-Irred_q4_d2.tex
  open-Irred_q4_d2.pdf
  #6-polys

irred_3_4:
  $(\text{ORBITER\_PATH})\text{orbiter.out-v.6}\$
  -define F=finite_field-q4-end
  -with F-do
  -finite_field_activity
  -make_table_of_irreducible_polynomials3-end
  pdflatex-Irred_q4_d3.tex
  open-Irred_q4_d3.pdf
  #20-polys

order_of_2_mod_n:
  $(\text{ORBITER\_PATH})\text{orbiter.out-v.3-order_of_q_mod_n23300}$
  $(\text{ORBITER\_PATH})\text{orbiter.out-v.1-csv_file_latex-order_of_q_mod_n_q23300.csv}$
  pdflatex-order_of_q_mod_n_q23300.tex
  open-order_of_q_mod_n_q23300.pdf

search_primitive_poly_2:
  $(\text{ORBITER\_PATH})\text{orbiter.out-v.3}\$
  -search_for_primitive_polynomial_in_range22210-#1-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 3 2
   .60
```

```bash
search_primitive_poly_4:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:4 4 2
   .30
```

```bash
search_primitive_poly_5:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:5 5 2
   .30
```

```bash
search_primitive_poly_7:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:7 7 2
   .20
```

```bash
search_primitive_poly_8:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:8 8 2
   .20
```

```bash
search_primitive_poly_9:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:9 9 2
   .15
```

```bash
search_primitive_poly_11:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:11 11
   .2 15
```

```bash
search_primitive_poly_13:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:13 13
   .2 15
```

```bash
search_primitive_poly_degree_16:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:2 2 1
   6 16
```

```bash
search_primitive_poly_32:

```(ORBITER PATH)orbiter.out -v 6 -search_for_primitive_polynomial_in_range:32 32
   .2 10
```

```bash
```
```bash
1880  poly_mult_mod_F4:
1881  $(ORBITER_PATH)orbiter.out-v.2:
1882  define F finite_field q.2-end:
1883  with F do:
1884  finite_field_activity:
1885  polynomial_mult_mod "1,1"."1,1,1"-end
1886  $(ORBITER_PATH)orbiter.out-v.2:
1887  define F finite_field q.2-end:
1888  with F do:
1889  finite_field_activity:
1890  polynomial_mult_mod "1,1,1"."1,1,1"-end
1891  $(ORBITER_PATH)orbiter.out-v.2:
1892  define F finite_field q.2-end:
1893  with F do:
1894  finite_field_activity:
1895  polynomial_mult_mod "0,1"."1,1,1"-end
1896  $(ORBITER_PATH)orbiter.out-v.2:
1897  define F finite_field q.2-end:
1898  with F do:
1899  finite_field_activity:
1900  polynomial_mult_mod "0,1"."0,1,1,1"-end
1901
mult_polynomials_2_5_7:
1902  $(ORBITER_PATH)orbiter.out-v.2:
1903  define F finite_field q.2-end:
1904  with F do:
1905  finite_field_activity mult_polynomials_5_7-end
1906  pdflatex polynomial_mult_5_7.tex
1907  open polynomial_mult_5_7.pdf
1908
division_ranked_2_27_13:
1909  $(ORBITER_PATH)orbiter.out-v.2:
1910  define F finite_field q.2-end:
1911  with F do:
1912  finite_field_activity polynomial_division_ranked_27_13-end
1913  pdflatex polynomial_division_27_13.tex
1914  open polynomial_division_27_13.pdf
1915
division_ranked_2_8_15:
1916  $(ORBITER_PATH)orbiter.out-v.2:
1917  define F finite_field q.2-end:
1918  with F do:
1919  finite_field_activity mult_polynomials_8_15-end
1920  pdflatex polynomial_mult_8_15.tex
1921
420
```
mult.polynomials_2.7.7:
$\text{(ORBITER PATH)orbiter.out -v 2:}$
\begin{verbatim}
define F finite_field -q 2 -end
with F -do
-finite_field_activity -mult.polynomials_7.7 -end
pdflatex polynomial_mult_7.7.tex
\end{verbatim}
open.polynomial_mult_7.7.pdf

mult.polynomials_2.4.6:
$\text{(ORBITER PATH)orbiter.out -v 2:}$
\begin{verbatim}
define F finite_field -q 2 -end
with F -do
-finite_field_activity -mult.polynomials_4.6 -end
pdflatex polynomial_mult_4.6.tex
\end{verbatim}
open.polynomial_mult_4.6.pdf

polynomial_division_ranked_2.24.13:
$\text{(ORBITER PATH)orbiter.out -v 2:}$
\begin{verbatim}
define F finite_field -q 2 -end
with F -do
-finite_field_activity -polynomial_division_ranked_24.13 -end
pdflatex polynomial_division_24.13.tex
\end{verbatim}
open.polynomial_division_24.13.pdf

mult.polynomials_1024_999_997:
$\text{(ORBITER PATH)orbiter.out -v 2:}$
\begin{verbatim}
define F finite_field -q 2 -end
with F -do
-finite_field_activity -mult.polynomials_999_997 -end
pdflatex polynomial_mult_999_997.tex
\end{verbatim}
1974  \( \triangleright \) open-polynomial_mult_999_997.pdf
1975
1976
1977  polynomial_division_ranked_2_349147_1033:
1978  \( \triangleright \) $(\text{ORBITER}\ PATH)\ orbiter\.out\ -v\ 2\ \backslash$
1979  \( \triangleright \) \( -\text{define}\ F\ -\text{finite}\_\text{field}\ -q\ 2\ -\text{end}\ \backslash$
1980  \( \triangleright \) \( -\text{with}\ F\ -\text{do}\ \backslash$
1981  \( \triangleright \) \( -\text{finite}\_\text{field}\_\text{activity}\ \backslash$
1982  \( \triangleright \) \( -\text{polynomial}\_\text{division}\_\text{ranked}\_349147\_1033\ \backslash$
1983  \( \triangleright \) \( -\text{end}\$
1984  \( \triangleright \) pdflatex-polynomial_division_349147_1033.tex
1985  \( \triangleright \) open-polynomial_division_349147_1033.pdf
1986
1987
1988
1989  mult_polynomials_1024_999_997_check:
1990  \( \triangleright \) $(\text{ORBITER}\ PATH)\ orbiter\.out\ -v\ 3\ \backslash$
1991  \( \triangleright \) \( -\text{define}\ F\ -\text{finite}\_\text{field}\ -q\ 1024\ -\text{end}\ \backslash$
1992  \( \triangleright \) \( -\text{with}\ F\ -\text{do}\ \backslash$
1993  \( \triangleright \) \( -\text{finite}\_\text{field}\_\text{activity}\_\text{-parse}\_\text{and}\_\text{evaluate}\ \backslash$
1994  \( \triangleright \) \"test\".\"\ a*b\".\"a=999,b=997\".-end
1995
1996  \#evaluates\ to\ 61
1997
1998
1999  mult_polynomials_17_12:
2000  \( \triangleright \) $(\text{ORBITER}\ PATH)\ orbiter\.out\ -v\ 2\ \backslash$
2001  \( \triangleright \) \( -\text{define}\ F\ -\text{finite}\_\text{field}\ -q\ 2\ -\text{end}\ \backslash$
2002  \( \triangleright \) \( -\text{with}\ F\ -\text{do}\ \backslash$
2003  \( \triangleright \) \( -\text{finite}\_\text{field}\_\text{activity}\_\text{-mult}\_\text{polynomials}\_17\_12\.-end$
2004  \( \triangleright \) pdflatex-polynomial_mult_17_12.tex
2005  \( \triangleright \) open-polynomial_mult_17_12.pdf
2006
2007  \#gives\ 204
2008
2009  polynomial_division_ranked_2_204_37:
2010  \( \triangleright \) $(\text{ORBITER}\ PATH)\ orbiter\.out\ -v\ 2\ \backslash$
2011  \( \triangleright \) \( -\text{define}\ F\ -\text{finite}\_\text{field}\ -q\ 2\ -\text{end}\ \backslash$
2012  \( \triangleright \) \( -\text{with}\ F\ -\text{do}\ \backslash$
2013  \( \triangleright \) \( -\text{finite}\_\text{field}\_\text{activity}\_\text{-polynomial}\_\text{division}\_\text{ranked}\_204\_37\.-end$
2014  \( \triangleright \) pdflatex-polynomial_division_204_37.tex
2015  \( \triangleright \) open-polynomial_division_204_37.pdf
2016
2017  \#answer\ is\ 18
2018
2019
2020
Section 3.4: Extension Fields

### Section_Extension.Fields:

```bash
F_4:
$\text{(ORBITER\_PATH)orbiter.out}\ -v\cdot3\$
> -define F\_finite_field\_q_4\_end
> -with F\_do\_finite_field\_activity\_cheat_sheet_GF\_end

F_4.tables:
$\text{(ORBITER\_PATH)orbiter.out}\ -v\cdot3\$
> -define F\_finite_field\_q_4\_end
> -with F\_do\_finite_field\_activity\_cheat_sheet_GF\_end

> -draw_matrix\_input\_csv\_file GF\_q_4\_addition_table.csv
> -box_width 40 \-bit_depth 24 \-partition 3\_3\_3\_end

> -draw_matrix\_input\_csv\_file GF\_q_4\_multiplication_table.csv
> -box_width 40 \-bit_depth 24 \-partition 3\_3\_3\_end

> #pdflatex GF_4.tex

> #open GF_4.pdf

trace_4:
$\text{(ORBITER\_PATH)orbiter.out}\ -v\cdot3\$
> -define F\_finite_field\_q_4\_end
> -with F\_do\_finite_field\_activity\_trace\_end

$\text{(ORBITER\_PATH)orbiter.out}\ -v\cdot3\$
-define F\_finite_field\_q_4\_end
> -with F\_do\_finite_field\_activity\_algebraic_normal_form F\_q_4\_trace.csv 2\_end

trace_4\_WH\_transform:
$\text{(ORBITER\_PATH)orbiter.out}\ -v\cdot3\$
> -define F\_finite_field\_q_4\_end
> -with F\_do\_finite_field\_activity\_Walsh_Hadamard_transform F\_q_4\_trace.csv 2\_end
```
F_8:

```bash
$(ORBITER_PATH) orbiter.out -v.3 \ndefine F -finite_field -q 8 -end \n-with F -do -finite_field_activity -cheat_sheet_GF -end
```

pdflatex GF_8.tex
open GF_8.pdf

```
# compute 2*3 and 3*3 in F8:
```

```
F8_arithmetic:
$(ORBITER_PATH) orbiter.out -v.3 -define F -finite_field -q 8 -end \n-define e1 -formula "test"."test".""a*b"... \n-define e2 -formula "a*b"."a"^2."a"^3 \n-evaluate e1.a=2,b=3 -end \n-evaluate e2.a=2,b=3 -end
```

```
F_16_7_power_5:
$(ORBITER_PATH) orbiter.out -v.3 -define F -finite_field -q 16 -end \n-with F -do -finite_field_activity \n-parse_and_evaluate "test".""(a*a)*(a*a)*a"".""a=7" -end
dot -Tpng test.gv > test.png
```

```
# the answer is 11.
```

```
F_8_near_bent_5:
$(ORBITER_PATH) orbiter.out -v.3 \n-define F -finite_field -q 8 -end \n-with F -do -finite_field_activity -identity_function -F8.csv -end
```

```
$(ORBITER_PATH) orbiter.out -v.3 \n-define F -finite_field -q 8 -end \n-with F -do -finite_field_activity -apply_power_function -F8.csv.5 -end
```

```
$(ORBITE...
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136

2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148

.csv·4·-end
. $(ORBITER PATH)orbiter.out·-v·3·-reformat·F8 power 5 trace.csv·F8 power 5 trace
8x1.csv·1
. $(ORBITER PATH)orbiter.out·-v·2·-draw matrix·\
. . -input csv file·F8 power 5 trace 8x1.csv·\
. . -box width·40·-bit depth·24··-partition·4·8·1·-end
. $(ORBITER PATH)orbiter.out·-v·2·-draw matrix·\
. . -input csv file·Walsh 01 3.csv·\
. . -box width·40·-bit depth·24··-partition·4·8·8·-end
. $(ORBITER PATH)orbiter.out·-v·3·-reformat·F8 power 5 trace transformed.csv·F8 p
ower 5 trace transformed 8x1.csv·1
. $(ORBITER PATH)orbiter.out·-v·2·-draw matrix·\
. . -input csv file·F8 power 5 trace transformed 8x1.csv·\
. . -box width·40·-bit depth·24··-partition·4·8·1·-end
bent 4:
. $(ORBITER PATH)orbiter.out·-v·3·-bent·4
bent 4a:
. $(ORBITER PATH)orbiter.out·-v·3·-reformat·bent function n4.csv·bent function n4
16x1.csv·1
. $(ORBITER PATH)orbiter.out·-v·3·\
. . -define·F·-finite field·-q·2·-end·\
. . -with·F·-do·-finite field activity·\
. . -Walsh Hadamard transform·bent function n4.csv·4·-end
. $(ORBITER PATH)orbiter.out·-v·3·\
. . -reformat·bent function n4 transformed.csv·\
. . bent function n4 transformed 16x1.csv·1
. #$(ORBITER PATH)orbiter.out·-v·2·-draw matrix·-input csv file·bent function n4
transformed 16x1.csv·-box width·40·-bit depth·24··-partition·4·16·1·-end
#0·:·0·/·448·:·(·0,·0,·0,·0,·0,·0,·1,·1,·0,·1,·0,·1,·0,·1,·1,·0·)·:·27328·:·(·0,·
0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·0,·
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·)·:·
homogeneous polynomial domain::print equation
#X0*X3*X4^2·+·X1*X2*X4^2

F8
.
.
.
.
.
.
.
.

over F2 field reduction:
$(ORBITER PATH)orbiter.out·-v·2·\
. -define·F·-finite field·-q·8·-end·\
. -loop·L·0·8·1·\
. -with·F·-do·\
. -finite field activity·-field reduction·"F8 red %L"·2·1·1·"%L"·-end·\
. -end loop
$(ORBITER PATH)orbiter.out·-v·2·-loop·L·0·8·1·\
. . -draw matrix·-input csv file·F8 red %L.csv·\

425


\#generator
\text{polynomial is } X^7 + 3X^6 + 3X^5 + 2X^3 + X^2 + X \cdot \text{2}

\#generator
\text{polynomial is } X^9 + 5X^8 + 5X^6 + 4X^4 + X^3 + 2X^2 + X \cdot \text{2}

\#generator
\text{polynomial is } X^9 + 5X^8 + 5X^6 + 4X^4 + X^3 + 5X + 4

\#generator
\text{polynomial is } X^9 + 5X^8 + 5X^6 + 4X^4 + X^3 + 5X + 4
\begin{verbatim}
2195 \texttt{\$\{ORBITER\_PATH\}oribter.out\_v.3:}
2196 \texttt{-define F\_finite\_field\_q\_8\_override\_polynomial\_11\_end}
2197 \texttt{-with F\_do\_finite\_field\_activity\_make\_BCH\_code\_21\_7\_end}
2198 #generator polynomial is \( X^9 - 10 X^9 + 2 X^9 + 5 X^7 + 2 X^6 + 4 X^4 + 5 X^3 + 5 X^2 + 6 X + 6 \)
2199 F_8 BCH code d8:
2200 #after reduction: \([63, 27]_2 r=36 \)
2201 F_9:
2202 #after reduction: \([63, 27]_2 r=36 \)
2203 trace_9:
2204 \texttt{\$\{ORBITER\_PATH\}oribter.out\_v.3:}
2205 \texttt{-define F\_finite\_field\_q\_9\_end}
2206 \texttt{-with F\_do\_finite\_field\_activity\_cheat\_sheet\_GF\_end}
2207 \texttt{\$\{ORBITER\_PATH\}oribter.out\_v.2\_draw\_matrix}
2208 \texttt{-input csv_file\_GF\_q9\_addition\_table.csv:}
2209 \texttt{-box width 40\_bit depth 24\_partition\_3\_3, 6\_3, 6\_end}
2210 \texttt{\$\{ORBITER\_PATH\}oribter.out\_v.2\_draw\_matrix}
2211 \texttt{-input csv_file\_GF\_q9\_multiplication\_table.csv:}
2212 \texttt{-box width 40\_bit depth 24\_partition\_3\_2, 6\_2, 6\_end}
2213 \texttt{\$\{ORBITER\_PATH\}oribter.out\_v.2\_draw\_matrix}
2214 \texttt{-input csv_file\_GF\_q9\_addition\_table\_reordered.csv:}
2215 \texttt{-box width 40\_bit depth 24\_partition\_3\_9, 9\_end}
2216 \texttt{\$\{ORBITER\_PATH\}oribter.out\_v.2\_draw\_matrix}
2217 \texttt{-input csv_file\_GF\_q9\_multiplication\_table\_reordered.csv:}
2218 \texttt{-box width 40\_bit depth 24\_partition\_3\_8, 8\_end}
2219 \texttt{pdflatex GF_9.tex}
2220 \texttt{open GF_9.pdf}
2221 #
2222 #
2223 #
2224 #
2225 #
2226 #
2227 #
2228 #
2229 #
2230 #
2231 #
2232 #
2233 #
2234 #
2235 #
2236 #
2237 #
2238 #
2239
\end{verbatim}
F_16:

```bash
$(ORBITER_PATH)orbiner.out-v.3-
$define-F-finite_field-q.16-end-
$with-F-do-finite_field_activity-cheat_sheet_GF-end
pdflatex-GF_16.tex
```

F_16_tables:

```bash
$(ORBITER_PATH)orbiner.out-v.3-
$define-F-finite_field-q.16-end-
$with-F-do-finite_field_activity-cheat_sheet_GF-end
$(ORBITER_PATH)orbiner.out-v.2-draw_matrix-
$input_csv_file-GF_q16_addition_table.csv-
$box_width-40-bit_depth-24-partition-3-16-16-end
$(ORBITER_PATH)orbiner.out-v.2-draw_matrix-
$input_csv_file-GF_q16_multiplication_table.csv-
$box_width-40-bit_depth-24-partition-3-16-16-end
$(ORBITER_PATH)orbiner.out-v.2-draw_matrix-
$input_csv_file-GF_q16_addition_table_reordered.csv-
$box_width-40-bit_depth-24-partition-3-15-15-end
$(ORBITER_PATH)orbiner.out-v.2-draw_matrix-
$input_csv_file-GF_q16_multiplication_table_reordered.csv-
$box_width-40-bit_depth-24-partition-3-15-15-end
```

trace_16:

```bash
$(ORBITER_PATH)orbiner.out-v.3-
$define-F-finite_field-q.16-end-
$with-F-do-finite_field_activity-trace-end
$(ORBITER_PATH)orbiner.out-v.3-reformat-F_q16_trace.csv-F_q16_trace_4x4.csv-4
```

```bash
$(ORBITER_PATH)orbiner.out-v.2-draw_matrix-
$input_csv_file-F_q16_trace_4x4.csv-
$box_width-40-bit_depth-24-partition-4-4-4-end
```

F_16.bent.wrong:

```bash
$(ORBITER_PATH)orbiner.out-v.3-
$define-F-finite_field-q.16-end-
$with-F-do-finite_field_activity-identity_function-F16.csv-end
$(ORBITER_PATH)orbiner.out-v.3-
$define-F-finite_field-q.16-end-
$with-F-do-finite_field_activity-apply_power_function-F16.csv-9-end
```
2286   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.3;}
2287   \texttt{-define F\textasciitilde finite_field\textasciitilde q16\textasciitilde -end;}
2288   \texttt{-with F\textasciitilde do\textasciitilde finite_field\textasciitilde activity\textasciitilde -apply\textunderscore trace\textunderscore function\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore csv\textasciitilde -end}
2289   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.3;}
2290   \texttt{-define F\textasciitilde finite_field\textasciitilde q2\textasciitilde -end;}
2291   \texttt{-with F\textasciitilde do\textasciitilde finite_field\textasciitilde activity\textasciitilde -Walsh\textunderscore Hadamard\textunderscore transform\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textasciitilde csv\textasciitilde 4\textasciitilde -end}
2292   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.3;\textasciitilde reformat\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textasciitilde csv\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textasciitilde transformed\textasciitilde 16x1\textunderscore csv\textasciitilde 1}
2293   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.2\textasciitilde -draw\textunderscore matrix;}
2294   \texttt{-input\textunderscore csv\textunderscore file\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textasciitilde 16x1\textunderscore csv\textasciitilde ;}
2295   \texttt{-box\textunderscore width\textasciitilde 40\textasciitilde -bit\textunderscore depth\textasciitilde 24\textasciitilde -partition\textasciitilde 4\textasciitilde 16\textasciitilde 1\textasciitilde -end}
2296   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.2\textasciitilde -draw\textunderscore matrix;}
2297   \texttt{-input\textunderscore csv\textunderscore file\textasciitilde Walsh\textunderscore 01\textunderscore 4\textunderscore csv\textasciitilde -box\textunderscore width\textasciitilde 40\textasciitilde -bit\textunderscore depth\textasciitilde 24\textasciitilde -partition\textasciitilde 4\textasciitilde 16\textasciitilde 1\textasciitilde -end}
2298   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.3\textasciitilde -reformat\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textunderscore transformed\.csv\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textunderscore transformed\textasciitilde 16x1\textunderscore csv\textasciitilde 1}
2299   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.2\textasciitilde -draw\textunderscore matrix;}
2300   \texttt{-input\textunderscore csv\textunderscore file\textasciitilde F16\textunderscore power\textunderscore 9\textunderscore trace\textunderscore transformed\textasciitilde 16x1\textunderscore csv\textasciitilde ;}
2301   \texttt{-box\textunderscore width\textasciitilde 40\textasciitilde -bit\textunderscore depth\textasciitilde 24\textasciitilde -partition\textasciitilde 4\textasciitilde 16\textasciitilde 1\textasciitilde -end}
2302   \texttt{2303}
2303
2304   \texttt{F_{16}.over.F_{4}.field\_reduction;}
2305   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.2;}
2306   \texttt{-define F\textasciitilde finite_field\textasciitilde q16\textasciitilde -end;}
2307   \texttt{-loop L.0.16.1;}
2308   \texttt{-with F\textasciitilde do;}
2309   \texttt{-finite_field\_activity;}
2310   \texttt{-field\_reduction\textasciitilde "F16\textunderscore over\textunderscore F4\textasciitilde \%L\textasciitilde .4\textasciitilde 1.1\textasciitilde \%L\textasciitilde ."\textasciitilde -end;}
2311   \texttt{-end\_loop;}
2312   \texttt{-loop L.0.16.1;}
2313   \texttt{-draw\textunderscore matrix\textasciitilde -input\textunderscore csv\textunderscore file\textasciitilde F16\textunderscore over\textunderscore F4\textasciitilde \%L\textasciitilde .csv\textasciitilde ;}
2314   \texttt{-box\textunderscore width\textasciitilde 40\textasciitilde -bit\textunderscore depth\textasciitilde 24\textasciitilde -partition\textasciitilde 4\textasciitilde 2.2\textasciitilde -end;}
2315   \texttt{-end\_loop}
2316
2317
2318
2319
2320   \#the\_polynomial\_31\textasciitilde is\_not\_primitive;
2321
2322   \texttt{F_{16}.poly31;}
2323   \texttt{(ORBITER\_PATH)orbiter.out\textasciitilde v.3;}
2324   \texttt{-define F\textasciitilde finite_field\textasciitilde q16\textasciitilde -override\_polynomial\_31\textasciitilde -end;}
2325   \texttt{-with F\textasciitilde do\textasciitilde finite_field\_activity;}
2326   \texttt{-cheat\_sheet\_GF\textasciitilde -end}
2327

429
2328
2329  F.25:
2330  ▶ $(ORBITER_PATH)orbiter.out-v.3:\
2331  ▶ ▶ -define F-\finite_field-q25-end\n2332  ▶ ▶ -with F-do-\finite_field\activity\n2333  ▶ ▶ -cheat_sheet_GF-end
2334  ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n2335  ▶ ▶ -input_csv_file.EOF_q25\addition_table.csv\n2336  ▶ ▶ -box_width40-bit_depth24--partition3-25-25-end
2337  ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n2338  ▶ ▶ -input_csv_file.EOF_q25\multiplication_table.csv\n2339  ▶ ▶ -box_width40-bit_depth24--partition3-24-24-end
2340  ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n2341  ▶ ▶ -input_csv_file.EOF_q25\addition_table.reordered.csv\n2342  ▶ ▶ -box_width40-bit_depth24--partition3-25-25-end
2343  ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n2344  ▶ ▶ -input_csv_file.EOF_q25\multiplication_table.reordered.csv\n2345  ▶ ▶ -box_width40-bit_depth24--partition3-24-24-end
2346
2347  trace_25:
2348  ▶ $(ORBITER_PATH)orbiter.out-v.3:\n2349  ▶ ▶ -define F-\finite_field-q25-end\n2350  ▶ ▶ -with F-do-\finite_field\activity-trace-end
2351  ▶ $(ORBITER_PATH)orbiter.out-v.3-reformat_F_q25_trace.csv-F_q25_trace_5x5.csv-5
2352  ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n2353  ▶ ▶ -input_csv_file.EOF_q25\addition_table.csv\n2354  ▶ ▶ -box_width40-bit_depth24--partition3-25-25-end
2355
2356
2357
2358  F.32:
2359  ▶ $(ORBITER_PATH)orbiter.out-v.3:\n2360  ▶ ▶ -define F-\finite_field-q32-end\n2361  ▶ ▶ -with F-do-\finite_field\activity\n2362  ▶ ▶ -cheat_sheet_GF-end
2363  ▶ pdflatex GF.32.tex
2364  ▶ open GF.32.pdf
2365
2366
2367
2368  F.49:
2369  ▶ $(ORBITER_PATH)orbiter.out-v.3:\n2370  ▶ ▶ -define F-\finite_field-q49-end\n2371  ▶ ▶ -with F-do-\finite_field\activity-cheat_sheet_GF-end
2372  ▶ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\n2373  ▶ ▶ -input_csv_file.EOF_q49\addition_table.csv\n
430
```bash
2374 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2376 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2377 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2378 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2379 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2380 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2381 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2382 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2383 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2384 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2385 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2386 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2387 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2388 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2389 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2390 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2391 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2392 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2393 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2394 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2395 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2396 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2397 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2398 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2399 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2400 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2401 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2402 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2403 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2404 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2405 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2406 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2407 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2408 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2409 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2410 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2411 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2412 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2413 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2414 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2415 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2416 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2417 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2418 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
2419 $ $(ORBITER_PATH) orbiter.out --v.2 --draw_matrix \
```

431
2420 \> \> -define F\-finite_field\-q64\-end\-
2421 \> \> -with F\-do\-finite_field_activity\-trace\-end
2422 \> \> $(ORBITER_PATH)orbiter.out\-v\-3\-reformat F_q64_trace.csv\-F_q64_trace_8x8.csv 8
2423 \> \> $(ORBITER_PATH)orbiter.out\-v\-2\-draw_matrix\-
2424 \> \> -input_csv_file F_q64_trace_8x8.csv\-
2425 \> \> -box_width 40\-bit_depth 24\-partition 3\-4\"1,1,1,1,1,1,1,1,1,1\"\"1,1,1,1,1,1,1,1,1,-end
2426 \> \> $(ORBITER_PATH)orbiter.out\-v\-3\-
2427 \> \> -define F\-finite_field\-q2\-end\-
2428 \> \> -with F\-do\-finite_field_activity\-
2429 \> \> -algebraic_normal_form F_q64_trace.csv 6\-end
2430
2431
trace_64_poly_2:
2432 \> \> $(ORBITER_PATH)orbiter.out\-v\-3\-
2433 \> \> -define F\-finite_field\-q64\-override_polynomial 115\-end\-
2434 \> \> -with F\-do\-finite_field_activity\-trace\-end
2435 \> \> $(ORBITER_PATH)orbiter.out\-v\-3\-reformat F_q64_trace.csv\-F_q64_trace_8x8.csv 8
2436 \> \> $(ORBITER_PATH)orbiter.out\-v\-2\-draw_matrix\-
2437 \> \> -input_csv_file F_q64_trace_8x8.csv\-
2438 \> \> -box_width 40\-bit_depth 24\-partition 3\-8\-8\-end
2439
2440
2441
F_64_over_F_8_field_reduction:
2442 \> \> $(ORBITER_PATH)orbiter.out\-v\-2\-
2443 \> \> -define F\-finite_field\-q64\-end\-
2444 \> \> -with F\-do\-
2445 \> \> -finite_field_activity\-field_reduction\"F64_over_F8\" 8\-8\-8\-
2446 \> \> \> "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,
2447 \> \> \> 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,
2448 \> \> \> 53,54,55,56,57,58,59,60,61,62,63\"\-end
2449 \> \> $(ORBITER_PATH)orbiter.out\-v\-2\-draw_matrix\-
2450 \> \> -input_csv_file F64_over_F8.csv\-
2451 \> \> -box_width 40\-bit_depth 24\-partition 4\"2,2,2,2,2,2\"\"2,2,2,2,2,2\"\-end
2452 \> \> pdf2latex\field_reduction_Q64_q8_8_8.tex
2453 \> \> open\field_reduction_Q64_q8_8_8.pdf
2454
2455 F_64_over_F_4_field_reduction:
2456 \> \> $(ORBITER_PATH)orbiter.out\-v\-2\-
2457 \> \> -define F\-finite_field\-q64\-end\-
2458 \> \> -with F\-do\-
2459 \> \> -finite_field_activity\-
2460 \> \> \> -field_reduction\"F64_over_F4\" 4\-8\-8\"0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,1
5, 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"-end
2461 ▷ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
2462 ▷ ▷ -input_csv_file:F64_over_F4.csv-
2463 ▷ ▷ -box_width:40-bit_depth:24-
2464 ▷ ▷ -partition:4:"3,3,3,3,3,3,3,3,3,3"-end
2465 ▷ pdflatex-field_reduction_Q64_q4_8_8.tex
2466 ▷ open-field_reduction_Q64_q4_8_8.pdf
2467
2468
2469 F64_over_F2_field_reduction:
2470 ▷ $(ORBITER_PATH)orbiter.out-v.2-
2471 ▷ ▷ -define:F:finite_field-q:64-end-
2472 ▷ ▷ -with:F:-do-
2473 ▷ ▷ -finite_field_activity-
2474 ▷ ▷ ▷ -field_reduction:"F64_over_F2":2:8:8:"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"-end
2475 ▷ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
2476 ▷ ▷ -input_csv_file:F64_over_F2.csv-
2477 ▷ ▷ -box_width:40-bit_depth:24-
2478 ▷ ▷ -partition:4:"6,6,6,6,6,6,6,6,6,6,6,6"-end
2479 ▷ pdflatex-field_reduction_Q64_q2_8_8.tex
2480 ▷ open-field_reduction_Q64_q2_8_8.pdf
2481
2482
2483
2484 trace_81:
2485 ▷ $(ORBITER_PATH)orbiter.out-v.3-
2486 ▷ ▷ -define:F:finite_field-q:81-end-
2487 ▷ ▷ -with:F:-do-finite_field_activity--trace-end
2488 ▷ $(ORBITER_PATH)orbiter.out-v.3-reformat_F_q81_trace.csv:F_q81_trace_9x9.csv:9
2489 ▷ $(ORBITER_PATH)orbiter.out-v.2-draw_matrix-
2490 ▷ ▷ -input_csv_file:F_q81_trace_9x9.csv-
2491 ▷ ▷ -box_width:40-bit_depth:24--partition:4:9:9--end
2492
2493
2494 F_125:
2495 ▷ $(ORBITER_PATH)orbiter.out-v.3-
2496 ▷ ▷ -define:F:finite_field-q:125-end-
2497 ▷ ▷ -with:F:-do-finite_field_activity--cheat_sheet_GF--end
2498 ▷ pdflatex-GF_125.tex
2499 ▷ open-GF_125.pdf
2500
2501
2502

433
F_256:
\begin{verbatim}
2503  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 3\\
2504  \$\$-define F\cdot finite\_field\textunderscore q\textunderscore 256\textunderscore -end\\
2505  \$\$-with F\cdot do\cdot finite\_field\_activity\textunderscore -cheat\_sheet\_GF\textunderscore -end\\
2506  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 2\textunderscore -draw\_matrix\\
2507  \$\$-input csv\textunderscore file\textunderscore GF\textunderscore q256\textunderscore addition\_table.csv\\
2508  \$\$-box\_width 40\textunderscore -bit\_depth 24\textunderscore -partition 3\textunderscore 256\textunderscore 256\textunderscore -end\\
2509  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 2\textunderscore -draw\_matrix\\
2510  \$\$-input csv\textunderscore file\textunderscore GF\textunderscore q256\textunderscore multiplication\_table.csv\\
2511  \$\$-box\_width 40\textunderscore -bit\_depth 24\textunderscore -partition 3\textunderscore 256\textunderscore 256\textunderscore -end\\
2512  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 2\textunderscore -draw\_matrix\\
2513  \$\$-input csv\textunderscore file\textunderscore GF\textunderscore q256\textunderscore addition\_table\_reordered.csv\\
2514  \$\$-box\_width 40\textunderscore -bit\_depth 24\textunderscore -partition 3\textunderscore 256\textunderscore 256\textunderscore -end\\
2515  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 2\textunderscore -draw\_matrix\\
2516  \$\$-input csv\textunderscore file\textunderscore GF\textunderscore q256\textunderscore multiplication\_table\_reordered.csv\\
2517  \$\$-box\_width 40\textunderscore -bit\_depth 24\textunderscore -partition 3\textunderscore 255\textunderscore 255\textunderscore -end\\
2518  \$\$pdf\_latex\_GF\textunderscore 256\textunderscore .tex\\
2519  \$\$open\_GF\textunderscore 256\textunderscore .pdf\\
2520\end{verbatim}

trace_256:
\begin{verbatim}
2523  trace_256:\\
2524  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 3\\
2525  \$\$-define F\cdot finite\_field\textunderscore q\textunderscore 256\textunderscore -end\\
2526  \$\$-with F\cdot do\cdot finite\_field\_activity\textunderscore -trace\textunderscore -end\\
2527  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 3\textunderscore -reformat F\_q256\textunderscore trace\_csv\_F\_q256\textunderscore trace\_16x16.csv\textunderscore 16\\
2528  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 2\textunderscore -draw\_matrix\\
2529  \$\$-input csv\textunderscore file\textunderscore F\_q256\textunderscore trace\_16x16.csv\\
2530  \$\$-box\_width 40\textunderscore -bit\_depth 24\\
2531  \$\$-partition 4\textquoteleft 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\textquoteright\\
2532  \$\$-v\textquoteleft 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\textquoteright\textunderscore -end\\
2533\end{verbatim}

F_289:
\begin{verbatim}
2536  F_289:\\
2537  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 3\\
2538  \$\$-define F\cdot finite\_field\textunderscore q\textunderscore 289\textunderscore -end\\
2539  \$\$-with F\cdot do\cdot finite\_field\_activity\textunderscore -cheat\_sheet\_GF\textunderscore -end\\
2540  \$\$pdf\_latex\_GF\textunderscore 289\textunderscore .tex\\
2541  \$\$open\_GF\textunderscore 289\textunderscore .pdf\\
2542\end{verbatim}

normal\_basis\_2\_3:
\begin{verbatim}
2545  normal\_basis\_2\_3:\\
2546  \$\$(ORBITER\_PATH)orbi\textunderscore out\textunderscore -v\textunderscore 2\\
2547  \$\$-define F\cdot finite\_field\textunderscore q\textunderscore 2\textunderscore -end\\
2548  \$\$-with F\cdot do\cdot finite\_field\_activity\textunderscore .\end{verbatim}
normal_basis_2.6:

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

-define F_=finite_field-q.2-end\\

-with F_=do_-finite_field_activity\\

-normal_basis_6-end

F_512:

$($(ORBITER_PATH)orbiter.out-v.3\$

-define F_=finite_field-q.512-end

#-User-time:2/100-seconds

F_1024:

$($(ORBITER_PATH)orbiter.out-v.3\$

-define F_=finite_field-q.1024-end

#-User-time:10/100-seconds

F_2048:

$($(ORBITER_PATH)orbiter.out-v.3\$

-define F_=finite_field-q.2048-end

F_4096:

$($(ORBITER_PATH)orbiter.out-v.3\$

-define F_=finite_field-q.4096-end

#-User-time:0:2

F_8192:

$($(ORBITER_PATH)orbiter.out-v.3\$

-define F_=finite_field-q.8192-end

#-User-time:0:8

F_16384:

$($(ORBITER_PATH)orbiter.out-v.3\$

-define F_=finite_field-q.16384-end

#-User-time:0:37
Section 3.5: Linear Algebra over Finite Fields
SECTION_LINEAR_ALGEBRA:

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

nullspace:
```bash
$ (ORBITER_PATH) orbiter.out -v:2
```
```bash
-define F2_finite_field -q:2 -end
-define v-vector-field F2-format:2
-dense:1,1,1,1,0,1,1,0,0,1
-end
-with F2-do-
-finite_field_activity
-nullspace v
-normalize_from_the_right
-end
```

eigenstuff:
```bash
$ (ORBITER_PATH) orbiter.out -v:6
```
```bash
-define F_finite_field -q:5 -end
-eigenstuff F:4:0,1,0,2,0,1,2,1,4,2,3,1,2,0,4,3
```

classes_GL_3_2:
```bash
$ (ORBITER_PATH) orbiter.out -v:7
```
```bash
-define F_finite_field -q:2 -end
-all_rational_normal_forms F:3
```

```bash
# pdflatex Class_reps_GL_3_2.tex
```
```bash
# open Class_reps_GL_3_2.pdf
```

classes_GL_4_2:
```bash
$ (ORBITER_PATH) orbiter.out -v:7
```
```bash
-define F_finite_field -q:2 -end
-all_rational_normal_forms F:4
```
RREF_demo_4_4_q5:
$\$(ORBITER\_PATH)\text{orbiter.out-\v.2}\$
$\$-\text{define}\_F-\text{finite}\_field-\text{q.5-\text{-end}}$$
$\$-\text{with}\_F-\text{-do}$$
$\$-\text{finite}\_field\_activity-\text{RREF}\_demo.4.4-\text{-end}$$
$\$\text{pdflatex-RREF}\_example.q5.4.4.txt$$
$\$\#\text{open-RREF}\_example.q5.4.4.pdf$$
$\$\text{gs-\sDEVICE=png16-\text{-dFIXEDMEDIA-\text{-dDEVICEWIDTHPOINTS=500-\text{-dDEVICEHEIGHTPOINTS=450}}}$
$\$\text{-r240-\text{-oRREF}\_example.q5.4.4.page\%02d.png}$
$\$\text{RREF}\_example.q5.4.4.pdf$$
$\$\#$\text{-dDEVICEWIDTHPOINTS=100-\text{-dDEVICEHEIGHTPOINTS=100}}$$
$\$\#$\text{-sPAPERSIZE=\text{hagaki}}$$
$\$\text{RREF}\_demo.4.6.q7:
$\$\$(ORBITER\_PATH)\text{orbiter.out-\v.2}\$
$\$-\text{define}\_F-\text{finite}\_field-\text{q.7-\text{-end}}$$
$\$-\text{with}\_F-\text{-do}$$
$\$-\text{finite}\_field\_activity-\text{RREF}\_\text{random}\_matrix.4.6-\text{-end}$
$\$\text{pdflatex-RREF}\_example.q7.4.6.txt$$
$\$\text{gs-\sDEVICE=png16-\text{-dFIXEDMEDIA-\text{-dDEVICEWIDTHPOINTS=500-\text{-dDEVICEHEIGHTPOINTS=450}}}$
$\$\text{-r240-\text{-oRREF}\_example.q7.4.6.page\%02d.png}$
$\$\text{RREF}\_example.q7.4.6.pdf$$
$\$\text{open-RREF}\_example.q7.4.6.pdf$$
$\$\text{RREF}\_demo.4.8.q8:
$\$\$(ORBITER\_PATH)\text{orbiter.out-\v.2}\$
$\$-\text{define}\_F-\text{finite}\_field-\text{q.8-\text{-end}}$$
$\$-\text{with}\_F-\text{-do}$$
$\$-\text{finite}\_field\_activity-\text{RREF}\_\text{\text{-random}\_matrix.4.8-\text{-end}}$
$\$\text{pdflatex-RREF}\_example.q8.4.8.txt$$
$\$\#\text{open-RREF}\_example.q8.4.8.pdf$$
$\$\text{gs-\sDEVICE=png16-\text{-dFIXEDMEDIA-\text{-dDEVICEWIDTHPOINTS=500-\text{-dDEVICEHEIGHTPOINTS=450}}}$
Section 4.1: Finite-Projective Spaces

PG_1_16:

- $(\text{ORBITER\_PATH})\text{orbiter.out}\$
- $-\text{define}\ F-\text{finite}\_\text{field}\-q\text{-end}\$
- $-\text{define}\ v-\text{vector}\-f\text{ield}\ F-\text{format}\-4\$
- $-\text{dense}\$(\text{CODE\_RS\_6\_4\_7})\$
- $-\text{end}\$
- $-\text{with}\ F-\text{do}\$
- $-\text{finite}\_\text{field}\_\text{activity}\-\text{RREF}\ v-\text{end}\$
- pdf\text{latex}\text{RREF}\_\text{example}\_q7\_4\_6\_.\text{tex}\$
- $-\text{gs}\-s\text{DEVICE=png16-}\text{dFIXEDMEDIA-}\text{dDEVICEWIDTHPOINTS=500-}\text{dDEVICEHEIGHTPOINTS=450}\$
- $-\text{pdflatex}\text{RREF}\_\text{example}\_q7\_4\_6\_.\text{pdf}\$

PG_2_2:
令 $F$ 为有限域，$	ext{q} = 2$ 的有限域，

```
\text{define } F \text{- finite field - q 2 - end}
```

令 $P$ 为 $F$ 上的射影空间，$	ext{q} = 2$ 的射影空间，

```
\text{define } P \text{- projective space - 2 F - end}
```

令 $F$ 为有限域，$	ext{q} = 4$ 的有限域，

```
\text{define } F \text{- finite field - q 4 - end}
```

令 $P$ 为 $F$ 上的射影空间，$	ext{q} = 4$ 的射影空间，

```
\text{define } P \text{- projective space - 2 F - end}
```

令 $F$ 为有限域，$	ext{q} = 8$ 的有限域，

```
\text{define } F \text{- finite field - q 8 - end}
```

令 $P$ 为 $F$ 上的射影空间，$	ext{q} = 8$ 的射影空间，

```
\text{define } P \text{- projective space - 2 F - end}
```

令 $F$ 为有限域，$	ext{q} = 13$ 的有限域，

```
\text{define } F \text{- finite field - q 13 - end}
```

令 $P$ 为 $F$ 上的射影空间，$	ext{q} = 13$ 的射影空间，

```
\text{define } P \text{- projective space - 2 F - end}
```
2823 \triangleright \triangleright -end
2824 \triangleright pdflatex PG_2_13.tex
2825 \triangleright open PG_2_13.pdf
2826
2827
2828
2829
2830 PG_2_64:
2831 \triangleright $(ORBITER_PATH)orbiter.out\
2832 \triangleright \triangleright -define F\ finite\ field\ -q\ 64\ -end\\
2833 \triangleright \triangleright -define P\ projective\ space\ -2\ F\ -end\\
2834 \triangleright \triangleright -with P\ do\ projective\ space\ activity\\
2835 \triangleright \triangleright \triangleright -cheat_sheet\
2836 \triangleright \triangleright -end
2837 \triangleright pdflatex PG_2_64.tex
2838 \triangleright open PG_2_64.pdf
2839
2840
2841
2842 PG_3_2:
2843 \triangleright $(ORBITER_PATH)orbiter.out\ -v\ 0\\
2844 \triangleright \triangleright -define F\ finite\ field\ -q\ 2\ -end\\
2845 \triangleright \triangleright -define P\ projective\ space\ -3\ F\ -end\\
2846 \triangleright \triangleright -with P\ do\ projective\ space\ activity\\
2847 \triangleright \triangleright \triangleright -cheat_sheet\
2848 \triangleright \triangleright -end
2849 \triangleright pdflatex PG_3_2.tex
2850 \triangleright open PG_3_2.pdf
2851
2852
2853 PG_3_4:
2854 \triangleright $(ORBITER_PATH)orbiter.out\ -v\ 10\\
2855 \triangleright \triangleright -define F\ finite\ field\ -q\ 4\ -end\\
2856 \triangleright \triangleright -define P\ projective\ space\ -3\ F\ -end\\
2857 \triangleright \triangleright -with P\ do\ projective\ space\ activity\\
2858 \triangleright \triangleright \triangleright -cheat_sheet\
2859 \triangleright \triangleright -end
2860 \triangleright pdflatex PG_3_4.tex
2861 \triangleright open PG_3_4.pdf
2862
2863 PG_3_5:
2864 \triangleright $(ORBITER_PATH)orbiter.out\
2865 \triangleright \triangleright -define F\ finite\ field\ -q\ 5\ -end\\
2866 \triangleright \triangleright -define P\ projective\ space\ -3\ F\ -end\\
2867 \triangleright \triangleright -with P\ do\ projective\ space\ activity\\
2868 \triangleright \triangleright \triangleright -cheat_sheet\
2869 \triangleright \triangleright -end

441
PG_3.7:
\$(ORBITER_PATH)orbiter.out:\
-define F: finite_field -q 7: end:\
-define P: projective_space 3 F: end:\
-with P: do -projective_space_activity:\
-\cheat_sheet:\
-end
-pdflatex PG_3.7.tex:
-open PG_3.7.pdf

PG_3.8:
\$(ORBITER_PATH)orbiter.out:\
-define F: finite_field -q 8: end:\
-define P: projective_space 3 F: end:\
-with P: do -projective_space_activity:\
-\cheat_sheet:\
-end
-pdflatex PG_3.8.tex:
-open PG_3.8.pdf

PG_3.16:
\$(ORBITER_PATH)orbiter.out:\
-define F: finite_field -q 16: end:\
-define P: projective_space 3 F: end:\
-with P: do -projective_space_activity:\
-\cheat_sheet:\
-end
-pdflatex PG_3.16.tex:
-open PG_3.16.pdf

PG_3.25:
\$(ORBITER_PATH)orbiter.out:\
-define F: finite_field -q 25: end:\
-define P: projective_space 3 F: end:\
-with P: do -projective_space_activity:\
-\cheat_sheet:\

### Section 4.2: Indexing Points

**SECTION_INDEXING_POINTS:**

- **PG_2.4_rank_point:**
  - \$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ -2\$
  - \$\text{define}\\ F\ -\text{finite_field}\ -q\ 4\ \text{-end}\$
  - \$\text{define}\\ P\ -\text{projective_space}\ 8\ F\ \text{-end}\$
  - \$\text{with}\\ P\ -\text{do}\ -\text{projective_space}\_\text{activity}\$
  - \$\text{-rank}\_\text{point}\_\text{in}\_\text{PG}\_2.3.3.1\ \text{-end}\$

**geometry_global::do_rank_point_in_PG::coeff:(3,3,1)::has_rank:20**
elliptic_curve_b1_c3_q11.txt:

```plaintext
$\text{ORBITER PATH}\text{orbiter.out}\cdot-v.2$

define F - finite_field - q.11 - end

define P - projective_space - 3 F - end

with P - do

projective_space_activity

define object EC

end

define object all_one

repeat 1 -7

draw matrix

define all_one - vector

input csv file PG_n1_q2 incidence_matrix.csv

box width 20 - bit_depth 8

partition 3

end

open PG_n1_q2 incidence_matrix draw.bmp

PG_2_2_incidence_matrix:

$\text{ORBITER PATH}\text{orbiter.out}\cdot-v.2$

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

$\text{ORBITER PATH}\text{orbiter.out}\cdot-v.2$

define all_one - vector

repeat 1 -7

draw matrix

input csv file PG_n2_q2 incidence_matrix.csv

box width 20 - bit_depth 8

partition 3

end

open PG_n2_q2 incidence_matrix draw.bmp

PG_2_4_incidence_matrix:

$\text{ORBITER PATH}\text{orbiter.out}\cdot-v.2$

define F - finite_field - q.4 - end

define P - projective_space - 2 F - end

with P - do

projective_space_activity

export point_line_incidence_matrix

end

$\text{ORBITER PATH}\text{orbiter.out}\cdot-v.2$

define all_one - vector

repeat 1 -21

draw matrix

input csv file PG_n2_q4 incidence_matrix.csv

box width 20 - bit_depth 8

partition 3
```

3010     all_one_all_one
3011     end
3012     open-PG_n2_q4_incidence_matrix_draw.bmp
3013
3014     # writes PG_n2_q4_incidence_matrix.csv
3015
3016
3017
3018     PG_2.8_incidence_matrix:
3019     $(ORBITER_PATH)orbiter.out-v.2\n3020     -define-F-finite_field-q.8-end\n3021     -define-P-projective_space-2-F-end\n3022     -with-P-do-projective_space_activity\n3023     -export_point_line_incidence_matrix\n3024     -end
3025     $(ORBITER_PATH)orbiter.out-v.2\n3026     -define-all_one-vector-repeat-1.73-end\n3027     -draw_matrix\n3028     -input_csv_file-PG_n2_q8_incidence_matrix.csv\n3029     -box_width:20-bit_depth:8\n3030     -partition-3\n3031     -all_one-all_one\n3032     -end
3033     open-PG_n2_q8_incidence_matrix_draw.bmp
3034
3035     PG_2.16_incidence_matrix:
3036     $(ORBITER_PATH)orbiter.out-v.2\n3037     -define-F-finite_field-q.16-end\n3038     -define-P-projective_space-2-F-end\n3039     -with-P-do-projective_space_activity\n3040     -export_point_line_incidence_matrix\n3041     -end
3042     $(ORBITER_PATH)orbiter.out-v.2\n3043     -define-all_one-vector-repeat-1.273-end\n3044     -draw_matrix\n3045     -input_csv_file-PG_n2_q16_incidence_matrix.csv\n3046     -box_width:20-bit_depth:8\n3047     -partition-3\n3048     -all_one-all_one\n3049     -end
3050     open-PG_n2_q16_incidence_matrix_draw.bmp
3051
3052
3053
3054
3055
3056
SECTION FINITE DESARGUESIAN PROJECTIVE PLANES:

PG_2_16:

PG_2_4_with_decomposition:

PG_2_4_incma_cyclic:

#PG_2_4_singer_incma_cyclic.csv
#PG_2_4_singer_incma_subgroup_index_3.csv
#PG_2_4_singer_incma_subgroup_index_7.csv

PG_2_4_incma_cyclic:
Section 4.4: The Grassmannian

SECTION_GRASSMANNIAN:

GR_3_2_2:

$(ORBITER_PATH)orbiter.out:

-define-F-finite_field-q2-end:

-with-F-do-finite_field_activity:
Section 4.5: Algebraic Sets

EC_11.txt:

conic_ideal_q13:

Hirschfeld_surface_q4.txt:
-define F -finite_field -q 4 -end \n-define P -projective_space -3 F -end \n-end \n
-define object H4 \n-projective_space_activity \n-end \n
-projective_variety "Hirschfeld_surface_q4" \n-3 \$(HIRSCHFELD_SURFACE_EQUATION) \n-monomial_type PART \n-end \n
-with H4 -do -combinatorial_object_activity -save \n-end \n
# creates Hirschfeld_surface_q4.txt

Hirschfeld_surface_q16.txt:

-define F -finite_field -q 16 -end \n-define P -projective_space -3 F -end \n-end \n
-with P -do \n-projective_space_activity \n-end \n
-projective_variety "Hirschfeld_surface_q16" \n-3 \$(HIRSCHFELD_SURFACE_EQUATION) \n-monomial_type PART \n-end \n
-with H16 -do -combinatorial_object_activity -save \n-end \n
# the coefficient vector is given as a list of pairs.
# 165 = binomial(11,3)

Endrass_F7.txt:

-define F -finite_field -q 7 -end \n-define eqn -vector -field F -sparse 165 \n-define eqn -vector -field F -sparse 165 \n-define (ENDRASS_SPARSE) -end \n-define P -projective_space -3 F -end \n-with P -do \n-projective_space_activity \n-end \n
-define object Endrass_F7 \n-projective_variety "Endrass_F7" -eqn \n-monomial_type LEX \n
449
3242  ▶  ▶  ▶  -end\n3243  ▶  ▶  -end\n3244  ▶  ▶  -with Endrass_F7-do-combinatorial_object_activity-save\n3245  ▶  ▶  -end
3246
3247
3248
3249  # we created a set of 33 points, called Endrass_F7.txt
3250
3251
3252
3253 octic_prepare:
3254 ▶  $(ORBITER_PATH)orbiter.out-v.4-define D\n3255 ▶  ▶  -diophant-label octic_monomials\n3256 ▶  ▶  -coefficient_matrix 1.4"1,1,1"-RHS"8,8,1"\n3257 ▶  ▶  -x_min global 0-x_max global 8\n3258 ▶  ▶  -end
3259 ▶  ▶  -with D-do\n3260 ▶  ▶  ▶  -diophant activity-solve mckay\n3261 ▶  ▶  ▶  -end
3262 ▶  ▶  sort -r octic_monomials.sol-octic_monomials_sorted.txt
3263
3264  #Found 165 solutions with 210 backtrack steps
3265  # 165=binomial(11,3)
3266
3267
3268
3269
3270  #########################################################################
3271  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " "  " 
3272  # Section 4.6: The Klein Quadric and Pluecker coordinates
3273
3274 SECTION_KLEIN_QUADRIC_AND_PLUECKERCOORDINATES:
3275
3276
3277
3278 GR_4_2_2:
3279 ▶  $(ORBITER_PATH)orbiter.out-v.2\n3280 ▶  ▶  -define F-finite_field-q.2-end\n3281 ▶  ▶  -with F-do-finite_field_activity\n3282 ▶  ▶  ▶  -cheat_sheet_Gr.4.2-end
3283 ▶  ▶  pdflatex Gr_4_2_2.tex.
3284 ▶  ▶  open Gr_4_2_2.pdf
3285
3286
3287  #########################################################################
# Section 4.7: Orthogonal spaces

SECTION ORTHOGONAL_SPACES:

Op 4.2:

```
$(ORBITER_PATH)orbiter.out -v:2\n-define:F:finite_field:-q:2:-end\n-define:0:orthogonal_space:1:4:F:-without_group:-end\n-with:0-do-orthogonal_space_activity\n-define:F:finite_field:-q:2:-end\n-define:O:orthogonal_space:0:5:F:-without_group:-end\n-with:0-do-orthogonal_space_activity\n-export_point_line_incidence_matrix\n-end
```

Op 5.2 incidence_matrix.csv:

```
$(ORBITER_PATH)orbiter.out -v:2\n-define:F:finite_field:-q:2:-end\n-define:0:orthogonal_space:0:5:F:-without_group:-end\n-with:0-do-orthogonal_space_activity\n-export_point_line_incidence_matrix\n-end
```

Op 6.2:

```
$(ORBITER_PATH)orbiter.out -v:2\n-define:F:finite_field:-q:2:-end\n-define:0:orthogonal_space:1:6:F:-without_group:-end\n-with:0-do-orthogonal_space_activity\n-export_point_line_incidence_matrix\n-end
```

Op 7.2:

```
$(ORBITER_PATH)orbiter.out -v:2\n-define:F:finite_field:-q:2:-end\n-define:0:orthogonal_space:1:7:F:-without_group:-end\n-with:0-do-orthogonal_space_activity\n-export_point_line_incidence_matrix\n-end
```
Op_6.2.incidence_matrix.csv:

\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n
\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \\n

#problem:  
#error-message:  
#stabilizer\_chain\_base\_data::allocate\_base\_data\_degree\_is\_too\_large  

Op_6.2:  

Op_8.2:  

Op_6.64:  

\$\{(ORBITER\_PATH)\}orbiter.out -v 2 \}
#problem, because we are trying to create PGL(6,64):

```
#define F-quintic_field-q64-end
#define O-orthogonal_space-1-6-F-without_group-end
with-0-do-orthogonal_space_activity

#define F-finite_field-q64-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
```

# use option -without_group to skip the group. This will work:

```
#define F-quintic_field-q64-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:

```
#define F-quintic_field-q8-end
#define O-orthogonal_space-1-6-F-without_group-end
with-0-do-orthogonal_space_activity
-orthogonal_activity
end
```

```
pdflatex 0_1_6_64_report.tex
open 0_1_6_64_report.pdf
```

```
op_6_8_2:
#define F-finite_field-q8-end
#define O-orthogonal_space-1-6-F-without_group-end
with-0-do-orthogonal_space_activity
-orthogonal_activity
end
```

453
\textbf{Section 4.8: Hermitian Varieties}

\textbf{SECTION HERMITIAN VARIETIES:}

\begin{verbatim}
H_2.4:
$(ORBITER_PATH)orbiter.out-v.2\$
\texttt{define}\ F\ \texttt{-finite_field}\ \texttt{-q}\ 4\ \texttt{-end}\$
\texttt{with}\ F\ \texttt{-do}\ \texttt{-finite_field}\ \texttt{-activity}\$
\texttt{-cheat}\ \texttt{sheet}\ \texttt{hermitian}\ 2\ \texttt{-end}
\texttt{pdflatex}\ H_2.4.tex$
\texttt{open}\ H_2.4.pdf

H_2.9:
$(ORBITER_PATH)orbiter.out-v.2\$
\texttt{define}\ F\ \texttt{-finite_field}\ \texttt{-q}\ 9\ \texttt{-end}\$
\texttt{with}\ F\ \texttt{-do}\ \texttt{-finite_field}\ \texttt{-activity}\$
\texttt{-cheat}\ \texttt{sheet}\ \texttt{hermitian}\ 2\ \texttt{-end}
\texttt{pdflatex}\ H_2.9.tex$
\texttt{open}\ H_2.9.pdf

H_3.4:
$(ORBITER_PATH)orbiter.out-v.2\$
\texttt{define}\ F\ \texttt{-finite_field}\ \texttt{-q}\ 4\ \texttt{-end}\$
\texttt{with}\ F\ \texttt{-do}\ \texttt{-finite_field}\ \texttt{-activity}\$
\texttt{-cheat}\ \texttt{sheet}\ \texttt{hermitian}\ 3\ \texttt{-end}
\texttt{pdflatex}\ H_3.4.tex$
\texttt{open}\ H_3.4.pdf

# H_3.4 = the Hirschfeld surface
\end{verbatim}

\#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.
SECTION PROJECTIVE SPACE ADVANCED TOPICS:

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

fix_structure_2B:
$(ORBITER_PATH)orbiter.out -v.2\ 
-definition-F-finite_field-q.4-end\ 
-definition-P-projective_space-3F-end\ 
-with-P-do\ 
-projective_space_activity\ 
-define-"1,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0")\ 
-fix_structure_2B.tex\ 
-end.\ 
-pdflatex-fix_structure_2B.tex\ 
-open-fix_structure_2B.pdf

fix_structure_2C:
$(ORBITER_PATH)orbiter.out -v.2\ 
-definition-F-finite_field-q.4-end\ 
-definition-P-projective_space-3F-end\ 
-with-P-do\ 
-projective_space_activity\ 
-define-"1,0,0,0,1,1,0,0,0,0,1,0,1,0,1,0")\ 
-fix_structure_2C.tex\ 
-end.
del_Pezzo_F13a_points.txt:

```plaintext
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
```

del_Pezzo_F13a_points.txt:

```plaintext
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
\$($\text{ORBITER\_PATH}$)orbiter.out -v 3
```
\[ x^*x*x+y*y*y+z*z*z+x*x*y*y \]

- 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_F13a_report.tex
pdflatex -del_Pezzo_F13b_report.tex
open -del_Pezzo_F13a_report.pdf
open -del_Pezzo_F13b_report.pdf
#dot -Tpng -del_Pezzo_F13a.gv > -del_Pezzo_F13a.png
#open -del_Pezzo_F13a.png

writes -del_Pezzo_F13a_points.txt

```
457
```
-projective_space_activity\n(define_object:E:\n(set:$(EDGE_CURVE_Q23_AS_POINTS):\nend:\nwith:E-do:\n-combinatorial_object_activity:\nsave:\nend:\nwith:E-do:\n-combinatorial_object_activity:\n-conic_type:6:\n-print_symbols

#############################################################################
#############################
# Chapter 5.- Group Theory
#############################

SECTION_PERMUTATION_GROUPS:

C13:
$(ORBITER_PATH)orbiter.out:-v.10:\n(define-gens:-vector:-dense:$(GEN_C13):-end:\n(define-G:-permutation_group:\n bsgs-C13-C_{13}:-13:13:0.1:\ngens:\nend:\nwith:G-do:\ngroup_theoretic_activity:\n-export_orbiter:\nend:\nwith:G-do:\ngroup_theoretic_activity:\
Symmetric_4:
Symmetric_4_stab:

# ToDo:

Gr = modified group from G:

restricted_action="1,2,3";
PGL\_4\_2\_export:

\>$\$(\text{ORBITER\_PATH})\text{orbiter.out}\text{-v}\_2\$
\>$\-\text{define}\_F\-\text{finite}\_\text{field}\-q\_2\-\text{-end}\$
\>$\-\text{define}\_G\-\text{linear}\_\text{group}\-\text{PGL}\_4\_F\-\text{-end}\$
\>$\-\text{with}\_G\-\text{-do}\$
\>$\-\text{group theoretic activity}\$
\>$\-\text{-report}\$
\>$\-\text{-end}\$
\>$\-\text{-with}\_G\-\text{-do}\$
\>$\-\text{group theoretic activity}\$
\>$\-\text{-export orbiter}\$
\>$\-\text{-end}\$

\>$\text{pdflatex-PGL}\_4\_2\_report.tex$
\>$\text{open-PGL}\_4\_2\_report.pdf$

C13\_as\_subgroup:

\>$\$(\text{ORBITER\_PATH})\text{orbiter.out}\text{-v}\_10\$
\>$\-\text{define}\_G\-\text{permutation group}\-\text{symmetric group}\_13\$
\>$\-\text{subgroup by generators C13\_13\_1}\$\text{GEN}_C13\$\-\text{-end}\$
\>$\-\text{-with}\_G\-\text{-do}\$
\>$\-\text{group theoretic activity}\$
\>$\-\text{-export orbiter}\$
\>$\-\text{-end}\$
\>$\-\text{-with}\_G\-\text{-do}\$
\>$\-\text{group theoretic activity}\$
\>$\-\text{-report}\$
\>$\-\text{-end}\$
\>$\-\text{-with}\_G\-\text{-do}\$
\>$\-\text{group theoretic activity}\$
\>$\-\text{-save elements csv }\text{"C13\_elts.csv}\$
\>$\-\text{-end}\$
\>$\#\text{pdflatex-Perm13\_Subgroup\_C13\_13\_report.tex}$
\>$\#\text{open-Perm13\_Subgroup\_C13\_13\_report.pdf}$

#pdflatex

#open
# Section 5.2: Linear Groups

## PGL_3_2:
```latex
dollar (ORBITER\_PATH)orbiter.out-v.2\$
d-define F-finite_field-q.2-end
-define G-linear_group-PGL.3-F-end
-with G-do
-group_theoretic_activity
-report
-end
```
```
pdflatex PGL_3_2_report.tex
open-PGL_3_2_report.pdf
```

## PGL_4_2:
```latex
dollar (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
```
```
pdflatex PGL_4_2_report.tex
open-PGL_4_2_report.pdf
```

## AGL_1_27:
```latex
dollar (ORBITER\_PATH)orbiter.out-v.2\$
-define F-finite_field-q.27-end
-define G-linear_group-AGL.1-F-end
-with G-do
-group_theoretic_activity
-report
-end
```
```
pdflatex AGL_1_27_report.tex
open-AGL_1_27_report.pdf
```

## Group table
```latex
- report
-end
```
```
pdflatex AGL_1_27_report.tex
open-AGL_1_27_report.pdf
```

## PGL_4_5:

461
$\text{ORBITER}\text{PATH}/\text{inter.out} -v.2$

```
define F finite_field q 5 end
define G linear_group PGL 4 F end
with G do
  group_theoretic_activity
  report
end
pdflatex PGL 4 5 report.tex
open PGL 4 5 report.pdf
```

```
PGL 4 2 wd:
define G linear_group PGL 4 2 wedge detached end
with G do
  group_theoretic_activity
  report
end
pdflatex PGL 4 2 Wedge 4 0 detached report.tex
open PGL 4 2 Wedge 4 0 detached report.pdf
```

```
PGL 4 2 wd_reverse:
define G linear_group PGL 4 2 wedge detached end
with G do
  group_theoretic_activity
  reverse_isomorphism_exterior_square
end
```

```
PGGL 3 4:
define G linear_group PGGL 3 4 end
with G do
  group_theoretic_activity
  report
  sylow
  classes
end
pdflatex PGGL 3 4 report.tex
open PGGL 3 4 report.pdf
```
# -sylow:

```bash
  #> > -export_schreier_trees
  #> > -tools_path:$(GRAPH_THEORY_PATH)
  #> > -report
  #> > -report_schreier_trees
```

PGGL_3.8:

```bash
  $(ORBITER_PATH)orbiter.out-v.5
  -define-G-linear_group-PGGL_3.8-end
```

PGGL_3.8_report:

```bash
  $(ORBITER_PATH)orbiter.out-v.3
  -define-G-linear_group-PGGL_3.8-end
  -with-G-do
  -group_theoretic_activity
  -report
  -end
```

```bash
  pdflatex:PGGL_3.8_report.tex
  open:PGGL_3.8_report.pdf
```

```bash
  #> > -draw_options:-radius 180-scale 0.30-line_width 0.75-end
```

PGO_5.2:

```bash
  $(ORBITER_PATH)orbiter.out-v.2
  -define-F-finite_field-q 2-end
  -define-G-linear_group-PGO_5.2-end
  -with-G-do
  -group_theoretic_activity
  -report
  -end
```

```bash
  pdflatex:PGO_5.2_report.tex
  open:PGO_5.2_report.pdf
```

PGGO_5.4:

```bash
  $(ORBITER_PATH)orbiter.out-v.2
  -define-F4-finite_field-q 4-end
  -define-G-linear_group-PGGO_5.4-end
  -with-G-do
  -group_theoretic_activity
  -report
  -end
```

```bash
  pdflatex:PGGO_5.4_report.tex
  open:PGGO_5.4_report.pdf
```

```bash
  pdflatex:PGG_5.2_report.tex
  open:PGG_5.2_report.pdf
```
PGOp\_6.2:

\$\text{(ORBITER\_PATH)orbiter.out}\ -v\ -2;\$

-define\(F\)\ -finite\_field\ -q\ 2\ -end;\$

-define\(G\)\ -linear\_group\ -PGOp\ 6\\_F\ -end;\$

-with\(G\)\ -do;\$

-group\_theoretic\_activity;\$

-report;\$

-end

pdflatex\ \PGOp\ 6\ 2\ \\report;tex

open\ \PGOp\ 6\ 2\ \\report.pdf

PGOm\_6.2:

\$\text{(ORBITER\_PATH)orbiter.out}\ -v\ -2;\$

-define\(F\)\ -finite\_field\ -q\ 2\ -end;\$

-define\(G\)\ -linear\_group\ -PGOm\ 6\\_F\ -end;\$

-with\(G\)\ -do;\$

-group\_theoretic\_activity;\$

-report;\$

-end

pdflatex\ \PGOm\ 6\ 2\ \\report;tex

open\ \PGOm\ 6\ 2\ \\report.pdf

PSP\_6.2:

\$\text{(ORBITER\_PATH)orbiter.out}\ -v\ -2;\$

-define\(F\)\ -finite\_field\ -q\ 2\ -end;\$

-define\(G\)\ -linear\_group\ -PGL\ 6\\_F\ -end;\$

-symplectic\_group;\$

-end;\$

-with\(G\)\ -do;\$

-group\_theoretic\_activity;\$

-report;\$

-end

pdflatex\ \PGL\ 6\ 2\ Sp\ 6\ 2\ \\report;tex

open\ \PGL\ 6\ 2\ Sp\ 6\ 2\ \\report.pdf

PGO\_7.2:

\$\text{(ORBITER\_PATH)orbiter.out}\ -v\ -2;\$

-define\(F\)\ -finite\_field\ -q\ 2\ -end;\$

-define\(G\)\ -linear\_group\ -PGO\ 7\\_F\ -end;\$

-with\(G\)\ -do;\$

-group\_theoretic\_activity;\$

-report;\$

-end

pdflatex\ \PGO\ 7\ 2\ \\report;tex

open\ \PGO\ 7\ 2\ \\report.pdf
SECTION 5.3: Subgroups

J1: $(\text{ORBITER PATH})\text{orbiter.out}$ -v 3

-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

PGL 3 11 singer:

$\text{ORBITER_PATH}\text{orbiter.out}$ -v 3

-define G -linear_group -PGL 3 11 -singer 19 -end

-with G -do

-group_theoretic_activity

-report

-end

pdflatex PGL_3_11_Singer_3_11_19_report.tex

open PGL_3_11_Singer_3_11_19_report.pdf

PGL 3 11 singer and frobenius:

$\text{ORBITER_PATH}\text{orbiter.out}$ -v 3

-define G -linear_group -PGL 3 11 -singer_and_frobenius 19 -end

-with G -do

-group_theoretic_activity

-report

-end

pdflatex PGL_3_11_Singer_and_Frob3_11_19_report.tex

open PGL_3_11_Singer_and_Frob3_11_19_report.pdf

PG 2 4 order 21:
$\text{orbiter.out}$

-define $G$-linear_group=PGL.3:4-end
-with $G$-do
-group_theoretic_activity-
-search_element_of_order 21
-end

quaternion:

-define $G$-linear_group=SL.2:3
-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

dfla$tex$-GL.2.3_Subgroup_quaternion.8_elements.tex
open-GL.2.3_Subgroup_quaternion.8_elements.pdf
pdf$f$-GL.2.3_Subgroup_quaternion.8_report.tex
open-GL.2.3_Subgroup_quaternion.8_report.pdf

cube_group:

-define gens-Vector-dense-
 "0,1,0,2,0,0,0,1,"
 0,0,1,0,1,0,2,0,0, 
 2,0,0,0,1,0,0,0,1"
-end-
-defin $G$-linear_group=GL.3:3
-subgroup_by Generators: "cube"."48".3
-gens-
-end
-with $G$-do
-group_theoretic_activity-
-print_elements.tex-
-report-
end

dfla$tex$-GL.3.3_Subgroup_cube.48_report.tex
open-GL.3.3_Subgroup_cube.48_report.pdf
pdf$f$-GL.3.3_Subgroup_cube.48_elements.tex
tetra_group:
$\$(ORBITER_PATH)orbiter.out\$\$\$v\$.3$\$
-define-Glinear_groupGL\$.3\$.3$\$
-subgroup_by_generators"tetra"."12".2$\$
"0,0,1,0,0,0,0,1,0,0,0,0,0,1,2,0,0,0,2,0"$\$
-\$end$\$
-define$\$:G\$linear_groupGL\$.3\$.3$\$
-with-G-do$\$
-group_theoretic_activity$\$
-print_elements.tex$\$
-report$\$
-end$\$
pdflatexGL\$3\$.3Subgroup\$tetra\$12\$report.tex\$
opendefinitions$\$
-tetra$\$
report.pdf$\$
end$\$

Hesse_group:
$\$(ORBITER_PATH)orbiter.out\$\$\$v\$.3$\$
-define-gens-vector-compact$\$
-define-G-linear_group-PGGL\$.3\$.4$\$
-subgroup_by_generators"Hesse"."432".7.gens$\$
-end$\$
-with-G-do$\$
-group_theoretic_activity$\$
-print_elements.tex$\$
-report$\$
-end$\$
pdflatex-PGGL\$3\$.3Subgroup\$Hesse\$432\$report.tex\$
opendefinitions$\$
-tetra$\$
report.pdf$\$
Hesse_group:
#Hesse-group:
$1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,1,2,0,0,0,2,0,$
$1,0,0,0,0,1,0,0,0,1,1,0,0,0,0,1,1,0,0,0,2,0,$
$1,0,0,0,0,2,2,0,0,2,0,0,0,0,0,1,0,$
$0,1,1,0,2,0,0,0,1,3,2,0,2,1,1,2,1,$
$1,1,0,0,0,0,3,0,3,2,2,0,1,2,3,3,1,$
\begin{verbatim}
Weyl_E8: $(ORBITER_PATH)orbiter.out -v.3 \-define-gens -vector -dense \-define "$(GENERATORS_WEYL_GROUP_E8)" \-end \-define G -linear_group -GL.8.3 \-subgroup_by_generators \-define "Weyl_E8"."696729600".2 \-end \-define G -linear_group -PGL.3.9 \-subgroup_by_generators "U_3.3"."6048".2 \-end \-with G -do \-group_theoretic_activity \-end \pdflatex GL.8.3_Subgroup_Weyl_E8_696729600_report.tex
\end{verbatim}


\textbf{Section 5.4: Linear Groups, Advanced Topics}

\textbf{SECTION_LINEAR_GROUPS_ADVANCED_TOPICS:}

\begin{verbatim}
U_3.3: $(ORBITER_PATH)orbiter.out -v.3 \-define F -finite_field -q 9 -override_polynomial "17" -end \-define G -linear_group -PGL.3.9 \-subgroup_by_generators "U_3.3"."6048".2 \-end \-with G -do \-group_theoretic_activity \-end \pdflatex PGL.3.9_Subgroup_U_3.3_6048_report.tex
\end{verbatim}
PGL_2.3:

\begin{verbatim}
> open-PGL_3.9_Subgroup_U_3.3_6048_report.pdf

> $(ORBITER\_PATH)orbiter.out\:-v.3:
> -define-G\:-linear_group\:-PGL.2.3\:-end:\
> -with-G\:-do:\
> -group_theoretic_activity:\
> -report:\
> -group_table:\
> -end

pdflatex PGL_2.3_group_table.order_24.tex

dflatex PGL_2.3_report.tex

open-PGL_2.3_group_table.order_24.pdf
open-PGL_2.3_report.pdf

PG_3.2_again:

\begin{verbatim}
> $(ORBITER\_PATH)orbiter.out:
> -define-F\:-finite_field\:-q.3\:-end:
> -finite_field_activity:\
> -cheat_sheet_PG.2\:-end
> pdflatex_PG_2.3.tex:
> open-PG_2.3.pdf
\end{verbatim}

#Co3-from-Conway-et-al.,.1985-(ATLAS)
#order=495766656000

#ToDo:

Co3:

\begin{verbatim}
> $(ORBITER\_PATH)orbiter.out\:-v.6:\
> -define-F\:-finite_field\:-q.2\:-end:\
> -define-gen1\:-vector\:-field\:-F\:-format\:22\:-compact\:$\\(CONWAY\_GEN1\)$\:-end:\
> -define-gen2\:-vector\:-field\:-F\:-format\:22\:-compact\:$\\(CONWAY\_GEN2\)$\:-end:\
> -define-G\:-linear_group\:-PGL.22.2:\
> -subgroup_by_generators\:"Co3".495766656000.2:\
> "gen1":
> "gen2":
> -end:
> -with-G\:-do:\
> -group_theoretic_activity:\
\end{verbatim}

#Co3

\begin{verbatim}
> $(ORBITER\_PATH)orbiter.out\:-v.6:\
> -define-F\:-finite_field\:-q.2\:-end:\
> -define-gen1\:-vector\:-field\:-F\:-format\:22\:-compact\:$\\(CONWAY\_GEN1\)$\:-end:\
> -define-gen2\:-vector\:-field\:-F\:-format\:22\:-compact\:$\\(CONWAY\_GEN2\)$\:-end:\
> -define-G\:-linear_group\:-PGL.22.2:\
> -subgroup_by_generators\:"Co3".495766656000.2:\
> "gen1":
> "gen2":
> -end:
> -with-G\:-do:\
> -group_theoretic_activity:\
\end{verbatim}

#ToDo:

Co3:
# Section 5.5: Induced Actions

## SECTION INDUCED ACTIONS:

T3 on tensors:

```plaintext
$\text{(ORBITER_PATH)orbiter.out}\ -v\ 4\$
```

```
define G

linear group GL_d q wreath Sym_n 2 2 3

on tensors end

with G do

group theoretic activity

report

end
```

```plaintext
pdflatex GL_2_2_wreath_Sym3_report.tex
```

```plaintext
open GL_2_2_wreath_Sym3_report.pdf
```
T3r1:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
define G
```

```bash
-linear_group -GL_d_q_wr_Sym_n:2:2:3
```

```bash
-on_rank_one_tensors -end
```

```bash
-with G -do
```

```bash
-group_theoretic_activity
```

```bash
-report
```

```bash
-end
```

```bash
pdflatex GL_2_2_wreath_Sym3_report.tex
```

```bash
open GL_2_2_wreath_Sym3_report.pdf
```

... (same structure as above) ...

T4r1:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
define G
```

```bash
-linear_group -GL_d_q_wr_Sym_n:2:2:4
```

```bash
-on_tensors -end
```

```bash
-with G -do
```

```bash
-group_theoretic_activity
```

```bash
-report
```

```bash
-end
```

```bash
pdflatex GL_2_2_wreath_Sym4_report.tex
```

```bash
open GL_2_2_wreath_Sym4_report.pdf
```

... (same structure as above) ...

PGGL_2_8_on_conic:

```bash
$(ORBITER_PATH)orbiter.out -v 4
```

```bash
define G
```

```bash
-linear_group -PGGL_2_8 -PGL2_on_conic -end
```

471
Section 5.6: Group-Theoretic Activities

### PGL\_3\_2_elements:

```bash
$ (ORBITER\_PATH)orbiter.out -v 5
```

```bash
define G - linear group - PGL_3_2 - end
```

```bash
with G - do
```

```bash
group_theoretic_activity
```

```bash
report
```

```bash
end
```

```bash
pdflatex PGGL_2_8_OnConic_2_8_report.tex
```

```bash
open PGGL_2_8_OnConic_2_8_report.pdf
```

### PGL\_3\_4_singer:

```bash
$ (ORBITER\_PATH)orbiter.out -v 5
```

```bash
define G - linear group - PGL_3_4 - end
```

```bash
with G - do
```

```bash
group_theoretic_activity
```

```bash
report
```

```bash
end
```

### GL\_2\_8_multiply:

```bash
$ (ORBITER\_PATH)orbiter.out -v 5
```

```bash
define G - linear group - GL_2_8 - end
```

```bash
with G - do
```

```bash
group_theoretic_activity
```

```bash
report
```

```bash
end
```

```bash
pdflatex GL_2_8_mult.tex
```
GL_2.7_multiply:

$\$(ORBITER_PATH)orbiter.out\$-v.5\$

\$define\$-G-linear_group-GL_2.7\$-end\$

\$with\$-G-do\$

\$group_theoretic_activity\$

\$multiply\$-"0,1,2,3","4,5,6,0"\$

\$end\$

pdflatex-GL_2.7_mult.tex

open-GL_2.7_mult.pdf

GL_2.7_inv:

$\$(ORBITER_PATH)orbiter.out\$-v.5\$

\$define\$-G-linear_group-GL_2.7\$-end\$

\$with\$-G-do\$

\$group_theoretic_activity\$

\$inverse\$-"0,1,2,3"\$

\$end\$

pdflatex-GL_2.7_inv.tex

open-GL_2.7_inv.pdf

GL_2.7_power:

$\$(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

PGL_3.2_classes:

$\$(ORBITER_PATH)orbiter.out\$-v.3\$

\$define\$-G-linear_group-PGL_3.2\$-end\$

\$with\$-G-do\$

\$group_theoretic_activity\$

\$classes_based_on_normal_form\$

\$end\$

pdflatex-PGL_3.2_classes_normal_form.tex

open-PGL_3.2_classes_normal_form.pdf

#pdflatex-PGL_3.2_classes_out.tex

#open-PGL_3.2_classes_out.pdf
4356 #> > -classes:
4357
4358
4359
4360
4361 normal_forms_PGL_4_4:
4362 > $(ORBITER_PATH)orbiter.out -v.7:
4363 > > -define-G-linear_group-PGGL.4.4-end:
4364 > > > -with-G-do:
4365 > > > -group_theoretic_activity:
4366 > > > > -classes_based_on_normal_form:
4367 > > > -end
4368 > pdflatex PGGL.4.4_classes_normal_form.tex
4369 > open-PGGL.4.4_classes_normal_form.pdf
4370 >
4371
4372
4373
4374 PGL.4.4_2A_rank:
4375 > $(ORBITER_PATH)orbiter.out -v.6:
4376 > > -define-G-linear_group-PGGL.4.4-end:
4377 > > > -with-G-do:
4378 > > > -group_theoretic_activity:
4379 > > > > -element_rank:"1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,1"
4380 > > > -end
4381 > >
4382 PGL.4.4_2A_unrank:
4383 > $(ORBITER_PATH)orbiter.out -v.6:
4384 > > -define-G-linear_group-PGGL.4.4-end:
4385 > > > -with-G-do:
4386 > > > -group_theoretic_activity:
4387 > > > > -element_unrank:"1"
4388 > > > -end
4389 > >
4390
4391
4392 #.ToDo:
4393
4394 cent_2A:
4395 > $(ORBITER_PATH)orbiter.out -v.6:
4396 > > -linear_group-PGGL.4.4:
4397 > > > -subgroup_by_generators-"centralizer_2A":"40320".10
4398 > > > > "1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,1,1"
4399 > > > > "1,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,1,1"
4400 > > > > "1,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,1,1,1"
4401 > > > > "1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1,0,0,0"
4402 > > > > "1,0,0,0,0,1,0,0,1,1,0,1,1,1,0,0,0,0"
PGL\(_{4,5}\)_rank:

```
$\text{(ORBITER PATH)}\text{orbiter.out-}\text{v-6}\
\text{-define G-linear group-PGL-4-5-end}\
\text{-with G-do}\
\text{-group theoretic activity}\
\text{-element rank:}0,0,0,1,\cdot,2,3,0,1,0,3,4,4,\cdot,0,1,2,1\
\text{-end}
```

PGL\(_{4,5}\)_unrank:

```
$\text{(ORBITER PATH)}\text{orbiter.out-}\text{v-6}\
\text{-define G-linear group-PGL-4-5-end}\
\text{-with G-do}\
\text{-group theoretic activity}\
\text{-element unrank:701459351}\
\text{-end}
```

#eigen 3A:
```
$\text{(ORBITER PATH)}\text{orbiter.out-}\text{v-6-}\text{-finite field activity-}q=5-\text{-eigenstuff-4-}0,0,1,2,0,1,2,1,0,3,4,4,\cdot,0,1,2,1-\text{-end}
```

#eigen 3B:
```
$\text{(ORBITER PATH)}\text{orbiter.out-}\text{v-6-}\text{-finite field activity-}q=5-\text{-eigenstuff-4-}0,0,0,1,2,3,0,1,0,3,4,4,0,1,2,1-\text{-end}
```

#element of order 31 in PGL(4,5):
```
\text{int-data[]=}\{1,0,0,0,0,1,0,\cdot,0,0,0,1,0,1,1,3,0\};
```

normal forms PGL\(_{4,5}\):
```
$\text{(ORBITER PATH)}\text{orbiter.out-}\text{v-7}\
\text{-define G-linear group-PGL-4-5-end}\
\text{-with G-do}\
\text{-group theoretic activity}\
\text{-classes based on normal form}\
\text{-end}
```
Section 5.7: Group-Theoretic Activities Based on Magma

SECTION_GROUP_THEORETIC_ACTIVITIES_BASED_ON_MAGMA:

PGGL_2.4_classes:

\>$\text{\textbackslash{}define-}G\text{\textbackslash{}}
\>$\text{\textbackslash{}-linear_group-}\text{PGGL_2.4\textbackslash{}}
\>$\text{\textbackslash{}-end\textbackslash{}do\textbackslash{}
\>$\text{\textbackslash{}-group_theoretic_activity\textbackslash{}}
\>$\text{\textbackslash{}-classes\textbackslash{}end\textbackslash{}

PGGL_2.4_cent_2A:

\>$\text{\textbackslash{}define-}G\text{\textbackslash{}}
\>$\text{\textbackslash{}-linear_group-}\text{PGGL_2.4\textbackslash{}end\textbackslash{}
\>$\text{\textbackslash{}-with-}G\text{do\textbackslash{}
\>$\text{\textbackslash{}-group_theoretic_activity\textbackslash{}}
\>$\text{\textbackslash{}-centralizer_of_element\textbackslash{}"2A\textbackslash{}"1,0,0,1,1\textbackslash{}
\>$\text{\textbackslash{}-report\textbackslash{}end\textbackslash{}

pdflatex PGGL_2.4_cent_2A out.tex

open PGGL_2.4_cent_2A out.pdf
4494 ▷ $(MAGMA\ PATH)\magma\cdot\text{element\_2A\_centralizer}.magma
4495 ▷ $(ORBITER\ PATH)\text{orbiter}\cdot\text{out}\cdot-v\cdot6\cdot\$
4496 ▷ ▷ -define-G\$
4497 ▷ ▷ -linear\_group-PGGL\_2\cdot4\cdot-end\$
4498 ▷ ▷ -with-G\-do\$
4499 ▷ ▷ -group\_theoretic\_activity\$
4500 ▷ ▷ ▷ -centralizer\_of\_element."2A":"1,0,0,1,1"\$
4501 ▷ ▷ ▷ -report\$
4502 ▷ ▷ -end
4503 ▷ pdflatex\ PGGL\_2\_4\_elt\_2A\_centralizer.tex
4504 ▷ open\ PGGL\_2\_4\_elt\_2A\_centralizer.pdf
4505 ▷
4506
4507
4508
4509
4510
4511
4512 PGGL\_3\_4\_classes:
4513 ▷ $(ORBITER\ PATH)\text{orbiter}\cdot\text{out}\cdot-v\cdot3\cdot$
4514 ▷ ▷ -define-G\$
4515 ▷ ▷ -linear\_group-PGGL\_3\cdot4\cdot$
4516 ▷ ▷ -end\$
4517 ▷ ▷ -with-G\-do\$
4518 ▷ ▷ -group\_theoretic\_activity\$
4519 ▷ ▷ ▷ -classes\$
4520 ▷ ▷ -end
4521 ▷ pdflatex\ PGGL\_3\_4\_classes\_out.tex
4522 ▷ open\ PGGL\_3\_4\_classes\_out.pdf
4523
4524
4525 ˄ #1,3,3,1,3,2,3,0,3,0,  
4526 ˄ # is an element of order 21  
4527
4528
4529 ˄ ▷ -subgroup\_by\_generators:"singer":"21".1\$
4530 ˄ ▷ "0,1,0,0,0,1,2,1,1"\$
4531
4532
4533
4534
4535
4536 classes\_PGGL\_4\_4:\n4537 ▷ $(ORBITER\ PATH)\text{orbiter}\cdot\text{out}\cdot-v\cdot3\cdot$
4538 ▷ ▷ -magma\_path/usr/local/magma/\$
4539 ▷ ▷ -define-G\$
4540 ▷ ▷ -linear\_group-PGGL\_4\_4\cdot-end\$

477
group-theoretic activity

end

# the -find_subgroup -command is too specialized

subgroups_PGL.4.5:

$ (ORBITER_PATH) orbiter.out -v.6 \$ (ORBITER_PATH) orbiter.out -v.6 \$ (ORBITER_PATH) orbiter.out -v.6 \$ (ORBITER_PATH) orbiter.out -v.6 \$ (ORBITER_PATH) orbiter.out -v.6

pdflatex PGL.4.5_report.tex

open PGL.4.5_report.pdf

classes_PGL.4.5:

pdflatex PGL.4.5_classes.out.tex

open PGL.4.5_classes.out.pdf

# 163-classes

# two-classes-of-elements-of-order-3

# Order-of-element.=3 Class.size.=310000 Centralizer.order.=93600 Normalizer.order.=187200

# of-order-3-and-with-0-fixed-points.

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

# Class.size.=10075000 Centralizer.order.=2880 Normalizer.order.=5760

# of-order-3-and-with-6-fixed-points.
PGL_4_5_3B_class_again:
>>> $(ORBITER_PATH)orbiter.out-v.6.-define G:\n
>>> -linear_group-PGL.4.5-end\n
>>> -with-G-do\n
>>> -group_theoretic_activity\n
>>> -conjugacy_class_of\n
>>> "0,0,0,1,2,3,0,1,0,3,4,0,1,2,1".\n
>>> -end\n
>>> \n
search_primitive_poly_q5_deg3:
>>> $(ORBITER_PATH)orbiter.out-v.6-\n
>>> -search_for_primitive_polynomial_in_range 5.5.3.3\n
>>> #OK, we found an irreducible and primitive polynomial \n
>>> X^3+X^2+2 \n
GL_3.5_singer_power:
>>> $(ORBITER_PATH)orbiter.out-v.6.-define G\n
>>> -linear_group-GL.3.5-end\n
>>> -with-G-do\n
>>> -group_theoretic_activity\n
>>> -raise_to_the_power\n
>>> "0,1,0,0,1,3,0,4".31\n
>>> -end\n
>>> pdflatex-GL_3.5_power.tex\n
>>> open-GL_3.5_power.pdf\n
PGL_4_5_norm_31:
>>> $(ORBITER_PATH)orbiter.out-v.6.-define G\n
>>> -linear_group-PGL.4.5-end\n
>>> -with-G-do\n
>>> -group_theoretic_activity\n
>>> -normalizer_of_cyclic_subgroup "31"\n
>>> "2,0,0,0,0,1,0,0,0,0,1,0,3,0,4"\n
>>> -end\n
>>> pdflatex-normalizer_of_31_in_PGL_4.5.tex\n
>>> open-normalizer_of_31_in_PGL_4.5.pdf\n
#normalizer_has-order-372\n
#1,0,0,0,0,4,0,0,0,0,4,0,0,0,0,4,\n
#1,0,0,0,0,3,0,0,0,0,3,0,0,0,0,3,\n
#1,0,0,0,0,4,0,0,0,0,2,0,0,0,0,4,\n
#1,0,0,0,0,1,0,0,0,0,1,0,1,1,3,\n
#\n
>>> -normalizer_of_cyclic_subgroup "31"."1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,3"
Normalizer of $\mathbb{Z}_2^2$ in $\text{PGL}_2(9)$:

```
$\text{ORBITER}\_\text{PATH}\verb|orbiter.out| -v 4
\text{define} G \text{--linear group --PGL}_2(9)
\text{subgroup by generators:} \mathbb{Z}_2^2 \cdot 2
"2,0,0,1,0,1,0,0,0":-end
\text{do}
\text{group theoretic activity}
\text{report}
\text{orbits on points}
\text{export trees}
\text{end}
```

```
\text{pdflatex PGL}_2(9)\_\text{Subgroup}_\mathbb{Z}_2^2\_4\_\text{normalizer}.\text{tex}
\text{open-PGL}_2(9)\_\text{Subgroup}_\mathbb{Z}_2^2\_4\_\text{normalizer}.\text{pdf}
```

Section 6.1: Orbit Algorithms

```
\text{T3r1_orbits:}
$\text{ORBITER}\_\text{PATH}\verb|orbiter.out| -v 4
\text{define} G 
\text{linear group --GL}_2(9)\_\text{wr Sym}_3 \cdot 2 \cdot 3
\text{on_rank_one_tensors--end}
\text{do}
\text{group theoretic activity}
\text{report}
\text{orbits on points}
\text{export trees}
\text{end}
```

# Chapter 6: -- Orbit Algorithms

# Section 6.1: -- Orbit Algorithms
\begin{verbatim}
4678  \pdfflatex GL_2_2_wreath_Sym3_orbits_report.tex
4679  \open GL_2_2_wreath_Sym3_orbits_report.pdf
4680  \>
4681  \texttt{T3r1.orbits}
4682  \>
4683  \>
4684  \>
4685  \>
4686  \>
4687  \>
4688  \>
4689  \>
4690  \>
4691  \>
4692  \>
4693  \>
4694  \>
4695  \>
4696  \>
4697  \>
4698  \>
4699  \>
4700  \>
4701  \>
4702  \>
4703  \>
4704  \>
4705  \>
4706  \>
4707  \>
4708  \>
4709  \>
4710  \>
4711  \>
4712  \>
4713  \>
4714  \>
4715  \>
4716  \>
4717  \>
4718  \>
4719  \>
4720  \>
4721  \>
4722  \>
4723  \>
4724  \>
\end{verbatim}
\begin{verbatim}
4725  \verbatimline{f(9,0,0,2,1,0,0,0,0,1,0,0,1,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0)\
4726  \verbatimline{f(1,0,0,0,1,1,2,0,0,0,1,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0)\
4727  \verbatimline{f(1,0,3,0,1,1,1,3,0,0,2,0,0,0,0,2,1,0,0,3,0,1,1,1,3,0,0,2,0,0,0,0,2,1)\
4728  \verbatimline{-end-}\
4729  \verbatimline{-with-G-do-}\
4730  \verbatimline{-group_theoretic_activity-}\
4731  \verbatimline{-orbits_on_group_elements_under_conjugation-}\
4732  \verbatimline{2C_orbit_under_PGGL_4_4_elements_coded.csv-}\
4733  \verbatimline{2C_orbit_under_PGGL_4_4_transporter.csv-}\
4734  \verbatimline{-end}\
4735  \verbatimline{pdflatex-subgroups_of_order_4.tex}\
4736  \verbatimline{open-subgroups_of_order_4.pdf}\
4737  \verbatimline{pdflatex-}\
4738  \verbatimline{#The distribution of orbit lengths is: \( (1, 2, 15, 20, 24^3, 30, 40, 240, 256, 480, 512, 960^2, 1280, 1920^2, 2560^4, 3840, 5120, 6144^3, 7680, 10240) \)\
4739  \verbatimline{#group_theoretic_activity: do_orbits_on_group_elements_under_conjugation-after_Classes.compute_all_point_orbits}\n4740  \verbatimline{#found:29 conjugacy classes}\n4741  \verbatimline{#User time: 0:57}\
4742  \verbatimline{orbits_on_conics_q13:}\
4743  \verbatimline{$(ORBITER\_PATH)\verb|orbiter.out-|v.4-}\
4744  \verbatimline{-define-G-linear_group-PGL-3-13-}\
4745  \verbatimline{-with-G-do-}\
4746  \verbatimline{-group_theoretic_activity-}\
4747  \verbatimline{-orbits_on_polynomials-2-}\
4748  \verbatimline{-end-}\
4749  \verbatimline{pdflatex-poly_orbits_d2_n2_q13.tex}\
4750  \verbatimline{open-poly_orbits_d2_n2_q13.pdf}\
4751  \verbatimline{orbits_cubic_curves_q2:}\
4752  \verbatimline{$(ORBITER\_PATH)\verb|orbiter.out-|v.4-}\
4753  \verbatimline{-define-G-linear_group-PGL-3-2-}\
4754  \verbatimline{-with-G-do-}\
4755  \verbatimline{-group_theoretic_activity-}\
4756  \verbatimline{-orbits_on_polynomials-3-}\
4757  \verbatimline{-end-}\
4758  \verbatimline{pdflatex-poly_orbits_d3_n3_q2.tex}\
4759  \verbatimline{open-poly_orbits_d3_n3_q2.pdf}\
4760  \verbatimline{orbits_cubic_curves_q2_with_draw_tree:}\
4761  \verbatimline{$(ORBITER\_PATH)\verb|orbiter.out-|v.4-}\
\end{verbatim}

4770 \> \> -draw_options-yout-500000-radius-15-nodes_empty\ 
4771 \> \> \> -line_width-0.5-y_stretch-0.25.embedded-end\ 
4772 \> \> \> -define G.linear_group-PGL.3.2.end\ 
4773 \> \> \> -with G-do\ 
4774 \> \> \> -group_theoretic_activity\ 
4775 \> \> \> \> -orbits_on_polynomials.3\ 
4776 \> \> \> \> \> -orbits_on_polynomials_draw_tree.6\ 
4777 \> \> \> \> -end\ .
4778
4779
4780 poly.orbits_d3.n3.q2.csv:
4781 \> \> $(ORBITER_PATH)orbiter.out.-v.4\ 
4782 \> \> \> -draw_options-yout-500000-radius-15-nodes_empty\ 
4783 \> \> \> \> -line_width-0.5-y_stretch-0.25.embedded-end\ 
4784 \> \> \> \> -define G.linear_group-PGL.4.2.end\ 
4785 \> \> \> \> -with G-do\ 
4786 \> \> \> \> -group_theoretic_activity\ 
4787 \> \> \> \> \> -orbits_on_polynomials.3\ 
4788 \> \> \> \> \> \> -orbits_on_polynomials_draw_tree.6\ 
4789 \> \> \> \> \> \> -end\ .
4790
4791 poly.orbits_d3.n3.q2.get.ranks:
4792 \> \> $(ORBITER_PATH)orbiter.out.-v.4\ 
4793 \> \> \> -csv_file_select_cols-poly.orbits_d3.n3.q2.csv.0
4794 \> \> \> #pdflatex-poly.orbits_d3.n3.q2.tex
4795 \> \> \> #open-poly.orbits_d3.n3.q2.pdf
4796
4797
4798
4799 T4.orbits:
4800 \> \> $(ORBITER_PATH)orbiter.out.-v.4\ 
4801 \> \> \> -define G\ 
4802 \> \> \> \> -linear_group-GL.d.q_wr_Sym.n.2.2.4\ 
4803 \> \> \> \> \> -on_tensors.end\ 
4804 \> \> \> \> \> \> -with G-do\ 
4805 \> \> \> \> \> \> \> -group_theoretic_activity\ 
4806 \> \> \> \> \> \> \> \> -report\ 
4807 \> \> \> \> \> \> \> \> \> -orbits_on_points\ 
4808 \> \> \> \> \> \> \> \> \> \> -end\ 
4809 \> \> \> \> \> \> \> \> \> pdflatex-GL.2.2.wreath_Sym4.res65535.orbits.tex
4810 \> \> \> \> \> \> \> \> open-GL.2.2.wreath_Sym4.res65535.orbits.pdf
4811 \> \> \> \> \> \> \> \> #pdflatex-GL.2.2.wreath_Sym4.report.tex
4812 \> \> \> \> \> \> \> \> #open-GL.2.2.wreath_Sym4.report.pdf
4813
4814
4815
4816 T4r1.orbits:
4817 > $(ORBITER_PATH)orbiter.out\-v\4\n
4818 > -define-G-linear_group-GL_d_q_wr_Sym_n\2:2:4\n
4819 > -on_rank_one_tensors-end\n
4820 > -with-G-do\n
4821 > -group_theoretic_activity\n
4822 > -orbits_on_points\n
4823 > -export_trees\n
4824 > -report\n
4825 > -end\n
4826 > pdflatex-GL_2:2_wreath_Sym4_orbits_report.tex

4827 > open-GL_2:2_wreath_Sym4_orbits_report.pdf

4828

4829

4830 T4r1.orbits_draw:

4831 > $(ORBITER_PATH)orbiter.out\-v\3\n
4832 > -draw_layered_graph-GL_2:2_wreath_Sym4_res81_0.layered_graph\n
4833 > -radius-400--spanning_tree--embedded\n
4834 > -line_width\1.1\-x_stretch\2.5\-scale\0.15\n
4835 > -end

4836 > #pdflatex-GL_2:2_wreath_Sym3_report.tex

4837 > #open-GL_2:2_wreath_Sym3_report.pdf

4838 > pdflatex-GL_2:2_wreath_Sym4_res81_0.draw.tex

4839 > open-GL_2:2_wreath_Sym4_res81_0_draw.pdf

4840

4841

4842 T4r1.orbits_4:

4843 > $(ORBITER_PATH)orbiter.out\-v\4\n
4844 > -orbiter_path$(ORBITER_PATH)\n
4845 > -define-G-linear_group-GL_d_q_wr_Sym_n\2:2:4\n
4846 > -on_rank_one_tensors-end\n
4847 > -with-G-do\n
4848 > -group_theoretic_activity\n
4849 > -poset_classification_control\-problem_label-T4r1-W\n
4850 > -bit_depth\4\-draw_options--end\-draw_poset--report--end\n
4851 > -end\n
4852 > -orbits_on_subsets\4\n
4853 > -report\n
4854 > -end

4855 > pdflatex-T4r1_poset.tex

4856 > open-T4r1_poset.pdf

4857 > #pdflatex-GL_2:2_wreath_Sym4_report.tex

4858 > #open-GL_2:2_wreath_Sym4_report.pdf

4859

4860

4861

4862 PGGL_2:8_on_conic_orbits:

4863 > $(ORBITER_PATH)orbiter.out\-v\4
```
4864 ▶ ▶ -define G \ 
4865 ▶ ▶ -linear_group -PGGL 2 8 -PGL20nConic -end \ 
4866 ▶ ▶ -with G -do \ 
4867 ▶ ▶ -group_theoretic_activity \ 
4868 ▶ ▶ ▶ -orbits_on_points \ 
4869 ▶ ▶ ▶ -report \ 
4870 ▶ ▶ ▶ -end \ 
4871 ▶ pdflatex PGGL 2 8 OnConic 2 8 orbits report.tex \ 
4872 ▶ open PGGL 2 8 OnConic 2 8 orbits report.pdf \ 
4873 \ 
4874 \ 
4875 \ 
4876 # example from the Fining manual, page 107: \ 
4877 \ 
4878 PGGL 7 8 orbits: \ 
4879 ▶ $(ORBITER_PATH) orbiter.out -v 4 \ 
4880 ▶ ▶ -define G \ 
4881 ▶ ▶ -linear_group -PGGL 7 8 -end \ 
4882 ▶ ▶ -with G -do \ 
4883 ▶ ▶ -group_theoretic_activity \ 
4884 ▶ ▶ ▶ -report \ 
4885 ▶ ▶ ▶ -orbits_on_points \ 
4886 ▶ ▶ ▶ -end \ 
4887 \ 
4888 # 1 min 31 sec on Mac \ 
4889 \ 
4890 \ 
4891 \ 
4892 # Section 6.2: Poset Classification \ 
4893 \ 
4894 SECTION POSET CLASSIFICATION: \ 
4895 \ 
4896 poset of 4 subsets: \ 
4897 ▶ $(ORBITER_PATH) orbiter.out -v 3 \ 
4898 ▶ ▶ -or iter_path $(ORBITER_PATH) \ 
4899 ▶ ▶ -define G -linear_group -PGL 2 3 -identity_group -end \ 
4900 ▶ ▶ -with G -do \ 
4901 ▶ ▶ -group_theoretic_activity \ 
4902 ▶ ▶ ▶ -poset_classification_control \ 
4903 ▶ ▶ ▶ ▶ -problem_label poset 4 \ 
4904 ▶ ▶ ▶ ▶ -w -depth 4 \ 
4905 ▶ ▶ ▶ ▶ -draw_options -radius 200 -end \ 
4906 ▶ ▶ ▶ ▶ -report -end \ 
4907 ▶ ▶ ▶ ▶ -draw_poset \ 
```
4910  ▷ ▷ ▷ -end\n4911  ▷ ▷ ▷ -orbits_on_subsets-4\n4912  ▷ ▷ ▷ -report\n4913  ▷ ▷ -end
4914  ▷ pdflatex-PGL_2_3_Identity_2_3_report.tex
4915  ▷ pdflatex-poset_4_poset.tex
4916  ▷ open-PGL_2_3_Identity_2_3_report.pdf
4917  ▷ open-poset_4_poset.pdf
4918
4919  poset_of_4subsets_draw:
4920  ▷ $(ORBITER_PATH)orbiter.out-v.3\n4921  ▷ ▷ -draw_layered_graph-poset_4_poset_lvl_4.layered_graph.\n4922  ▷ ▷ -radius 300-embedded-line_width 1.1-\ y_stretch 0.9-scale 0.25-\n4923  ▷ ▷ -end
4924  ▷ pdflatex-poset_4_poset_lvl_4.draw.tex
4925  ▷ open-poset_4_poset_lvl_4.draw.pdf
4926
4927
4928  poset_of_5subsets:
4929  ▷ $(ORBITER_PATH)orbiter.out-v.3\n4930  ▷ ▷ -orbiter_path-$ORBITER_PATH-\n4931  ▷ ▷ -define G-linear_group-PGL_2_4-\ identity_group-end-\n4932  ▷ ▷ -with G-do-\n4933  ▷ ▷ -group_theoretic_activity-\n4934  ▷ ▷ ▷ -poset_classification_control\:-\ problem_label-poset_5-\n4935  ▷ ▷ ▷ ▷ -W-depth 5-draw_options-radius 150-end-\n4936  ▷ ▷ ▷ ▷ -report-end-draw_poset-end-\n4937  ▷ ▷ ▷ -orbits_on_subsets 5-\n4938  ▷ ▷ ▷ -report-\n4939  ▷ ▷ -end
4940  ▷ pdflatex-poset_5_poset.tex
4941  ▷ open-poset_5_poset.pdf
4942
4943  poset_of_5subsets_draw:
4944  ▷ $(ORBITER_PATH)orbiter.out-v.3\n4945  ▷ ▷ -draw_layered_graph-poset_5_poset_lvl_5.layered_graph.\n4946  ▷ ▷ -radius 300-embedded-line_width 1.1-\ y_stretch 0.9-scale 0.25-\n4947  ▷ ▷ -end
4948  ▷ pdflatex-poset_5_poset_lvl_5.draw.tex
4949  ▷ open-poset_5_poset_lvl_5.draw.pdf
4950
4951
4952
4953  V_3.2_trivial:
4954  ▷ $(ORBITER_PATH)orbiter.out-v.5-\n4955  ▷ ▷ -orbiter_path-$ORBITER_PATH-\n4956  ▷ ▷ -define G-linear_group-PGL_3_2-identity_group-end-\n
486
-with:G-do-

-group_theoretic_activity-
-poset_classification_control-

-problem_label:V_3.2_trivial-
-W-depth:3-node_label_is_element-
-draw_options-
-radius:200-embedded-
-end-
-report:-end-
-draw_poset-
-end-
-orbits_on_subspaces:3-
-report-
-end-

-define:G-linear_group-PGL_4.2-identity_group-
-with:G-do-
-group_theoretic_activity-
-poset_classification_control-

-problem_label:V_4.2_trivial-
-W-depth:3-node_label_is_element-
-draw_options-
-radius:200-embedded-
-end-
-report:-end-
-draw_poset-
-end-
orbits_on_subspaces:4-
-report-
-end-

#pdflatex PGL_4.2_Identity_4.2_report.tex
#open PGL_4.2_Identity_4.2_report.pdf
#pdflatex PGL_4.2_Identity_4.2_poset.tex
#open PGL_4.2_Identity_4.2_poset.pdf
Section 6.3: Orbits on Subsets

SECTION ORBITS ON SUBSETS:

PG(2,2) has
\(2^3 + 2^2 + 2^1 + 1 = 15\) points.

PG(3,3) has
\(3^3 + 3^2 + 3^1 + 1 = 27 + 9 + 3 + 1 = 40\) points.
PG_3.2_subsets_again:

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

-\text{orbiter_path}$\text{(ORBITER\_PATH)}\backslash$

-define-F\text{-finite_field-q.2-end}\backslash$

-define-G\text{-linear_group-PGL.4-F-end}\backslash$

-with-G-do$

-group_theoretic_activity$

-poset_classification_control$

-problem_label-PGL.4.2$

-depth.15$

-draw_options$

-radius.200-embedded$

-end$

-report-end$

-orbits_on_subsets.15$

-report$

-end$

\text{pdflatex-PGL.4.2\_poset\_lvl\_15.tex}$

\text{open-PGL.4.2\_poset\_lvl\_15.pdf}$

\text{pdflatex-PGL.4.2\_poset.tex}$

\text{open-PGL.4.2\_poset.pdf}$

\text{pdflatex-PGL.4.2\_poset\_detailed\_lvl\_15.tex}$

\text{open-PGL.4.2\_poset\_detailed\_lvl\_15.pdf}$
PGL\_3\_2\_singer:

$(\text{ORBITER\_PATH})\text{orbiter.out}$-\text{v.3}\
-\text{orbiter_path}$$(\text{ORBITER\_PATH})$$\cdot$
-\text{define}\text{-linear}\text{\_group}$-$\text{PGL\_3\_2\_singer\_1}$-\text{-end}\
-\text{with}\text{-G\_do}\
-\text{-group\_theoretic\_activity}\
-\text{-poset\_classification\_control}\
-\text{-problem\_label}$\text{PGL\_3\_2\_singer\_1}\text{-W\_depth\_7}$\
-\text{-draw_poset}\
-\text{-report\_end}\
-\text{-end}\
-\text{orbits\_on\_subsets\_7}\
-\text{-report}\
-\text{-end}\
pdflatex \text{PGL\_3\_2\_singer\_1\_poset.tex}\
open \text{PGL\_3\_2\_singer\_1\_poset.pdf}\
PGL\_3\_2\_on\_lines:

$(\text{ORBITER\_PATH})\text{orbiter.out}$-\text{v.3}\
-\text{orbiter_path}$$(\text{ORBITER\_PATH})$$\cdot$
-\text{define}\text{-linear}\text{\_group}$-$\text{PGL\_3\_2\_on\_k\_subspaces\_2\_end}\
-\text{with}\text{-G\_do}\
-\text{-group\_theoretic\_activity}\
-\text{-poset\_classification\_control}\
-\text{-problem\_label}$\text{PGL\_3\_2\_lines}\text{-W\_depth\_7}$\
-\text{-draw_poset}\
-\text{-report\_end}\
-\text{-end}\
-\text{orbits\_on\_subsets\_7}\
-\text{-report}\
-\text{-end}\
pdflatex \text{PGL\_3\_2\_lines\_poset.tex}\
open \text{PGL\_3\_2\_lines\_poset.pdf}\
PGL\_2\_5\_on\_subsets:

$(\text{ORBITER\_PATH})\text{orbiter.out}$-\text{v.10}\
-\text{orbiter_path}$$(\text{ORBITER\_PATH})$$\cdot$
-\text{define}\text{-linear}\text{\_group}$-$\text{PGL\_2\_5\_end}\
-\text{with}\text{-G\_do}\
-\text{-group\_theoretic\_activity}
PGL\_2\_7\_on\_subsets:
\$\text{(ORBITER PATH)}\text{orbiter.out} -v.10\$
\text{-orbiter_path}\$\text{(ORBITER PATH)}\$
\text{-define}\text{-G}\text{-linear}\text{-group}\text{-PGL}\_2\_7\text{-end}\$
\text{-with}\text{-G}\text{-do}\$
\text{-group}\text{-theoretic}\text{-activity}\$
\text{-poset}\text{-classification}\text{-control}\$
\text{-problem}\text{-label}\text{-PGL}\_2\_7\text{-W}\text{-depth}\_8\$
\text{-draw}\text{-poset}\$
\text{-draw_options}\text{-radius}200\text{-end}\$
\text{-report}\text{-end}\$
\text{-end}\$
\text{-orbits}\text{-on}\text{-subsets}\_8\$
\text{-report}\$
\text{-end}\$
\text{pdflatex}\text{-PGL}\_2\_7\_poset.pdf
\text{open}\text{-PGL}\_2\_7\_poset.pdf

PGGL\_2\_8\_on\_subsets:
\$\text{(ORBITER PATH)}\text{orbiter.out} -v.10\$
\text{-orbiter_path}\$\text{(ORBITER PATH)}\$
\text{-define}\text{-G}\text{-linear}\text{-group}\text{-PGGL}\_2\_8\text{-end}\$
\text{-with}\text{-G}\text{-do}\$
\text{-group}\text{-theoretic}\text{-activity}\$
\text{-poset}\text{-classification}\text{-control}\$
\text{-problem}\text{-label}\text{-PGGL}\_2\_8\text{-W}\text{-depth}\_9\$
\text{-draw}\text{-poset}\$
\text{-draw_options}\text{-radius}200\text{-end}\$
\text{-report}\text{-end}\$
\text{-end}\$
\text{-orbits}\text{-on}\text{-subsets}\_9\$
\text{-report}\$
\text{-end}\$
\text{pdflatex}\text{-PGGL}\_2\_8\_poset.pdf
\text{open}\text{-PGGL}\_2\_8\_poset.pdf
PGGL_2_9_on_subsets:
$\langle ORBITER\ PATH\rangle\ orbiter.out\ -v.10\$
-\ orbiter_path\ $(\langle ORBITER\ PATH\rangle)\$
-\ define-G\ -linear_group\ -PGGL_2_9\ -end\$
-\ with-G\ -do\$
-\ group_theoretic_activity\$
-\ poset_classification_control\$
-\ problem_label.PGGL_2_9\ -W\ -depth.10\$
-\ draw_poset\$
-\ draw_options\ -radius.200\ -end\$
-\ report\ -end\$
-\ orbits_on_subsets.10\$
-\ report\$
-\ end\$
-\ pdflatex\ PGGL_2_9_poset.tex
-\ open-PGGL_2_9_poset.pdf
PGGL_2_11_on_subsets:
$\langle ORBITER\ PATH\rangle\ orbiter.out\ -v.10\$
-\ orbiter_path\ $(\langle ORBITER\ PATH\rangle)\$
-\ define-G\ -linear_group\ -PGL_2_{11}\ -end\$
-\ with-G\ -do\$
-\ group_theoretic_activity\$
-\ poset_classification_control\$
-\ problem_label.PGL_2_{11}\ -W\ -depth.12\$
-\ draw_poset\$
-\ draw_options\ -radius.200\ -end\$
-\ report\ -end\$
-\ orbits_on_subsets.12\$
-\ report\$
-\ end\$
-\ pdflatex\ PGL_2_{11_poset.tex}
-\ open-PGL_2_{11_poset.pdf}
PGGL_2_16_on_subsets:
$\langle ORBITER\ PATH\rangle\ orbiter.out\ -v.3\$
-\ orbiter_path\ $(\langle ORBITER\ PATH\rangle)\$
-\ define-G\ -linear_group\ -PGGL_2_{16}\ -end\$
-\ with-G\ -do\$
-\ group_theoretic_activity\$
5238 ⋄ ⋄ ⋄ -poset_classification_control\$
5239 ⋄ ⋄ ⋄ ⋄ -problem_label:PGGL_2_16-W-depth:10\$
5240 ⋄ ⋄ ⋄ ⋄ -draw_poset\$
5241 ⋄ ⋄ ⋄ ⋄ -report-\end\$
5242 ⋄ ⋄ ⋄ -end\$
5243 ⋄ ⋄ ⋄ -orbits_on_subsets:10\$
5244 ⋄ ⋄ ⋄ -report\$
5245 ⋄ ⋄ -end
5246 ⪑ pdflatex PGGL_2_16_poset.tex
5247 ⪑ open-PGGL_2_16_poset.pdf
5248
5249
5250 PGGL_2_32_on_subsets:
5251 ⋄ \$(\text{ORBITER\_PATH})\text{orbiter.out-\_v:3}\$
5252 ⋄ ⋄ -orbiter_path:\$(\text{ORBITER\_PATH})\$
5253 ⋄ ⋄ -define-G-linear_group-PGGL_2-32-\end\$
5254 ⋄ ⋄ -with-G-do\$
5255 ⋄ ⋄ -group_theoretic_activity\$
5256 ⋄ ⋄ ⋄ -poset_classification_control\$
5257 ⋄ ⋄ ⋄ ⋄ -problem_label:PGGL_2_32-W-depth:8\$
5258 ⋄ ⋄ ⋄ ⋄ -draw_poset\$
5259 ⋄ ⋄ ⋄ ⋄ -report-\end\$
5260 ⋄ ⋄ ⋄ -end\$
5261 ⋄ ⋄ ⋄ -orbits_on_subsets:8\$
5262 ⋄ ⋄ ⋄ -report\$
5263 ⋄ ⋄ -end
5264 ⪑ pdflatex PGGL_2_32_poset.tex
5265 ⪑ open-PGGL_2_32_poset.pdf
5266
5267
5268
5269
5270
5271 PG_3_4_subsets:
5272 ⋄ \$(\text{ORBITER\_PATH})\text{orbiter.out-\_v:3}\$
5273 ⋄ ⋄ -orbiter_path:\$(\text{ORBITER\_PATH})\$
5274 ⋄ ⋄ -define-G-linear_group-PGGL_4-4-\end\$
5275 ⋄ ⋄ -with-G-do\$
5276 ⋄ ⋄ -group_theoretic_activity\$
5277 ⋄ ⋄ ⋄ -poset_classification_control:problem_label-PGGL_4_4\$
5278 ⋄ ⋄ ⋄ ⋄ -depth:5\$
5279 ⋄ ⋄ ⋄ ⋄ -draw_poset-draw_options-radius:200-\end\$
5280 ⋄ ⋄ ⋄ ⋄ -report-\end\$
5281 ⋄ ⋄ ⋄ -end\$
5282 ⋄ ⋄ ⋄ -orbits_on_subsets:5\$
5283 ⋄ ⋄ ⋄ -report\$
5284 ⋄ ⋄ -end
PGGL_2.9_orbits:
\$(\text{ORBITER \ PATH})\text{orbiter.out}\ -v\ 3\$
\-\text{orbiter_path}\$(\text{ORBITER \ PATH})\$
\-\text{define G=linear_group-PGGL.2.9=-end}\$
\-\text{with G=-do}\$
\-\text{group_theoretic_activity}\$
\-\text{poset_classification_control}\$
\-\text{problem_label-PGGL.2.9-W-depth.5}\$
\-\text{report=-end}\$
\-\text{draw_poset=-draw_options=-radius.200=-end}\$
\-\text{-end}\$
\-\text{orbits_on_subsets.5}\$
\-\text{-report}\$
\-\text{-end}\$
\text{pdflatex-PGGL.2.9_poset.tex}
\text{open-PGGL.2.9_poset.pdf}

PG_3.7_lines:
\$(\text{ORBITER \ PATH})\text{orbiter.out}\ -v\ 7\$
\-\text{orbiter_path}\$(\text{ORBITER \ PATH})\$
\-\text{define F=finite_field-q.7=-end}\$
\-\text{define G=linear_group-PGL.4.F=-on_k_subspaces.2=-end}\$
\-\text{with G=-do}\$
\-\text{group_theoretic_activity}\$
\-\text{poset_classification_control=-problem_label.7_lines}\$
\-\text{depth.3=-report=-end}\$
\-\text{draw_poset}\$
\-\text{draw_options=-radius.200=-embedded=-end}\$
\-\text{-end}\$
\-\text{orbits_on_subsets.3}\$
\-\text{-report}\$
\-\text{-end}\$
\text{pdflatex.7_lines_poset.tex}
\text{open.7_lines_poset.pdf}

PG0_5.2_on_subsets:
\$(\text{ORBITER \ PATH})\text{orbiter.out}\ -v\ 3\$
\-\text{orbiter_path}\$(\text{ORBITER \ PATH})\$

494
\begin{verbatim}
5332 \>>> -define F\finite_field\-q\2\-end\;
5333 \>>> -define G\linear_group\-PGO\5\-F\-end\;
5334 \>>> -with G\-do\;
5335 \>>> -group_theoretic_activity\;
5336 \>>> \>>> -poset_classification_control\;
5337 \>>> \>>> \>>> -problem_label\-PGO\5\2\;
5338 \>>> \>>> \>>> -depth\15\;
5339 \>>> \>>> \>>> -report\-end\;
5340 \>>> \>>> \>>> -draw_poset\;
5341 \>>> \>>> \>>> -w\;
5342 \>>> \>>> \>>> -end\;
5343 \>>> \>>> \>>> -orbits_on_subsets\15\;
5344 \>>> \>>> \>>> -report\;
5345 \>>> \>>> \>>> -end\;
5346 \>>> pflatex\-PGO\5\2\poset.tex
5347 \>>> open\-PGO\5\2\poset.pdf
5348
5349
5350
5351
5352 #----------------------------------------------------------------------------------------------
5353 # Section 6.4: Orbits on Subspaces
5354
5355
5356 SECTION_ORBITS_ON_SUBSPACES:
5357 \>
5358
5359 subspaces_\Op\4\2:
5360 \>
5361 \>$\\$(ORBITER_PATH)\orbiter.out\-v\5\;
5362 \>
5363 \>
5364 \>
5365 \>
5366 \>
5367 \>
5368 \>
5369 \>
5370 \>
5371 \>
5372 \>
5373 \>
5374 \>
5375 \>
5376
5377
\end{verbatim}
5378 #> $(ORBITER\_PATH)\texttt{orbiter.out} -v.5:\$
5379  #> > -\texttt{draw\_layered\_graph}\texttt{-PGL\_4\_2\_Orthogonal\_plus\_4\_2\_poset\_lvl\_4\_layered\_graph}:
5380  #> > > -radius.300.\texttt{--line\_width.1.1--y\_stretch.0.9--scale.0.25}:
5381  #> > > -end
5382  #> > pdflatex\texttt{PGL\_4\_2\_Orthogonal\_plus\_4\_2\_poset\_lvl\_4\_draw.tex}
5383  #> > open\texttt{PGL\_4\_2\_Orthogonal\_plus\_4\_2\_poset\_lvl\_4\_draw.pdf}
5384  #> > pdflatex\texttt{PGL\_4\_2\_Orthogonal\_plus\_4\_2\_report.tex}
5385  #> > open\texttt{PGL\_4\_2\_Orthogonal\_plus\_4\_2\_report.pdf}
5386  #> > pdflatex\texttt{PGL\_4\_2\_report.tex}
5387  #> > open\texttt{PGL\_4\_2\_report.pdf}
5388  #> > pdflatex\texttt{PGL\_4\_2\_poset.tex}
5389  #> > open\texttt{PGL\_4\_2\_poset.pdf}
5390  #> > PGL\_4\_2\_on\_subspaces:
5391  #> > > $(ORBITER\_PATH)\texttt{orbiter.out} -v.5:\$
5392  #> > > > -orbiter\_path $(ORBITER\_PATH)\$
5393  #> > > > -define-G -linear\_group -PGL\_4\_2 -end\$
5394  #> > > > -with-G -do\$
5395  #> > > > -group\_theoretic\_activity\$
5396  #> > > > > -poset\_classification\_control\$
5397  #> > > > > > -problem\_label -PGL\_4\_2 -W -depth.4\$
5398  #> > > > > > > -report -end\$
5399  #> > > > > > > -end\$
5400  #> > > > > > -orbits\_on\_subspaces.4\$
5401  #> > > > > > -report\$
5402  #> > > > > -end
5403  #> > > pdflatex\texttt{PGL\_4\_2\_report.tex}
5404  #> > open\texttt{PGL\_4\_2\_report.pdf}
5405  #> > pdflatex\texttt{PGL\_4\_2\_poset.tex}
5406  #> > open\texttt{PGL\_4\_2\_poset.pdf}
5407  #> > PGL\_4\_2\_singer_on\_subspaces:
5408  #> > > $(ORBITER\_PATH)\texttt{orbiter.out} -v.5:\$
5409  #> > > > -orbiter\_path $(ORBITER\_PATH)\$
5410  #> > > > -define-G -linear\_group -PGL\_4\_2 -singer.1 -end\$
5411  #> > > > -with-G -do\$
5412  #> > > > -group\_theoretic\_activity\$
5413  #> > > > > -node\_label\_is\_element\$
5414  #> > > > > > -draw\_poset\$
5415  #> > > > > > > -draw\_options -end\$
5416  #> > > > > > > > -problem\_label -PGL\_4\_2\_singer -W -depth.4\$
5417  #> > > > > > > > > -report -end\$
5418  #> > > > > > > > > -end\$
5419  #> > > > > > > > -orbits\_on\_subspaces.4\$
5420  #> > > > > > > > -report\$
5421  #> > > > > > > -end\$
5422  #> > > > > > -end\$
5423  #> > > > > > -report\$
5424  #> > > > > -end
PGL_8_2_singer_on_subspaces:

$\text{(ORBITER PATH)}$orbiter.out-v.5:

-exec -orbiter_path:$\text{(ORBITER PATH)}$

-exec -define G=linear_group-PGL_8_2-singer.1-end:

-exec -with G-do:

-exec -group_theoretic_activity:

-exec -poset_classification_control:

-exec -node_label_is_element:

-exec -draw_poset-draw_options-radius.150-end:

-exec -problem_label-PGL_8_2_singer-W-depth.8-report-end:

-exec -end:

-exec -orbits_on_subspaces.8:

-exec -report:

-exec -end:

-exec pdflatex-PGL_8_2_Singer_8_2_1_poset.tex

-exec open-PGL_8_2_Singer_8_2_1_poset.pdf

# May 7, 2020: 16:sec on Mac

# 1643 orbits in total

Op_6_2_orbits_on_subspaces:

$\text{(ORBITER PATH)}$orbiter.out-v.5:

-exec -orbiter_path:$\text{(ORBITER PATH)}$

-exec -define G=linear_group-PGL_6_2-orthogonal.1-end:

-exec -with G-do:

-exec -group_theoretic_activity:

-exec -poset_classification_control:

-exec -node_label_is_element:

-exec -draw_poset-draw_options-radius.200-end:

-exec -problem_label-Op_6_2-W-depth.6-report-end:

-exec -end:

-exec -orbits_on_subspaces.6:

-exec -report:

-exec -end:

-exec pdflatex-PGL_6_2_Orthogonal_plus_6_2_report.tex

-exec open-PGL_6_2_Orthogonal_plus_6_2_report.pdf

-exec pdflatex-PGL_6_2_Orthogonal_plus_6_2_poset.tex

-exec open-PGL_6_2_Orthogonal_plus_6_2_poset.pdf

Op_6_3_orbits_on_subspaces:
5472  \$\{\text{ORBITER\_PATH}\}\text{orbiter\_out\_v\_5}\$
5473  \>
5474  \>
5475  \>
5476  \>
5477  \>
5478  \>
5479  \>
5480  \>
5481  \>
5482  \>
5483  \>
5484  \>
5485  \>
5486  \>
5487  \>
5488  \#June\_3,\_2020\_on\_Mac:\_0\_sec
5489 5490 5491 5492  Op\_6\_11\_orbits\_on\_subspaces:
5493  \$\{\text{ORBITER\_PATH}\}\text{orbiter\_out\_v\_5}\$
5494  \>
5495  \>
5496  \>
5497  \>
5498  \>
5499  \>
5500  \>
5501  \>
5502  \>
5503  \>
5504  \>
5505  \>
5506  \>
5507  \>
5508  \>
5509 5510 5511  \#June\_3,\_2020\_on\_Mac:\_12\_sec
5512 5513 5514  Op\_8\_2\_orbits\_on\_subspaces:
5515  \$\{\text{ORBITER\_PATH}\}\text{orbiter\_out\_v\_5}\$
5516  \>
5517  \>
5518  \>
5519 5520 5498
Section 6.5: Arcs and Caps in Projective Spaces

SECTION_ARCS_AND_CAPS_IN_PROJECTIVE_SPACES:

PGL_3_27:

\$ (ORBITER_PATH) orbiter.out -v 5 \$

PGO_7_2_on_subspaces:

\$ (ORBITER_PATH) orbiter.out -v 20 \$

\$ -orbits_on_subspaces 7 \$

pdflatex PGL_8_2_Orthogonal_plus_8_2_poset.tex

open-PGL_8_2_Orthogonal_plus_8_2_poset.pdf
AGGL_2.27:
\$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\cdot 5\$
\begin{verbatim}
define G
linear_group PGL 3 27 end
with G do
group_theoretic_activity
do report
end
pdflatex PGL_3_27_report.tex
open PGL_3_27_report.pdf
\end{verbatim}

hyperoval 4 classify:
\$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\cdot 4$
\begin{verbatim}
define F finite_field q 4 end
define P projective_space 2 F end
with P do
projective_space_activity
do classify arcs
poset_classification_control
problem_label hyperoval q4 W depth 6
report end
d
end
end
pdflatex hyperoval q4 poset.tex
open hyperoval q4 poset.pdf
\end{verbatim}

hyperoval 8 classify:
\$(\text{ORBITER\ PATH})\text{orbiter.out}\ -v\cdot 4$
\begin{verbatim}
-orbit$\text{ORBITER\_PATH}\$
-define F\text{-finite\_field\text{-q\_8\_end}}
-define P\text{-projective\_space\_2\text{-F\_end}}
-with P\text{-do}
-projective\_space\_activity
-classify\_arcs
\text{-poset\_classification\_control}
-problem\_label\_hyperoval\_q\_8\_W\_depth\_10
-report\_end
-draw\_poset\_draw\_options\_radius\_200\_end
-target\_size\_10
-end
\text{pdflatex\_hyperoval\_q\_8\_poset.tex}
\text{open\_hyperoval\_q\_8\_poset.pdf}

frame\_stabilizer\_PGL:
\text{}\$(\text{ORBITER\_PATH})orbiter.out\_v\_4$
-define G
-linear_group\_PGL\_3\_8\_end
-with G\_do
-group\_theoretic\_activity
\text{-poset\_classification\_control}
-problem\_label\_frame\_q\_8\_W\_depth\_4
-draw\_options\_end
-report\_end
-end
-classify\_arcs
-target\_size\_4
-q\_8
-n\_3
-d\_2
-end
\text{pdflatex\_frame\_q\_8\_poset.tex}
\text{open\_frame\_q\_8\_poset.pdf}

frame\_stabilizer\_PGL:
\text{}\$(\text{ORBITER\_PATH})orbiter.out\_v\_4$
-define G
-linear_group\_PGL\_3\_8\_end
-with G\_do

frame\_stabilizer\_PGL:
\text{}\$(\text{ORBITER\_PATH})orbiter.out\_v\_4$
-define G
-linear_group\_PGL\_3\_8\_end
-with G\_do

501
```
5659 \[\triangleright\triangleright -\text{group-theoretic_activity}\backslash\]
5660 \[\triangleright\triangleright \ \triangleright\triangleright -\text{poset-classification_control}\backslash\]
5661 \[\triangleright\triangleright \ \triangleright\triangleright -\text{problem_label-frame_q8-}\backslash-	ext{W-}\text{depth-4}\backslash\]
5662 \[\triangleright\triangleright \ \triangleright\triangleright -\text{draw_options}\text{-end}\backslash\]
5663 \[\triangleright\triangleright \ \triangleright\triangleright -\text{report}\text{-end}\backslash\]
5664 \[\triangleright\triangleright \ \triangleright\triangleright -\text{end}\backslash\]
5665 \[\triangleright\triangleright \ \triangleright\triangleright -\text{classify_arcs}\backslash\]
5666 \[\triangleright\triangleright \ \triangleright\triangleright -\text{target_size-4}\backslash\]
5667 \[\triangleright\triangleright \ \triangleright\triangleright -q.8\backslash\]
5668 \[\triangleright\triangleright \ \triangleright\triangleright -n.3\backslash\]
5669 \[\triangleright\triangleright \ \triangleright\triangleright -d.2\backslash\]
5670 \[\triangleright\triangleright \ \triangleright\triangleright -\text{end}\backslash\]
5671 \[\triangleright\triangleright -\text{end}\backslash\]
5672 \[pdflatex-frame_q8_poset.tex\]
5673 \[open-frame_q8_poset.pdf\]
5674
5675
5676
5677  \texttt{hyperoval\_16\_classify:\}
5678 \[\triangleright \ $(\text{ORBITER\_PATH})\text{orbiter.out\_\_v.4}\backslash\]
5679 \[\triangleright \ \triangleright -\text{orbiter_path}\text{-}(\text{ORBITER\_PATH})\backslash\]
5680 \[\triangleright \ \triangleright -\text{defineF-}\text{-finite_field-}\text{-q.16}\text{-end}\backslash\]
5681 \[\triangleright \ \triangleright -\text{defineP-}\text{-projective_space-2}\text{-F-}\text{-end}\backslash\]
5682 \[\triangleright \ \triangleright -\text{-withP-}\text{-do}\backslash\]
5683 \[\triangleright \ \triangleright -\text{-projective_space_activity}\backslash\]
5684 \[\triangleright \ \triangleright \ \triangleright -\text{-classify_arcs}\backslash\]
5685 \[\triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-poset_classif}\text{-ication_control}\backslash\]
5686 \[\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-problem_label-hyperoval_q16-}\backslash-	ext{W-}\text{-depth-18}\backslash\]
5687 \[\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-report-}\text{-end}\backslash\]
5688 \[\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-end}\backslash\]
5689 \[\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-target_size-18}\backslash\]
5690 \[\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -d.2\backslash\]
5691 \[\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-end}\backslash\]
5692 \[\triangleright \ \triangleright -\text{-end}\backslash\]
5693 \[pdflatex-hyperoval_q16_poset.tex\]
5694 \[open-hyperoval_q16_poset.pdf\]
5695
5696
5697  \#\triangleright \ \triangleright \ \triangleright \ \triangleright \ \triangleright -\text{-draw_poset-}\text{-draw_options}\text{-end}\backslash\]
5698
5699
5700
5701  \texttt{hyperoval\_16\_1\_conic_type:\}
5702 \[\triangleright \ $(\text{ORBITER\_PATH})\text{orbiter.out\_\_v.2}\backslash\]
5703 \[\triangleright \ \triangleright -\text{defineF-}\text{-finite_field-}\text{-q.16}\text{-end}\backslash\]
5704 \[\triangleright \ \triangleright -\text{-defineP-}\text{-projective_space-2}\text{-F-}\text{-end}\backslash\]
5705 \[\triangleright \ \triangleright -\text{-withP-}\text{-do}\backslash\]
```
hyperoval $16 \cdot 1$ nonconical type:

# We found 17028 non-conical 6-subsets

# Eckardt point number distribution: $13^{252}, 9^{720}, 5^{2304}, 3^{13752}$

hyperoval $16 \cdot 2$ nonconical type:
We found $6188 = \binom{17}{5}$ 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
1,1,3,7,8
2,10,12,13,15
3,1,5,10,11
4,3,5,6,13
5,8,9,11,12
6,7,11,13,17
7,7,10,14,16
8,1,9,13,16
9,2,8,13,14
10,1,2,15,17
11,6,8,10,17
12,6,7,9,15
13,2,6,11,16
14,5,9,14,17
15,5,8,15,16
16,1,6,12,14
17,2,5,7,12
18,3,12,16,17
19,3,11,14,15

#END

hyperoval_16_stab_0_disjoint_sets_graph:

$(\text{ORBITER}\_\text{PATH})\text{orbiter.out-\text{-v-2}}$

$\text{define} -\text{-G-graph-disjoint_sets_graph}$

neighbors_of_0_with_4_removed.csv
5799  ▶  ▶  -end\n5800  ▶  ▶  -with\G\-do\n5801  ▶  ▶  ▶  -graph\-theoretic\-activity\n5802  ▶  ▶  ▶  -find\-cliques\n5803  ▶  ▶  ▶  ▶  -target\-size\-4\n5804  ▶  ▶  ▶  -end\n5805  ▶  ▶  -end\n5806  ▶  ▶  -print\-symbols
5807
5808
5809  #\-5\-cliques\-of\-size\-4
5810  #ROW,C0,C1,C2,C3
5811  #0,0,6,15,16
5812  #1,1,2,13,14
5813  #2,3,9,12,18
5814  #3,4,5,7,10
5815  #4,8,11,17,19
5816  #END
5817
5818  #clique\-0:\n5819  #0,2,3,9,10
5820  #6,7,11,13,17
5821  #15,5,8,15,16
5822  #16,1,6,12,14
5823  #partition::(1,6,12,14|2,3,9,10|5,8,15,16|7,11,13,17)
5824  #4\-is\-missing,\ it\-is\-the\-nucleus
5825  #0\-is\-missing\-is\-the\-chosen\-point
5826
5827
5828
5829
5830
5831
5832
5833  #\-nonconical\-6\-arcs\-are\-used\-for\-classifying\-cubic\-surfaces:\n5834
5835
5836
5837
5838
5839  nc\_arcs\_16:
5840  ▶  ▶  $(\text{ORBITER\_PATH})\text{orbiter.out}\-\text{-v\-4}\n5841  ▶  ▶  ▶  -define\F\-\text{-finite\_field}\-q\-16\-\text{-end}\n5842  ▶  ▶  ▶  -define\P\-\text{-projective\_space}\-2\-\text{-F\-end}\n5843  ▶  ▶  ▶  -with\P\-\text{-do}\n5844  ▶  ▶  ▶  -\text{-projective\_space\_activity}\n5845  ▶  ▶  ▶  ▶  -classify\_arcs\n
5893  ▷ $(ORBITER\_PATH) orbiter.out -v.3\`
5894  ▷ ▷ -define F -finite_field -q 64 -end\`
5895  ▷ ▷ -define f -formula "f"."f"."a*a+a"...\`
5896  ▷ ▷ -with F -do -finite_field_activity\`
5897  ▷ ▷ ▷ -evaluate f."a=2" -end
5898
5899  F64_frob:
5900  ▷ $(ORBITER\_PATH) orbiter.out -v.3\`
5901  ▷ ▷ -define F -finite_field -q 64 -end\`
5902  ▷ ▷ -define f -formula "f"."f"."a*a*a*a*a*a*a*a"\`
5903  ▷ ▷ -with F -do -finite_field_activity\`
5904  ▷ ▷ ▷ -evaluate f."a=61" -end
5905
5906
5907  # surfaces with 13 Eckardt points have OCN=0,98,99
5908
5909  surface_64_0:
5910  ▷ $(ORBITER\_PATH) orbiter.out -v.3\`
5911  ▷ ▷ -define F -finite_field -q 64 -end\`
5912  ▷ ▷ -define P -projective_space 3 F -end\`
5913  ▷ ▷ -with P -do\`
5914  ▷ ▷ -projective_space_activity\`
5915  ▷ ▷ ▷ -define_surface S -q 64 -catalogue 0 -end\`
5916  ▷ ▷ ▷ -end\`
5917  ▷ ▷ -end\`
5918  ▷ ▷ -with S -do\`
5919  ▷ ▷ -cubic_surface_activity\`
5920  ▷ ▷ ▷ -report\`
5921  ▷ ▷ ▷ ▷ -report_with_group\`
5922  ▷ ▷ ▷ -end
5923  ▷ pdflatex -surface_catalogue_q64_iso0_with_group.tex
5924  ▷ open -surface_catalogue_q64_iso0_with_group.pdf
5925
5926
5927
5928
5929  # makes it slow:
5930  ▷ ▷ ▷ ▷ ▷ -test_nb_Eckardt_points 13\`
5931  ▷ ▷ ▷ ▷ ▷ -report -select_orbits_by_level 6 -select_orbits_by_stabilizer_order_multiple_of 24 -end\`
5932  ▷ User time: 0:3
5933
5934
5935
5936  nc_arcs_128:
5937  ▷ $(ORBITER\_PATH) orbiter.out -v.4\`
5938  ▷ ▷ -define F -finite_field -q 128 -end\`

507
-define P = projective_space 2 : F - use_projectivity_subgroup - end \\
-define P = projective space 2 : F - use_projectivity subgroup - end \\
-define P = projective space activity \\
-classify arcs \\
-poset_classification_control \\
-problem_label nc arcs q128 d2 - W - depth 6 - end \\
-report - select_orbits_by_level 6 - end \\
-select_orbits_by_stabilizer_order_multiple_of 24 - end \\
-projective space activity \\
-classify arcs \\
-poset_classification_control \\
-problem_label nc arcs q256 d2 - W - depth 6 - report - select_orbits_by_level 6 - end \\
-select_orbits_by_stabilizer_order_multiple_of 24 - end \\
-target size 6 - end \\
-conic test \\
-end \\
-pdflatex nc arcs q256 d2 poset tex \\
-open nc arcs q256 d2 poset pdf
Example_F64:

\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \\
-define\text{-finite\_field} -q 64 -end \\
-define\text{-projective\_space} -3F -end \\
-with\text{-do} \\
\text{define\_space}\text{-activity}\text{-family\_general\_abcd\_52\_8\_52\_q\_64} \\
\text{-end} \\
\text{-with}\text{-S64\_abcd\_52\_8\_52\_do} \\
\text{-cubic\_activity} \\
\text{-report} \\
\text{pdflatex\_surface\_family\_general\_abcd\_q\_64\_a52\_b8\_c8\_d52\_report\_tex} \\
\text{six\_arcs\_4\_nbE13:} \\
\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \\
-define\text{-finite\_field} -q 4 -end \\
-define\text{-projective\_space} -2F -end \\
-with\text{-do} \\
\text{-projective\_space\_activity} \\
\text{-control\_six\_arcs\_problem\_label\_sixarcs\_q\_4} -end \\
\text{-six\_arcs\_not\_on\_conic\_filter\_by\_nb\_Eckardt\_points\_13} -end \\
\text{six\_arcs\_8\_nbE13:} \\
\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \\
-define\text{-finite\_field} -q 8 -end \\
-define\text{-projective\_space} -2F -end \\
-with\text{-do} \\
\text{-projective\_space\_activity} \\
\text{-control\_six\_arcs\_problem\_label\_sixarcs\_q\_8} -end \\
\text{-six\_arcs\_not\_on\_conic\_filter\_by\_nb\_Eckardt\_points\_13} -end \\
\text{six\_arcs\_16\_nbE13:} \\
\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \\
-define\text{-finite\_field} -q 16 -end \\
-define\text{-projective\_space} -2F -end \\
-with\text{-do} \\
\text{-projective\_space\_activity} \\
\text{six\_arcs\_8\_nbE13:} \\
\$(\text{ORBITER\_PATH})\text{orbiter.out} -v 3 \\
-define\text{-finite\_field} -q 8 -end \\
-define\text{-projective\_space} -2F -end \\
-with\text{-do} \\
\text{-projective\_space\_activity}
six_arcs_32_nbE13:

six_arcs_64_nbE13:

six_arcs_128_nbE13:

six_arcs_256_nbE13:
five_arcs_q13:

\begin{verbatim}
\$\text{ORBITER\_PATH}\text{oribiter.out}$-v-4\$
\$\text{define}\text{-}F\text{-}finite\text{-}field\text{-}q\text{13}\text{-}end\$
\$\text{define}\text{P}\text{-}projective\text{-}space\text{\_}2\text{\_}F\text{-}end\$
\$\text{with}\text{-}P\text{-}do\$
\$\text{-projective\_space\_activity\_}
\$\text{-classify\_arcs\_}
\$\text{-poset\_classification\_control\_}
\$\text{-problem\_label\_five\_arcs\_q13\_W\text{-}depth\text{-}5\_}
\$\text{-report\_end\_}
\$\text{-end\_}
\$\text{-target\_size\_5\_}
\$\text{-d\_2\_}
\$\text{-end\_}
\$\text{-end\_}
\$\text{pdflatex\_five\_arcs\_q13\_poset.tex}
\$\text{open\_five\_arcs\_q13\_poset.pdf}
\$\text{SECTION\_CUBIC\_CURVES:}
\$\text{cubic\_curves\_PG\_2\_4:}
\$\text{-orbit\_path\_}
\$\text{-define}\text{-}F\text{-}finite\text{-}field\text{-}q\text{3}\text{-}end\$
\$\text{-define}\text{P}\text{-}projective\text{-}space\text{\_}2\text{\_}F\text{-}end\$
\$\text{-with}\text{-}P\text{-}do\$
\$\text{-projective\_space\_activity\_}
\$\text{-classify\_cubic\_curves\_q\text{4}\_target\_size\text{\_}9\_n\text{\_}3\_d\text{\_}3\_}
\$\text{-poset\_classification\_control\_}
\end{verbatim}

Section 6.6: Cubic Curves
cubic_curves_PG_2.4:  
\$\{ORBITER_PATH\}orbiter.out-v.3\$
\-draw_options--radius 200--embedded--end\$
\-report--end\$
\-end\$
\pdflatex\-cc_4_poset.tex
\open\-cc_4_poset.pdf
\pdflatex\-cc_4_poset_lvl_9.tex
\open\-cc_4_poset_lvl_9.pdf
\pdflatex\Cubic_curves_q4.tex
\open\-Cubic_curves_q4.pdf
\pdflatex\-cc_4_poset_lvl_9.draw.tex
\open\-cc_4_poset_lvl_9.draw.pdf
\cubic_curves_PG_2.8:
\$\{ORBITER_PATH\}orbiter.out-v.3\-define G\$
\-define F\-finite_field=q.8--end\$
\-define F\-projective_space=2 F--end\$
\-with F\-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\$
\$\{ORBITER_PATH\}orbiter.out-v.2\-draw_matrix\$
\-input.csv.file=cc.8_KM.6.9.csv
\-box width=50--bit depth=8--end
\pdflatex\Cubic_curves_q8.tex
\open\-Cubic_curves_q8.pdf
\pdflatex\-cc_8_tree_lvl_9.tex
\open\-cc_8_tree_lvl_9.pdf
cubic_curves_PG_2.8.draw:

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

-\text{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:6.7-9.1$

-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:

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

-define F:\text{finite\_field}\_q.4-end$

-define P:\text{projective\_space}\_3\_F-end$

-with P-do$

-projective\_space\_activity$

-define_surface S:\text{q.4 catalogue}\_0-end$

-end
Family general F7:

Family general abcd 2 3 3 4:

Family general abcd 2 3 3 4 - do:

Family general abcd 2 3 3 4 - report:

Family general abcd 2 3 3 4 - report with group:

Family general abcd 2 3 3 4 - all quartic curves:

Family general abcd 2 3 3 4 - end

Family general abcd 2 3 3 4 - with S7_0 - do:

Family general abcd 2 3 3 4 - cubic surface activity:

Family general abcd 2 3 3 4 - report:

Family general abcd 2 3 3 4 - report with group:

Family general abcd 2 3 3 4 - end

Family general abcd 2 3 3 4 - with S7_0 - do:

Family general abcd 2 3 3 4 - cubic surface activity:

Family general abcd 2 3 3 4 - report:

Family general abcd 2 3 3 4 - report with group:

Family general abcd 2 3 3 4 - all quartic curves:

Family general abcd 2 3 3 4 - end
6264 ▷ pdflatex-surface_family_general_abcd_q7_a2_b3_c3_d4_report.tex
6265 ▷ open-surface_family_general_abcd_q7_a2_b3_c3_d4_report.pdf
6266 ▷
6267  # Fermat-with-18-Eckardt-points
6268  # no-automorphism-group, so no-report_with_group and no-all_quartic_curves
6269
6270
6271  # Joel:
6272
6273  eckardt_13_4_12:
6274  ▷ $(ORBITER_PATH)orbiter.out-v.6:\
6275  ▷ ▷ -define-F-finite_field-q.13-end:\
6276  ▷ ▷ -define-P-projective_space-3-F-end:\
6277  ▷ ▷ -with-P-do-\n6278  ▷ ▷ -projective_space_activity-\n6279  ▷ ▷ ▷ -define_surface-S2.1-q.13-\n6280  ▷ ▷ ▷ -family_Eckardt-4.12-end-\n6281  ▷ ▷ ▷ -end-\n6282  ▷ ▷ ▷ -with-S2.1-do-\n6283  ▷ ▷ ▷ -cubic_surface_activity-\n6284  ▷ ▷ ▷ -report-\n6285  ▷ ▷ ▷ -report_with_group-\n6286  ▷ ▷ ▷ -end
6287
6288
6289
6290
6291
6292
6293
6294  surface_8_0_catalogue:
6295  ▷ $(ORBITER_PATH)orbiter.out-v.3:\
6296  ▷ ▷ -define-F-finite_field-q.8-end:\
6297  ▷ ▷ -define-P-projective_space-3-F-end:\
6298  ▷ ▷ -with-P-do-\n6299  ▷ ▷ -projective_space_activity-\n6300  ▷ ▷ -define_surface-S8.0-q.8-catalogue-0-end-\n6301  ▷ ▷ -end-\n6302  ▷ ▷ -with-S8.0-do-\n6303  ▷ ▷ -cubic_surface_activity-\n6304  ▷ ▷ ▷ -report-\n6305  ▷ ▷ ▷ -report_with_group-\n6306  ▷ ▷ ▷ -end
6307  ▷ pdflatex-surface_catalogue_q8_iso0_report.tex
6308  ▷ open-surface_catalogue_q8_iso0_report.pdf
6309  ▷ pdflatex-surface_catalogue_q8_iso0_with_group.tex
6310  ▷ open-surface_catalogue_q8_iso0_with_group.pdf

515
surface_8_0_clean:

```bash
$(ORBITER_PATH)orbiter.out -v 3
```

```bash
-define F -finite_field -q 8 -end
```

```bash
-define P -projective_space -3 F -end
```

```bash
-with P -do
```

```bash
-projective_space_activity
```

```bash
-define_surface S8_0 -q 8 -catalogue 0
```

```bash
-select_double six "15,11,22,19,24,5,16,10,23,20,25,4"
```

```bash
-select_double six "3,2,1,0,5,4,9,8,7,6,11,10"
```

```bash
-transform_inverse "1,4,4,0,6,0,0,6,2,0,1,7,0,4,0,0"
```

```bash
-transform "4,4,0,0,0,1,0,1,0,0,0,0,0,1,0"
```

```bash
-transform_inverse "2,0,0,0,2,0,0,0,2,0,1,1,2,3,0"
```

```bash
-end -end
```

```bash
-with S8_0 -do
```

```bash
-cubic_surface_activity
```

```bash
-report
```

```bash
-report_with_group
```

```bash
-end
```

```bash
dpdfflatex surface_catalogue_q8_iso0_report.tex
```

```bash
don -surface_catalogue_q8_iso0_report.pdf
```

---

#clean_equation_for_Tekirdag-1:

surface_8_0b:

```bash
$(ORBITER_PATH)orbiter.out -v 3
```

```bash
-define F -finite_field -q 8 -end
```

```bash
-define P -projective_space -3 F -end
```

```bash
-with P -do
```

```bash
-projective_space_activity
```

```bash
-define_surface S8_0 -q 8 -catalogue 0
```

```bash
-select_double six "15,11,22,19,24,5,16,10,23,20,25,4"
```

```bash
-select_double six "3,2,1,0,5,4,9,8,7,6,11,10"
```

```bash
-transform_inverse "3,1,1,0,0,1,0,0,0,1,0,0,0,0,1,0"
```

```bash
-transform_inverse "2,0,0,0,1,0,0,0,1,0,0,0,0,1,0"
```

```bash
-end
```

```bash
-end
```

```bash
-with S8_0 -do
```

```bash
-cubic_surface_activity
```

```bash
-report
```

```bash
-report_with_group
```

```bash
-end
```

516
Eckardt_13:

$\text{(ORBITER \ PATH)orbiter.out-\v{3}}$

-define $F$ finite_field $q13$-end
-define $P$ projective_space $3F$-end

-with $P$-do

-projective_space_activity

-define_surface $S_{q13}$-q13

-family Eckardt $3:1$-end

-with $S_{q13}$-do

-cubic_surface_activity

-report

-report_with_group

-end

pdflatex surface_family_Eckardt_q13_a3_b1_with_group.tex

open_surface_family_Eckardt_q13_a3_b1_with_group.pdf

surface_13_0:

$\text{(ORBITER \ PATH)orbiter.out-\v{3}}$

-define $F$ finite_field $q13$-end
-define $P$ projective_space $3F$-end

-with $P$-do

-projective_space_activity

-define_surface $S_{13,0}$-q13-catalogue $0$-end

-end

-with $S_{13,0}$-do

-cubic_surface_activity

-report
surface_16_0:  
$\text{(ORBITER PATH)}$orbiter.out-v.3\$

#-define-F-\text{finite\_field}\-q\text{16}-\text{end}\$

#-define-P-\text{projective\_space\_3}\-F\-\text{end}\$

#-with-P-do$

#-projective\_space\_activity$

#-define\_surface\_S16_0\-q\text{16}-\text{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$

#-with\_S16_0-do$

#-cubic\_surface\_activity$

#-report$

#-report\_with\_group$

#-end$

#-pdflatex\text{\_surface\_catalogue\_q13\_iso0\_report\_tex}$

#-open\text{\_surface\_catalogue\_q13\_iso0\_report.pdf}$

#-clean\_equation\_for\_Tekirdag\_2$

#-rank\_of\_lines\((-\text{66591},,\text{26737},,\text{4093},,\text{69904},,\text{28376},,\text{26470},,\text{70160},,\text{69855},,\text{26208},,\text{5847},,\text{369},,\text{32230},,\text{529},,\text{30293},,\text{70068},,\text{2178},,\text{261},,\text{28666},,\text{8575},,\text{105},,\text{31694},,\text{0},,\text{51784},,\text{25209},,\text{22193},,\text{49862},,\text{274})$}

#-Rank\_of\_points\_on\_Klein\_quadric\:(,\text{29181},,\text{4677},,\text{29950},,\text{33},,\text{62496},,\text{429},,\text{1},,\text{9205},,\text{37},,\text{29964},,\text{29364},,\text{21501},,\text{4656},,\text{54735},,\text{5425},,\text{30105},,\text{754},,\text{6680},,\text{13354},,\text{758},,\text{30106},,\text{0},,\text{29209},,\text{48736},,\text{25595},,\text{33780},,\text{4657})$}

#-ai:\text{29181},,\text{4677},,\text{29950},,\text{33},,\text{62496},,\text{429}$

#-bi:\text{1},,\text{9205},,\text{37},,\text{29964},,\text{29364},,\text{21501}$

#-Tekirdag\_1
G13_8:

\$\text{\verb=$(ORBITER\_PATH)orbiter.out-v.3\textbackslash$}\$

-define F finite_field -q 8 -end
-define P projective_space 3 F -end
-with P do
-projective_space_activity

-define_surface T1 family G13_2 -q 8 -end
-with T1 do
-cubic_surface_activity
-report
-report_with_group
-end

踏入F13_q8_a2_\verb=with\_group::\textbackslash=pdf\textbackslash=latex\textbackslash=surface\_family\_G13\_q8\_a2\_\verb=with\_group::\textbackslash=text

open_surface_family G13_q8_a2_with_group.pdf

F13_8:

\$\text{\verb=$(ORBITER\_PATH)orbiter.out-v.3\textbackslash$}\$

-define F finite_field -q 16 -end
-define P projective_space 3 F -end
-with P do
-projective_space_activity

-define_surface T1 family F13_2 -q 8 -end
-with T1 do
-cubic_surface_activity
-report
-report_with_group
-end

踏入F13_q8_a2_\verb=with\_group::\textbackslash=pdf\textbackslash=latex\textbackslash=surface\_family\_F13\_q8\_a2\_\verb=with\_group::\textbackslash=text

open_surface_family F13_q8_a2_with_group.pdf

# Tekirdag-2:

F13_16:

\$\text{\verb=$(ORBITER\_PATH)orbiter.out-v.3\textbackslash$}\$

-define F finite_field -q 16 -end
-define P projective_space 3 F -end
-with P do
-projective_space_activity

-define_surface T2 family F13_2 -q 16 -end
-with T2 do
6495 \ldv \ldv -cubic_surface_activity\backslash \\
6496 \ldv \ldv \ldv -report\backslash \\
6497 \ldv \ldv \ldv \ldv -report_with_group\backslash \\
6498 \ldv \ldv \ldv \ldv -end \\
6499 \ldv \ldv \ldv \ldv pdflatex\surface_family\text{F13}\_q16\_a2\_with_group.tex \\
6500 \ldv \ldv \ldv \ldv open\_surface_family\text{F13}\_q16\_a2\_with_group.pdf \\
6501 \\
6502 \\
6503 #Tekirdag-3: \\
6504 \\
6505 \text{F13}\_32: \\
6506 \ldv \ldv $(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_3\backslash \\
6507 \ldv \ldv \ldv -define\_F\_finite_field\_q\_32\_end\backslash \\
6508 \ldv \ldv \ldv -define\_P\_projective_space\_3\_F\_end\backslash \\
6509 \ldv \ldv \ldv -with\_P\_do\backslash \\
6510 \ldv \ldv \ldv \ldv \ldv -projective_space_activity\backslash \\
6511 \ldv \ldv \ldv \ldv \ldv \ldv -define\_surface\_T3\_family\text{F13}\_2\_q\_32\_end\backslash \\
6512 \ldv \ldv \ldv \ldv \ldv \ldv \ldv -end\backslash \\
6513 \ldv \ldv \ldv \ldv \ldv \ldv \ldv -with\_T3\_do\backslash \\
6514 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -cubic_surface_activity\backslash \\
6515 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -report\backslash \\
6516 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -report_with_group\backslash \\
6517 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -end \\
6518 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv pdflatex\surface_family\text{F13}\_q32\_a2\_with_group.tex \\
6519 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv open\_surface_family\text{F13}\_q32\_a2\_with_group.pdf \\
6520 \\
6521 \\
6522 #Kapadokya-1: \\
6523 \\
6524 \text{F13}\_64a: \\
6525 \ldv \ldv $(\text{ORBITER\_PATH})\text{orbiter.out}\_v\_3\backslash \\
6526 \ldv \ldv \ldv -define\_F\_finite_field\_q\_64\_end\backslash \\
6527 \ldv \ldv \ldv \ldv -define\_P\_projective_space\_3\_F\_end\backslash \\
6528 \ldv \ldv \ldv \ldv \ldv -with\_P\_do\backslash \\
6529 \ldv \ldv \ldv \ldv \ldv \ldv -projective_space_activity\backslash \\
6530 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -define\_surface\_K1\_family\text{F13}\_2\_q\_64\_end\backslash \\
6531 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -end\backslash \\
6532 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -with\_K1\_do\backslash \\
6533 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -cubic_surface_activity\backslash \\
6534 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -report\backslash \\
6535 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -report_with_group\backslash \\
6536 \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv \ldv -end \\
6537 \\
6538 \\
6539 #Kapadokya-2: \\
6540 \\
6541 \text{F13}\_64b:
Colorado1:

```bash
$ (ORBITER_PATH)orbiter.out -v.3
> -define_F:-finite_field-q:64-end
> -define_P:-projective_space:3F-end
> -projective_space_activity
> -define_surface_K2-family_F13:18-q:64-end
> -end
> -with_K2:-do
> -cubic_surface_activity
> -report
> -report_with_group
> -end
```

```
define_P:-projective_space:3F-end
> -with_P:-do
```

```
define_surface_K2-family_F13:18-q:64-end
> -end
> -with_K2:-do
```

```
define_surface_K2-family_F13:18-q:64-end
> -end_inverse:"1,0,0,0,0,1,0,96,0,0,1,96,0,0,0,1,0"
> -end
> -end
> -with_CO-1:-do
```

```
cubic_surface_activity
> -report
> -report_with_group
> -end
```

```
cubic_surface_activity
> -report
> -report_with_group
> -end
```

```
cubic_surface_activity
```

Colorado2:

```bash
$ (ORBITER_PATH)orbiter.out -v.3
> -define_F:-finite_field-q:128-end
> -define_P:-projective_space:3F-end
> -projective_space_activity
> -define_surface_CO-2-family_F13:18-q:64-end
> -end
> -with_CO-2:-do
```

```
cubic_surface_activity
```

# recognize the arcs from Colorado-1,2,3:

```bash
define_surface_CO-2-family_F13:18-q:64-end
> -end_inverse:"1,0,0,0,0,1,0,32,0,0,1,32,0,0,0,1,0"
> -end
> -end
> -with_CO-1:-do
```

```
cubic_surface_activity
```

```bash
define_surface_CO-2-family_F13:18-q:64-end
> -end_inverse:"1,0,0,0,0,1,0,32,0,0,1,32,0,0,0,1,0"
> -end
> -end
> -with_CO-2:-do
```

```
cubic_surface_activity
```

521
Colorado3:
\begin{verbatim}
> $(ORBITER_PATH)orbiter.out -v.3:
> -define F=finite_field -q128 -end:
> -define P=projective_space 3 F -end:
> -with P -do:
> -projective_space_activity:
> -define_surface CO-3 -q128 -catalogue 928:
> -transform_inverse "1,0,0,0,0,1,0,59,0,0,1,59,0,0,0,1,0"
> -end:
> -end:
> -with P -do:
> -projective_space_activity:
> -define_surface CO-3 -family F13 -q128 -end:
> -end:
> -with P -do:
> -cubic_surface_activity:
> -report:
> -report with group:
> -end
\end{verbatim}

Colorado1:
\begin{verbatim}
> $(ORBITER_PATH)orbiter.out -v.3:
> -define F=finite_field -q128 -end:
> -define P=projective_space 3 F -end:
> -with P -do:
> -projective_space_activity:
> -define_surface CO-1 -family F13 -q128 -end:
> -end:
> -with P -do:
> -cubic_surface_activity:
> -report:
> -report with group:
> -end
\end{verbatim}

Colorado2:
\begin{verbatim}
> $(ORBITER_PATH)orbiter.out -v.3:
> -define F=finite_field -q128 -end:
> -define P=projective_space 3 F -end:
> -with P -do:
> -projective_space_activity:
> -define_surface CO-2 -family F13 -q128 -end:
> -end:
\end{verbatim}

# Colorado-1:

# Colorado-2:
# Colorado-3:

```
F13_128c:
$(ORBITER_PATH)orbiter.out --v=3
-defining_finite_field [-q=128] --end
-defining_projective_space [-3F] --end
-with CO-3 --do
-projective_space_activity
-report
-report_with_group
-end

move_two_lines:
$(ORBITER_PATH)orbiter.out --v=5
-defining_finite_field [-q=8] --end
-with CO-3 --do
-cubic_surface_activity
-report
-report_with_group
-end

F_alpha_beta_gamma_delta:
$(ORBITER_PATH)orbiter.out --v=3
-defining_finite_field [-q=7] --end
-with CO-3 --do
-projective_space_activity
-report
-report_with_group
-end

parse_and_evaluate
"F_alpha_beta_gamma_delta" "x0,x1,x2,x3"

alpha=2,beta=3,gamma=4,delta=5

end

dot --Tpng F_alpha_beta_gamma_delta.gv > F_alpha_beta_gamma_delta.png
```
F_abcd:

$(\text{ORBITER\_PATH})\text{orbiter.out}\_v.3$

-define F\_finite_field\_q.7\_end

-define F\_finite_field\_activity

-parse_and_evaluate "Fabcd" "X0,X1,X2,X3"

$(F\_abcd\text{eqn}) \text{"a=4,b=2,c=2,d=4"}$

-end

\#dot\_T.png \text{F} alpha beta gamma delta.gv \text{> F} alpha beta gamma delta.png

F_abcd_sweep_4.27.q7:

-define F\_finite_field\_q.7\_end

-define P\_projective_space\_3\_F\_end

-sweep_4.27.sweep_4.27.q7\_q.7\_by_equation "F\_abcd"

$(F\_abcd\text{eqn}) \text{"a=2,b=3,c=4,d=5"}$

-end

end

dot\_T.png \text{F} alpha beta gamma delta.gv \text{> F} alpha beta gamma delta.png

F\_alpha\_beta\_gamma\_delta\_q7\_override_group:

-define F\_finite_field\_q.7\_end

-define P\_projective_space\_3\_F\_end

-projective_space\_activity

-sweep_4.27.sweep_4.27.q7\_q.7\_by_equation "F\_abcd"

$(F\_abcd\text{eqn}) \text{"a=2,b=3,c=4,d=5"}$

-end

end

dot\_T.png \text{F} alpha beta gamma delta.gv \text{> F} alpha beta gamma delta.png

F\_alpha\_beta\_gamma\_delta\_q7\_override_group:

-define F\_finite_field\_q.7\_end

-define P\_projective_space\_3\_F\_end

-projective_space\_activity

-sweep_4.27.sweep_4.27.q7\_q.7\_by_equation "F\_abcd"

$(F\_abcd\text{eqn}) \text{"a=2,b=3,c=4,d=5"}$

-end

end

end

524
\[ F(\alpha, \beta, \gamma, \delta) = x_0, x_1, x_2, x_3 \]

\[ (F \alpha, \beta, \gamma, \delta) = 1, 2, 3, 4, 5 \]

\[ \alpha = 2, \beta = 3, \gamma = 4, \delta = 5 \]

\[ (F \alpha, \beta, \gamma, \delta) = 1, 0, 2, 5, 0, 1, 6, 1, 0, 0, 3, 5, 0, 0, 4, 4 \]

---

\# cubic surfaces with 15 lines:

\[ F(\alpha, \beta, \gamma, \delta) sweep 4 q3: \]

\[ $(ORBITER \ PATH) orbiter.out -v 3 \]

\[ define F - finite field - q 3 - end \]

\[ define P - projective space - 3 F - end \]

\[ with P - do \]

\[ projective space activity \]

\[ sweep 4 q3 - q 3 - by equation "F(\alpha, \beta, \gamma, \delta)" \]

\[ $(ORBITER \ PATH) orbiter.out -v 3 \]

\[ define F - finite field - q 7 - end \]

\[ define P - projective space - 3 F - end \]

\[ with P - do \]

\[ projective space activity \]

\[ sweep 4 q7 - q 7 - by equation "F(\alpha, \beta, \gamma, \delta)" \]
-projective_space_activity\n-sweep_4-sweep_4_q7:-q7:-by_equation\n"F_alpha_beta_gamma_delta"\n"\DF_{\{alpha,\beta,\gamma,\delta}\}D"\n"x0,x1,x2,x3"\n\$(F_ALPHA_BETA_GAMMA_DELTA)\n"alpha=2,beta=3,\gamma=4,\delta=5"\n"\Dalpha=2,\beta=3,\gamma=4,\delta=5\D"\n-end\n
#User-time: 0:30
#348 parameter sets

F_alpha_beta_gamma_delta_q7_recognize:
$\text{(ORBITER}\_\text{PATH})\text{or}biter\text{.out:}-v:2:\n\text{-define-F-\text{finite-field}-}\text{-q:49-}\text{-end}\n\text{-define-P-\text{projective}\_\text{space}-3F-}\text{-end}\n\text{-with-P-}\text{-do}\n\text{-projective}\_\text{space}\_\text{activity}\n\text{-classify}\_\text{surfaces}\_\text{with}\_\text{double}\_\text{sixes}\_\text{Surf-\text{-W-}\text{-end}\n}\text{-end}\n\text{-with-Surf-}\text{-do}\n\text{-classification}\_\text{of}\_\text{cubic}\_\text{surfaces}\_\text{with}\_\text{double}\_\text{sixes}\_\text{activity}\n\text{-recognize}\n\text{-recognize}\n\text{-q:49}\n\text{-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=1,\delta=2"\n"\Dalpha=2,\beta=1,\gamma=1,\delta=2\D"\n-end\n\text{-end}\n\text{-with-Surf-}\text{-do}\n\text{-classification}\_\text{of}\_\text{cubic}\_\text{surfaces}\_\text{with}\_\text{double}\_\text{sixes}\_\text{activity}\n\text{-recognize}\n\text{-recognize}\n\text{-q:49}\n\text{-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=2"\n"\Dalpha=2,\beta=1,\gamma=2,\delta=2\D"\n-end\n\text{-end}
-end\"  
-recognize\"  
-q-49\"  
-by_equation."F_alpha_beta_gamma_delta".\"  
"DF\{alpha, beta, gamma, delta\}D"."x0,x1,x2,x3".\"  
$(F_ALPHA_BETA_GAMMA_DELTA)\"  
"alpha=2, beta=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\".\"  
"alpha=2, beta=1, gamma=4, delta=3".\"  
"D\alpha=2, \beta=1, \gamma=4, \delta=3\".\"  

527
surf49.recognize:
$\{\text{ORBITER\ PATH}\}\ orbiter\ .out-v.3$

-define F-\finite_field-\q49-\end

-define P-\projective_space3-F-\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

-\by.coefficients"2,5,1,16,4,10,1,18,4,8,4,9,3,11,4,14"

-end

-end

-print_symbols

McKean_15lines_q7:

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

-define F-\finite_field-\q7-\end

-define P-\projective_space3-F-\end

-with P-do

-projective_space_activity

-define_surface_S-\by.coefficients\$(\text{SURFACE_MCKEAN_15_LINES})-\q7$

-end

-end

#pdf\latex\-surface_by\coefficients_q7_report.tex

#open\-surface_by\coefficients_q7_report.pdf

#2-Eckardt-points

F_4.4.3.3.q7:

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

-define F-\finite_field-\q7-\end

-define P-\projective_space3-F-\end

-with P-do

-projective_space_activity
-define_surface-q7-by_equation-
"F_alpha_beta_gamma_delta".
"\DF_\{\alpha,\beta,\gamma,\delta}\D".
"x_0,x_1,x_2,x_3".
$(F\_ALPHA\_BETA\_GAMMA\_DELTA)\$
"\alpha=4,\beta=4,\gamma=3,\delta=3\".
"\alpha=4,\beta=4,\gamma=3,\delta=3\D".
-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\_q7\_end-
-define-P\_projective_space\_3\_F\_end-
-with-P\_do-
-projective_space\_activity-
-sweep\_nothing-
-define_surface\_q7\_by_equation-
"F_alpha_beta_gamma_delta".
"\DF_\{\alpha,\beta,\gamma,\delta}\D".
"x_0,x_1,x_2,x_3".
$(F\_ALPHA\_BETA\_GAMMA\_DELTA)\$
"\alpha=2,\beta=3,\gamma=4,\delta=5\".
"\alpha=2,\beta=3,\gamma=4,\delta=5\D".
-end-
-end-

#Section 7.2: Cubic Surfaces and Quartic Curves
SECTION CUBIC SURFACES AND QUARTIC CURVES:
quartic_curve_9_0_report:
$(ORBITER\_PATH)orbiter.out\_v\_3$
-define-F\_finite_field\_q9\_end-
7011
7012
7013
7014
7015
7016  surface_4_0_quartic_curves:
7017  \$\textit{(ORBITER\_PATH)}\texttt{orbiter\_out\_\(-v\_3\)}\$
7018  \$\textit{define}\_F\_finite\_field\_q\_4\_\texttt{-end}\$
7019  \$\textit{define}\_P\_projective\_space\_3\_F\_\texttt{-end}\$
7020  \$\textit{with}\_P\_\texttt{-do}\$
7021  \$\textit{projective\_space\_activity}\$
7022  \$\textit{define}\_surface\_S4\_0\_q\_4\_\texttt{-catalogue}\_0\_\texttt{-end}\$
7023  \$\textit{-end}\$
7024  \$\textit{-with}\_S4\_0\_\texttt{-do}\$
7025  \$\textit{-cubic\_surface\_activity}\$
7026  \$\textit{-report}\$
7027  \$\textit{-report\_with\_group}\$
7028  \$\textit{-all\_quartic\_curves}\$
7029  \$\textit{-end}\$
7030  \$\textit{pdflatex}\_surface\_catalogue\_q4\_iso0\_report\.tex$
7031  \$\textit{open}\_surface\_catalogue\_q4\_iso0\_report\.pdf$
7032  \$\textit{pdflatex}\_surface\_catalogue\_q4\_iso0\_with\_group\.tex$
7033  \$\textit{open}\_surface\_catalogue\_q4\_iso0\_with\_group\.pdf$
7034  \$\#pdflatex\_surface\_catalogue\_q4\_iso0\_quartics\.tex$
7035  \$\#open\_surface\_catalogue\_q4\_iso0\_quartics\.pdf$
7036
7037
7038
7039  \#full\_del\_Pezzo\_surfaces:
7040
7041
7042  \textit{NB\_CUBIC\_SURFACES\_Q7}\_1
7043
7044  quartic\_curves\_q7:
7045  \$\textit{(ORBITER\_PATH)}\texttt{orbiter\_out\_\(-v\_3\)}\$
7046  \$\textit{list\_arguments}\$
7047  \$\textit{define}\_F\_finite\_field\_q\_7\_\texttt{-end}\$
7048  \$\textit{define}\_P\_projective\_space\_3\_F\_\texttt{-end}\$
7049  \$\textit{loop}\_L\_0\_\$\textit{(NB\_CUBIC\_SURFACES\_Q7)}\_1\$
7050  \$\textit{-with}\_P\_\texttt{-do}\$
7051  \$\textit{-projective\_space\_activity}\$
7052  \$\textit{-define}\_surface\_S\_%L\_q\_7\_\texttt{-catalogue}\_%L\_\texttt{-end}\$
7053  \$\textit{-end}\$
7054  \$\textit{-end\_loop}\$
7055  \$\textit{-print\_symbols}\$
7056  \$\textit{-loop}\_L\_0\_\$\textit{(NB\_CUBIC\_SURFACES\_Q7)}\_1\$
7057  \$\textit{-with}\_S\_%L\_\texttt{-do}\$

531
quartic_curves_q7_classify:

$\$(ORBITER\_PATH)orbiter.out\:-v.3\$

-defineF\-finite_field\-q.7\-end\$

-defineP\-projective_space\-2F\-end\$

-withP\-do\$

-projective_space_activity\$

-classify_quartic_curves_with_substructure\$

-surface_catalogue_q7_iso%d_quartics.csv\$

$\$(NB\_CUBIC\_SURFACES\_Q7)\-3\-4\-quartic\_curves\_q7\$

-end\$

-print_symbols\$


NB_CUBIC\_SURFACES\_Q13=4

quartic_curves_q13:

$\$(ORBITER\_PATH)orbiter.out\:-v.3\$

-defineF\-finite_field\-q.13\-end\$

-defineP\-projective_space\-3F\-end\$

-loopL.0\-$(NB\_CUBIC\_SURFACES\_Q13)\-1\$

-withP\-do\$

-projective_space_activity\$

-define\_surface\_S13\_\%L\_\:-q.13\:-catalogue\_\%L\:-end\$

-end\$

-print_symbols\$

-loopL.0\-$(NB\_CUBIC\_SURFACES\_Q13)\-1\$

-with\_S13\_\%L\:-do\$

-cubic\_surface\_activity\$

-export_all\_quartic\_curves\$

-end\$

-end_loop\$

-print_symbols\$

-loopL.0\-$(NB\_CUBIC\_SURFACES\_Q13)\-1\$

-cubic\_surface\_activity\$

-export_all\_quartic\_curves\$

-end\$

-end_loop\$

-print_symbols
quadric_curve_q13_classify:
$ $(ORBITER_PATH) orbiter.out -v -3 \n$ -list_arguments \n$ -define F -finite_field -q 13 -end \n$ -define P -projective_space -2 F -end \n$ -with P -do \n$ -projective_space_activity \n$ -classify_quadric_curve_with_substructure \n$ -surface_catalogue_q13_iso%d_quartics.csv \n$ -end \n$ -print_symbols \n
#q13
#The number of types of quadric curves is 2
#idx : ago
#0 : 24
#1 : 48

NB_CUBIC_SURFACES Q17 = 7
quadric_curve_q17:
$ $(ORBITER_PATH) orbiter.out -v -3 \n$ -list_arguments \n$ -define F -finite_field -q 17 -end \n$ -define P -projective_space -3 F -end \n$ -loop L 0 $(NB_CUBIC_SURFACES Q17) 1 \n$ -with P -do \n$ -projective_space_activity \n$ -define_surface_S17 %L -q 17 -catalogue %L -end \n$ -end \n$ -end_loop \n$ -print_symbols \n$ -loop L 0 $(NB_CUBIC_SURFACES Q17) 1 \n$ -with S17 %L -do \n$ -cubic_surface_activity \n$ -report \n
quartic_curves_q17.classify:

$\texttt{(ORBITER\_PATH)orbiter.out\ -v\ 3}$

-\ export\_all\_quartic\_curves\$

-\ end\$

-\ end\_loop\$

-\ print\_symbols

-\ #pdflatex\ surface\_catalogue\_q17\_iso0\_report.tex

-\ #open\ surface\_catalogue\_q17\_iso0.pdf

#q17

#The\ number\ of\ types\ of\ quartic\ curves\ is\ 7

#idx:: ago

#User\ time:: 2:33

#The\ number\ of\ types\ of\ quartic\ curves\ is\ 7

#idx:: ago

NB\ CUBIC\ SURFACES\ Q19=10

quartic_curves_q19:

$\texttt{(ORBITER\_PATH)orbiter.out\ -v\ 3}$

-\ 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\_q17\_iso%d\_quartics.csv$

-\ $(\texttt{NB\_CUBIC\_SURFACES\_Q19})\cdot 3\cdot 4\cdot \texttt{quartic\_curves\_q19}\$

-\ end\$

-\ print\_symbols\$

#User\ time:: 2:33
quartic_curves_q19.classify:

$\text{(ORBITER\_PATH)orbiter.out\texttt{\textquotesingle}-v.3\textquotesingle}$

-define:F\texttt{\textquotesingle}-finite_field\texttt{\textquotesingle}-q.19\texttt{\textquotesingle}-end\texttt{\textquotesingle}
-define:P\texttt{\textquotesingle}-projective_space\texttt{\textquotesingle}-2.F\texttt{\textquotesingle}-end\texttt{\textquotesingle}
-with:P\texttt{\textquotesingle}-do\texttt{\textquotesingle}

-projective_space_activity\texttt{\textquotesingle}

-surface_catalogue_q19_iso%d_quartics.csv$

-quartic_curves_q19_set_stabilizer:

$\text{(ORBITER\_PATH)orbiter.out\texttt{\textquotesingle}-v.3\textquotesingle}$

-define:F\texttt{\textquotesingle}-finite_field\texttt{\textquotesingle}-q.19\texttt{\textquotesingle}-end\texttt{\textquotesingle}
-define:P\texttt{\textquotesingle}-projective_space\texttt{\textquotesingle}-2.F\texttt{\textquotesingle}-end\texttt{\textquotesingle}
-with:P\texttt{\textquotesingle}-do\texttt{\textquotesingle}

-projective_space_activity\texttt{\textquotesingle}

-set_stabilizer4\texttt{\textquotesingle}

-surface_catalogue_q19_iso%d_quartics.csv$

-$\text{(NB\_CUBIC\_SURFACES\_Q19).4\cdot4\cdotquartic\_curves\_q19}\texttt{\textquotesingle}$

-end\texttt{\textquotesingle}

-writes:

-quartic_curves_q19_canonical_data.csv

-quartic_curves_q19_canonical.tex

-14\texttt{\textquotesingle}-isomorphism\texttt{\textquotesingle}-types:
-ago-dist::4\texttt{\textquotesingle}-1,9\texttt{\textquotesingle}-1,2\texttt{\textquotesingle}-4,6\texttt{\textquotesingle}-2,8\texttt{\textquotesingle}-3,24\texttt{\textquotesingle}-3

-quartic_curves_q19_set_stabilizer:

$\text{(ORBITER\_PATH)orbiter.out\texttt{\textquotesingle}-v.3\textquotesingle}$

-define:F\texttt{\textquotesingle}-finite_field\texttt{\textquotesingle}-q.19\texttt{\textquotesingle}-end\texttt{\textquotesingle}
-define:P\texttt{\textquotesingle}-projective_space\texttt{\textquotesingle}-2.F\texttt{\textquotesingle}-end\texttt{\textquotesingle}
-with:P\texttt{\textquotesingle}-do\texttt{\textquotesingle}

-projective_space_activity\texttt{\textquotesingle}

-set_stabilizer4\texttt{\textquotesingle}

-surface_catalogue_q19_iso%d_quartics.csv$

-$\text{(NB\_CUBIC\_SURFACES\_Q19).\textquotesingle}pts\_on\_curve\textquotesingle\texttt{\textquotesingle}$

-end\texttt{\textquotesingle}

-writes:

-quartic_curves_q19_canonical_data.csv

-quartic_curves_q19_canonical.tex

-14\texttt{\textquotesingle}-isomorphism\texttt{\textquotesingle}-types:
surface_13_0_quartics:
$\text{(ORBITER\_PATH)orbiter.out-\text{-v.3\text{"}}}$
\text{\textbackslash }
-define F finite_field-q13-end
-define P projective_space-3-F-end
-with P do
-projective_space_activity
-define_surface S13_0-q13-catalogue-0-end
-with S13_0 do
-cubic_surface_activity
-report
-export_all_quartic_curves
-end
\text{pdflatex surface\_catalogue\_q13\_iso0\_quartics.tex}
\text{open surface\_catalogue\_q13\_iso0\_quartics.pdf}

surface_13_1_quartics:
$\text{(ORBITER\_PATH)orbiter.out-\text{-v.3\text{"}}}$
-define F finite_field-q13-end
-define P projective_space-3-F-end
-with P do
-projective_space_activity
-define_surface S13_1-q13-catalogue-1-end
-with S13_1 do
-cubic_surface_activity
-report
-export_all_quartic_curves
-end
\text{pdflatex surface\_catalogue\_q13\_iso1\_quartics.tex}
\text{open surface\_catalogue\_q13\_iso1\_quartics.pdf}

quartic_curve_13_2_group:
$\text{(ORBITER\_PATH)orbiter.out-\text{-v.3\text{"}}}$
-define G linear_group-PGL3-13
-subgroup_by_generators "quartic_13_2" 
-24336-5
pdflatex PGL_3_13_Subgroup_quartic_13_2_24336_report.tex
open-PGL_3_13_Subgroup_quartic_13_2_24336_report.pdf
#pdflatex PGGL_3_4_report.tex
#open PGGL_3_4_report.pdf

Knecht_13_1:
$\$(ORBITER_PATH)orbiter.out-v:2$
 -define\-F\-finite_field\-q:13\-end
 -define\-P\-projective_space\-2\-F\-end
 -with\-P\-do
 -projective_space\activity
 -define_object\-Knecht_13_1
 -define\-projective\-variety\"Knecht_1\"\-4\-\$(KNECHT_13_1\_AS\_VECTOR)\end''
 -monomial\_type\PART
 -end
 -end
 -with\-Knecht_13_1\-do
 -combinatorial\_object\activity
 -save
 -end
 -with\-Knecht_13_1\-do
 -combinatorial\_object\activity
 -conic\_type\6
 -end
 -print\_symbols
Knecht_13_2:
$\text{VECTOR(}\text{orbiter.out-v.2}}$
$\text{-define-F\text{-finite_field-q13-end}}$
$\text{-define-P\text{-projective_space-2F-end}}$
$\text{-with-P\text{-do}}$
$\text{-projective_space_activity}$
$\text{-define_object\text{-Knecht_13_2}}$
$\text{-projective_variety\text{-Knecht_2\text{-4\text{-KNECHT_13_2\text{-AS\_VECTOR}}}}}$
$\text{-monomial_type\text{-PART}}$
$\text{-end}$
$\text{-do}$
$\text{-combinatorial_object_activity}$
$\text{-save}$
$\text{-end}$

Nathan_19_1:
$\text{VECTOR(}\text{orbiter.out-v.2}}$
$\text{-define-F\text{-finite_field-q19-end}}$
$\text{-define-P\text{-projective_space-2F-end}}$
$\text{-with-P\text{-do}}$
$\text{-projective_space_activity}$
$\text{-define_object\text{-N19_1}}$
$\text{-projective_variety\text{-Nathan19_1\text{-4}}}$
$\text{16,0,9,3,10,4,11,9,14,12,9,5,4,14,3,7,16,1,4,6,17,11,17,8,6,2}$
$\text{-monomial_type\text{-PART}}$
$\text{-end}$
$\text{-do}$
$\text{-combinatorial_object_activity\text{-save}}$
$\text{-end}$

surface_25_0:
$\text{VECTOR(}\text{orbiter.out-v.3}}$
$\text{-define-F\text{-finite_field-q25-end}}$
$\text{-define-P\text{-projective_space-3F-end}}$
$\text{-with-P\text{-do}}$
$\text{-projective_space_activity}$
$\text{-define_surface\text{-S25_0\text{-25\text{-catalogue\text{-0\text{-end}}}}}]}$
$\text{-end}$
$\text{-do}$
$\text{-cubic_surface_activity}$
$\text{-report}$
quartic_curve_25_report:\n$\$(ORBITER_PATH)orbiter.out\-v.3\$
-define F-finite_field-q.25-end\n-define F-projective_space-2-F-end\n-loop L-0.18-1\n-define_quartic_curve QC25_%L-q.25-catalogue_%L-end\n-end_loop\n-print_symbols\n-loop L-0.18-1\n-with QC25_%L-do\n-quartic_curve_activity\n-report\n-end\n-end_loop\n-print_symbols
pdflatex-quartic_curve_catalogue_q25_iso0_report.tex
pdflatex-quartic_curve_catalogue_q25_iso1_report.tex
pdflatex-quartic_curve_catalogue_q25_iso2_report.tex
pdflatex-quartic_curve_catalogue_q25_iso3_report.tex
pdflatex-quartic_curve_catalogue_q25_iso4_report.tex
pdflatex-quartic_curve_catalogue_q25_iso5_report.tex
pdflatex-quartic_curve_catalogue_q25_iso6_report.tex
pdflatex-quartic_curve_catalogue_q25_iso7_report.tex
pdflatex-quartic_curve_catalogue_q25_iso8_report.tex
pdflatex-quartic_curve_catalogue_q25_iso9_report.tex
pdflatex-quartic_curve_catalogue_q25_iso10_report.tex
pdflatex-quartic_curve_catalogue_q25_iso11_report.tex
pdflatex-quartic_curve_catalogue_q25_iso12_report.tex
pdflatex-quartic_curve_catalogue_q25_iso13_report.tex
pdflatex-quartic_curve_catalogue_q25_iso14_report.tex
pdflatex-quartic_curve_catalogue_q25_iso15_report.tex
pdflatex-quartic_curve_catalogue_q25_iso16_report.tex
pdflatex-quartic_curve_catalogue_q25_iso17_report.tex
gs -sDEVICE=pdfwrite -r120 -o quartic_curve_catalogue_q25.pdf "quartic_curve_catalogue_q25_iso0_report.pdf:" "quartic_curve_catalogue_q25_iso1_report.pdf:"
quartic_curve_catalogue_q25_iso0_report.pdf

quartic_curve_catalogue_q25_iso1_report.pdf

quartic_curve_catalogue_q25_iso2_report.pdf

quartic_curve_catalogue_q25_iso3_report.pdf

quartic_curve_catalogue_q25_iso4_report.pdf

quartic_curve_catalogue_q25_iso5_report.pdf

quartic_curve_catalogue_q25_iso6_report.pdf

quartic_curve_catalogue_q25_iso7_report.pdf

quartic_curve_catalogue_q25_iso8_report.pdf

quartic_curve_catalogue_q25_iso9_report.pdf

quartic_curve_catalogue_q25_iso10_report.pdf

quartic_curve_catalogue_q25_iso11_report.pdf

quartic_curve_catalogue_q25_iso12_report.pdf

quartic_curve_catalogue_q25_iso13_report.pdf

quartic_curve_catalogue_q25_iso14_report.pdf

quartic_curve_catalogue_q25_iso15_report.pdf

quartic_curve_catalogue_q25_iso16_report.pdf

quartic_curve_catalogue_q25_iso17_report.pdf

quartic_curve_catalogue_q25_iso18_report.pdf

#open quartic_curve_catalogue_q25_iso0_report.pdf

quartic_curve_13_table:

$\$(\text{ORBITER\_PATH})\text{orbiter\_out-\text{-}v\_3}\$

-define F\_finite_field\_q\_13\_end

-define P\_projective_space\_2\_F\_end

-with P\_do

-projective_space_activity

-table_of_quartic_curves

-end

#quartic_curves_q13_info.csv

quartic_curve_19_table:

$\$(\text{ORBITER\_PATH})\text{orbiter\_out-\text{-}v\_3}\$

-define F\_finite_field\_q\_19\_end

-define P\_projective_space\_2\_F\_end

-with P\_do

-projective_space_activity

-table_of_quartic_curves

-end

quartic_curve_19_table_latex:

$\$(\text{ORBITER\_PATH})\text{orbiter\_out-\text{-}v\_3}\$

-csv_file_latex\_1\_quartic\_curves\_q19\_info.csv

~\text{/bin/tth-quartic\_curves\_q19\_info.tex}
quartic_curve_25_table:

$(ORBITER_PATH)orbiter.out-v.3\$

-define-F-finite_field-q 25-end\$

-define-P-projective_space-2 F-end\$

-with-P-do\$

-projective_space_activity\$

-table_of_quartic_curves\$

-end

#quartic_curves_q25_info.csv

quartic_curve_27_table:

$(ORBITER_PATH)orbiter.out-v.3\$

-define-F-finite_field-q 27-end\$

-define-P-projective_space-2 F-end\$

-with-P-do\$

-projective_space_activity\$

-table_of_quartic_curves\$

-end

#quartic_curves_q27_info.csv

quartic_curve_29_table:

$(ORBITER_PATH)orbiter.out-v.3\$

-define-F-finite_field-q 29-end\$

-define-P-projective_space-2 F-end\$

-with-P-do\$

-projective_space_activity\$

-table_of_quartic_curves\$

-end

#quartic_curves_q29_info.csv

quartic_curve_31_table:

$(ORBITER_PATH)orbiter.out-v.3\$

-define-F-finite_field-q 31-end\$

-define-P-projective_space-2 F-end\$

-with-P-do\$

-projective_space_activity\$

-table_of_quartic_curves\$

-end
surface_25_12:

$\text{define } F \text{ finite field } q^{25} \text{ end}$

$\text{define P projective space 3 F end}$

$\text{end}$

$\text{define projective space activity}$

$\text{define surface S25_12 q^{25} catalogue 12 end}$

$\text{end}$

$\text{define surface activity}$

$\text{define surface S25_12 q^{25} catalogue 12 end}$

$\text{end}$

$\text{transform } "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"$.

$\text{end}$

$\text{end}$

$\text{define surface activity}$

$\text{define surface S25_12 q^{25} catalogue 12 end}$

$\text{end}$

$\text{transform } "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"$.

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$

$\text{define surface activity}$

$\text{define surface S25_12 q^{25} catalogue 12 end}$

$\text{end}$

$\text{transform } "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"$.

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$

$\text{define surface activity}$

$\text{define surface S25_12 q^{25} catalogue 12 end}$

$\text{end}$

$\text{transform } "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"$.

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$

$\text{define surface activity}$

$\text{define surface S25_12 q^{25} catalogue 12 end}$

$\text{end}$

$\text{transform } "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"$.

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$

$\text{end}$
surface 25_12_t3:
$\text{(ORBITER PATH)}\text{orbiter.out -v:3}$\n-define F -finite_field -q:25 -end\n-define P -projective_space 3 F -end\n-with P -do\n-projective_space_activity\n-define_surface S25_12 -q:25 -catalogue 12\n-transform "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,0"\n-transform_inverse "16,0,1,0,-3,5,1,0,0,0,1,0,0,0,1,0,0"\n-end\n-end\n-with S25_12 -do\n-cubic_surface_activity\n-report\n-report_with_group\n-end
pdflatex surface_catalogue_q25_iso12_with_group.tex
open_surface_catalogue_q25_iso12_with_group.pdf

surface 25_12_t4:
$\text{(ORBITER PATH)}\text{orbiter.out -v:3}$\n-define F -finite_field -q:25 -end\n-define P -projective_space 3 F -end\n-with P -do\n-projective_space_activity\n-define_surface S25_12 -q:25 -catalogue 12\n-transform "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,0"\n-transform_inverse "16,0,1,0,-3,5,1,0,0,0,1,0,0,0,1,0,0"\n-end\n-end\n-with S25_12 -do\n-cubic_surface_activity\n-report\n-report_with_group\n-end
pdflatex surface_catalogue_q25_iso12_with_group.tex
open_surface_catalogue_q25_iso12_with_group.pdf
-transform "3,0,0,0,-0,1,0,0,-0,0,1,0,0,0,1,-0"
-define 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

7637 $ (ORBITER_PATH) orbiter.out -v 3

7638 -define F -finite_field -q 25 -end

7639 -define P -projective_space 3 F -end

7640 -with P -do

7641 -define_surface S25_12 -q 25 -catalogue 12
-define_surface -projective_space_activity
-end

7642 -transform "1,0,0,16,-0,1,0,18,0,0,1,0,1,0,-0,0,1,-0"
-define_surface -transform_inverse "16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,0,1,0"

7644 -define_surface -transform_inverse "3,0,0,0,-0,1,0,0,-0,0,1,0,0,0,1,-0"
-define_surface -transform_inverse "1,0,0,0,0,1,0,0,-0,0,1,0,0,-0,0,1,0"

7647 -define_surface -transform_inverse "1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0"

7648 -end

7651 -cubic_surface_activity
-report
-report_with_group
-end

7655 open_surface_catalogue_q25_iso12_with_group.pdf

7658 PG_2.25:

PG_2.25_lines:
-subgroup_by_generators: "triangle_sta"-6912.7\n-"1,0,0,0,1,0,0,0,1,1,1,\n1,0,0,0,13,0,0,0,13,1,\n1,0,0,0,4,0,0,0,6,1,\n1,0,0,0,17,0,0,0,13,0,\n1,0,0,0,18,0,0,0,4,1,\n1,0,0,0,11,0,1,0,0,\n0,1,0,0,20,14,0,0,0"\n-end:\n-with-\G-do-\n-group_theoretic_activity-\n-poset_classification_control-\n-problem_label-PGGL_3_25-\n-depth-3-draw_poset-draw_options-radius-200-end-\n-recognize."8,44,226".\n-end-\n-orbits_on_subsets-3-\n-end-\n
surface_25_12_t7:$(ORBITER_PATH)orbiter.out-v.3-\n-define-F-finite_field-q-25-end-\n-define-P-projective_space-3F-end-\n-with-P-do-\n-projective_space_activity-\n-define_surface-S25_12-q-25-catalogue-12-\n-transform-"1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,0"-\n-transform_inverse-"16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,0,1,0"-\n-transform_inverse-"3,0,0,0,-0,1,0,0,-0,0,1,0,0,0,1,0,-0\n-transform_inverse-"1,0,0,0,-0,1,0,0,-0,0,1,0,13,2,2,1,0"-\n-transform_inverse-"1,0,0,0,-0,1,0,0,-0,0,1,0,13,1,1,0,-0,0,0,1,0"-\n-transform_inverse-"3,8,8,0,22,13,22,0,14,19,15,0,0,0,0,1,1,1"-\n-transform_inverse-"16,0,0,0,-0,16,0,0,21,21,21,0,-0,0,0,1,0,0"-\n-transform-"1,0,0,0,-0,5,0,0,-0,0,17,0,-0,0,0,1,1,1"-\n-end-\n-end-\n-with-S25_12-do-\ncubic_surface_activity-\n-report-\n-report_with_group-\n-end-\n-pdflatex:surface_catalogue_q25_isol2_with_group.tex-\n-open:surface_catalogue_q25_isol2_with_group.pdf
7763  surface_25_12_t8:
7765  > $(ORBITER_PATH)orbiter.out-v.3\n7766  > define F=finite field-q.25-end\n7767  > define P=projective space 3 F-end\n7768  > with P-do\n7769  > > -projective space activity\n7770  > > > define surface S25_12=q.25-catalogue 12\n7771  > > > transform "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"\n7772  > > > transform_inverse "16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,1,0,-0,0,1,0\n7773  > > > transform "3,0,0,0,0,1,0,0,-0,0,1,0,0,0,1,-0"\n7774  > > > transform_inverse "1,0,0,0,0,1,0,0,-0,0,1,0,0,0,1,0\n7775  > > > transform_inverse "1,0,0,0,0,1,0,0,-0,0,1,0,0,0,1,0\n7776  > > > transform "3,8,8,0,22,13,22,0,14,19,15,0,0,0,1,1\n7777  > > > transform_inverse "16,0,0,0,0,-0,16,0,0,21,21,21,0,-0,0,0,1,0\n7778  > > > transform "1,0,0,0,0,5,0,0,-0,0,17,0,-0,0,1,1\n7779  > > > transform "1,0,0,0,0,1,0,0,-0,0,1,0,0,0,24,0\n7780  > > > -end\n7781  > > -end\n7782  > > -with S25_12-do\n7783  > > cubic_surface activity\n7784  > > -report\n7785  > > -report_with_group\n7786  > > -end
7787  > pdflatex surface catalogue q25_isol2 with group.tex
7788  > open surface catalogue q25_isol2 with group.pdf
7789
7790
7791  surface_25_12_t9:
7792  > $(ORBITER_PATH)orbiter.out-v.3\n7793  > define F=finite field-q.25-end\n7794  > define P=projective space 3 F-end\n7795  > with P-do\n7796  > > -projective space activity\n7797  > > > define surface S25_12=q.25-catalogue 12\n7798  > > > transform "1,0,0,16,-0,1,0,18,-0,0,1,8,-0,0,1,1,-0"\n7799  > > > transform_inverse "16,0,1,0,-3,5,1,0,0,0,1,0,-0,0,1,0\n7800  > > > transform "3,0,0,0,0,1,0,0,-0,0,1,0,0,0,1,-0"\n7801  > > > transform_inverse "1,0,0,0,0,1,0,0,-0,0,1,0,0,0,1,13,2,2,1,0\n7802  > > > transform_inverse "1,0,0,0,0,1,0,0,-0,0,13,1,0,0,0,1,0\n7803  > > > transform "3,8,8,0,22,13,22,0,14,19,15,0,0,0,1,1\n7804  > > > transform_inverse "16,0,0,0,0,-0,16,0,0,21,21,21,0,-0,0,0,1,0\n7805  > > > transform "1,0,0,0,0,5,0,0,-0,0,17,0,-0,0,1,1\n7806  > > > transform "1,0,0,0,0,1,0,0,-0,0,1,0,0,0,24,0\n7807  > > > -end\n7808  > > -end\n7809  > > -end\n
547
surface_25_12_t8_quartic_curves:
$\$(\text{ORBITER\_PATH})\text{orbiter.out\_v.3}\$
-define\_F\_finite\_field\_q\_25\_end\$
-define\_P\_projective\_space\_3\_F\_end\$
-define\_S25\_12\_q\_25\_catalogue\_12\$
-transform\_"1,0,0,16,0,1,0,18,0,0,0,1,1,0\"\$
-transform\_inverse\_"16,0,1,0,3,5,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0\"\$
-transform\_"3,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0\"\$
-transform\_inverse\_"1,0,0,0,0,1,0,0,0,1,0,0,1,0,1,0,3,2,2,1,0\"\$
-transform\_inverse\_"1,0,0,0,0,1,0,0,0,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0\"\$
-transform\_"3,8,8,0,22,13,22,0,14,19,15,0,0,0,0,1,0,\$
-transform\_inverse\_"16,0,0,0,0,16,0,0,21,21,21,0,0,0,0,0,1,0,1,0\"\$
-transform\_inverse\_"1,0,0,0,0,5,0,0,0,0,17,0,0,0,0,0,1,0,1,0,\$
-transform\_"1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,24,0\"\$
-end\$
-with\_S25\_12\_do\$
-cubic\_surface\_activity\$
-all\_quartic\_curves\$
-end\$
-with\_S25\_12\_do\$
-cubic\_surface\_activity\$
-end\$
-with\_S25\_12\_do\$
-cubic\_surface\_activity\$
-export\_all\_quartic\_curves\$
-end
pdflatex\_surface\_catalogue\_q25\_iso12\_quartics.tex
open\_surface\_catalogue\_q25\_iso12\_quartics.pdf

quartic\_curve\_13\_0\_surface:\$
$\$(\text{ORBITER\_PATH})\text{orbiter.out\_v.3}\$
-define\_F\_finite\_field\_q\_13\_end\$
-define\_P\_projective\_space\_2\_F\_end\$
-with\_P\_do\$

quartic\_curve\_13\_0\_surface:
-projective_space_activity\ 
.define_quartic_curve:QC13_0-q.13--catalogue-0--end\ 
.end\ 
.with:QC13_0-do\ 
.quartic_curve_activity\ 
.create_surface\ 
.end\ 
#surface_equation: -0,-0,9,-0,-3,0,1,-0,12,-7,0,-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:QC13_0_surface_create:

$\texttt{(ORBITER\ PATH)orbiter.out}\ -v\ 5\ 
.define:F\ -finite_field\ -q.13--end\ 
.define:P\ -projective_space:3F--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\ 

.report\ 

#report\ with\ group\ 

#############################################################################
#############################################################################
# Section: 7.3: Cubic Surfaces Classification

SECTION_CUBIC_SURFACES_CLASSIFICATION:

.surface_classify_q4:

$\texttt{(ORBITER\ PATH)orbiter.out}\ -v\ 5\ 
.define:F\ -finite_field\ -q.4--end\ 

surface_classify_q4.arc_lifting_two_lines:

surface_classify_q7:
7950
7951
7952 surface_classify_q13:
7953 $$(\text{ORBITER \ PATH})\text{orbiter.out}\cdot-v:\5\backslash$
7954 $\backslash$-define-F\-finite_field-q\cdot13\-end\backslash$
7955 $\backslash$-define-P\-projective_space\-3\-F\-end\backslash$
7956 $\backslash$-with-P\-do\backslash$
7957 $\backslash$-projective_space_activity\backslash$
7958 $\backslash$-define\-finite\-field\-q\-13\-end\backslash$
7959 $\backslash$-define\-projective_space\-3\-F\-end\backslash$
7960 $\backslash$-with\-P\-do\backslash$
7961 $\backslash$-projective_space_activity\backslash$
7962 $\backslash$-classify\-surfaces\-with\-double\-sixes\-C\-W\-end\backslash$
7963 $\backslash$-end\backslash$
7964 $\backslash$-report\-end\backslash$
7965 $\backslash$-print\_symbols
7966 $\backslash$pdflatex\-Surfaces\_q13.tex
7967 $\backslash$open\-Surfaces\_q13.pdf
7968
7969
7970 # Section 7.4: Cubic Surfaces - Isomorphism Testing and Recognition
7971
7972
7973
7974 SECTION_CUBIC_SURFACES_ISOMORPHISM_TESTING_AND_RECOGNITION:
7975
7976
7977 surface_recognize_q7_abcd_2_3_3_4:
7978 $$(\text{ORBITER \ PATH})\text{orbiter.out}\cdot-v:\3\backslash$
7979 $\backslash$-define-F\-finite_field-q\cdot7\-end\backslash$
7980 $\backslash$-define-P\-projective_space\-3\-F\-end\backslash$
7981 $\backslash$-with-P\-do\backslash$
7982 $\backslash$-projective_space_activity\backslash$
7983 $\backslash$-classify\-surfaces\-with\-double\-sixes\-Surf\-W\-end\backslash$
7984 $\backslash$-end\backslash$
7985 $\backslash$-with\-Surf\-do\backslash$
7986 $\backslash$-classification\_of\_cubic\_surfaces\_with\_double\_sixes\_activity\backslash$
7987 $\backslash$-recognize\backslash$
7988 $\backslash$-q\cdot7\backslash$
7989 $\backslash$-family\_general\_abcd\cdot2\cdot3\cdot3\cdot4\backslash$
7990 $\backslash$-end\backslash$
7991 $\backslash$-end\backslash$
7992 $\backslash$-end
7993
7994
7995 surface_isomorph_16:
$\$(\text{ORBITER_PATH})\text{orbiter.out} -v.3$

-define $F$ finite_field -q 16 -end
-define $P$ projective_space 3 $F$ -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:
-q 16 -by_coefficients:
"1,5,1,8,1,9,1,10,1,11,1,12,6,14,6,15,7,18,7,19" -end
-q 16 -by_coefficients:
"13,6,3,8,3,11,13,13,1,19" -end
-end
-end

-with Surf27 -do
-classification_of_cubic_surfaces_with_double_sixes_activity
-recognize:
-q 8 -by_coefficients: "1,6,1,8,1,11,1,13,1,19" -end
-end
-end
-print_symbols

surface_recognize_S:

surface_recognize_F13_q4:
Section 7.5: Cubic Surfaces of Dickson Type

SECTION_CUBIC_SURFACES_DICKSON:

D6.q2:

```bash
$ (ORBITER_PATH) orbiter.out -v 3
```

```bash
define F - finite_field - q 2 - end
```

```bash
define P - projective_space - 3 F - end
```

```bash
with P - do
```

```bash
classify_surfaces_with_double_sixes Surf27 - W - end
```

```bash
end
```

```bash
with Surf27 - do
```

```bash
classification_of_cubic_surfaces_with_double_sixes activity
```

```bash
identify F13
```

```bash
end
```

```bash
print_symbols
```

```bash
surface_sweep Cayley_13:
```

```bash
$ (ORBITER_PATH) orbiter.out -v 3
```

```bash
define F - finite_field - q 13 - end
```

```bash
define P - projective_space - 3 F - end
```

```bash
with P - do
```

```bash
projective_space_activity
```

```bash
classify_surfaces_with_double_sixes Surf27 - W - end
```

```bash
end
```

```bash
with Surf27 - do
```

```bash
classification_of_cubic_surfaces_with_double_sixes_activity
```

```bash
sweep
```

```bash
end
```

```bash
print_symbols
```

```bash
553
```
projective space activity
-define_surface:S_D6.q2-\(q^2\)-by_coefficients:$(D6)$-end\
-end\
-with:S_D6.q2-do\
cubic_surface_activity\
-report\
-end
-pdflatex:surface_by_coefficients_q2_report.tex
-open:surface_by_coefficients_q2_report.pdf
-mv:surface_by_coefficients_q2_points.txt:surface_by_coefficients_q2_D6_points.txt

1-line-over-GF(2)
D3.q4:
$(\text{ORBITER\_PATH})$\text{orbiter.out-v.3}$
-define:F-finite_field-q.4-end
-define:P-projective_space.3-F-end
-with:P-do
-projective_space_activity
-report
-end
-with:S_D3.q4-do
-cubic_surface_activity
-report
-end
-pdflatex:surface_by_coefficients_q4_report.tex
-open:surface_by_coefficients_q4_report.pdf
-mv:surface_by_coefficients_q4_points.txt:surface_by_coefficients_q4_D3_points.txt

#surface_by_coefficients_q4_points.txt

D4.q8:
$(\text{ORBITER\_PATH})$\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
-report
-end

554
D6_q4:

```
D6_q4: $(ORBITER_PATH)orbiter.out -v 3
define-F finite_field -q 4 -end
define-P projective_space 3 F -end
with-P do
projective_space activity
define_surface S_D6_q4 -q 4 -by_coefficients $(D6) -end
end
with S_D6_q4 do
-cubic_surface_activity
-report
end
```

# D6 has 7 lines over GF(4)

D8_q4:

```
D8_q4: $(ORBITER_PATH)orbiter.out -v 3
define-F finite_field -q 4 -end
define-P projective_space 3 F -end
with-P do
projective_space activity
define_surface S_D8_q4 -q 4 -by_coefficients $(D8) -end
end
with S_D8_q4 do
-cubic_surface_activity
-report
```
D1_q8:

D1_q4_with_select_double_six:

#surface_by_coefficients_q8_points.txt

##cleaning D1 · with 15 · lines · over · F2 · and 27 · lines · over · F4:

mv · surface_by_coefficients_q4_points.txt · surface_by_coefficients_q4_D8_points.txt
D1_q4_with_select_double_six_b:

```bash
$ (ORBITER_PATH) orbiter.out --v 3

-define F -finite_field -q 4 -end:
-define P -projective_space 3 F -end:
-with P -do
-projective_space_activity:

-define_surface S D1 q4 -q 4 -by_coefficients $(D1):
-select_double_six "3, 9, 15, 19, 22, 26, 4, 10, 14, 18, 21, 25":
-select_double_six "1, 2, 3, 4, 5, 0, 7, 8, 9, 10, 11, 6":
-end:

-with S D1 q4 -do:
-cubic_surface_activity:
-report:
-end:

mv surface_by_coefficients q4_report.tex D1_q4.tex
pdflatex D1_q4.tex
open D1_q4.pdf
```

D1_q4_trans:

```bash
$ (ORBITER_PATH) orbiter.out --v 5 -define F -finite_field -q 4 -end:

-with F -do -finite_field_activity:

-move_two_lines_in_hyperplane_stabilizer_text:

"1, 0, 0, 0, 0, 0, 1, 1", "0, 1, 1, 1, 1, 0, 1, 0",

"1, 0, 0, 0, 0, 0, 1", "0, 1, 0, 1, 1, 0, 1, 0",

-end:

-with S D1 q4 -do:
-cubic_surface_activity:
-report:
-end:

mv surface_by_coefficients q4_report.tex D1_q4.tex
pdflatex D1_q4.tex
```

D1_q4_with_select_double_six_c:

```bash
$ (ORBITER_PATH) orbiter.out --v 3

-define F -finite_field -q 4 -end:
-define P -projective_space 3 F -end:
-with P -do
-projective_space_activity:

-define_surface S D1 q4 -q 4 -by_coefficients $(D1):
-select_double_six "3, 9, 15, 19, 22, 26, 4, 10, 14, 18, 21, 25":
-select_double_six "1, 2, 3, 4, 5, 0, 7, 8, 9, 10, 11, 6":
-transform "1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0":
-end:

-end:

-with S D1 q4 -do:
-cubic_surface_activity:
-report:
-end:

mv surface_by_coefficients q4_report.tex D1_q4.tex
pdflatex D1_q4.tex
```
orbits_cubic_surfaces_q3:
$(ORBITER_PATH)orbiter.out-v.4-
-define-G-linear_group-PGL.4.3-end-
-with-G-do-
-group_theoretic_activity-
orbits_on_polynomials.3-
-end.
pdflatex-poly_orbits_d3_n3_q3.tex
open-poly_orbits_d3_n3_q3.pdf

#this.takes.3.days.and.about.150.GB.memory.on.ripoff

orbits_cubic_curves_q2_again:
$(ORBITER_PATH)orbiter.out-v.4-
-define-G-
-linear_group-PGL.3.2-
-end-
-with-G-do-
-group_theoretic_activity-
orbits_on_polynomials.3-
-end.
pdflatex-poly_orbits_d3_n2_q2.tex
open-poly_orbits_d3_n2_q2.pdf

orbits_cubic_curves_q3:
$(ORBITER_PATH)orbiter.out-v.4-
-define-G-
-linear_group-PGL.3.3-
-end-
-with-G-do-
-group_theoretic_activity-
orbits_on_polynomials.3-
-end.
pdflatex-poly_orbits_d3_n2_q3.tex
open-poly_orbits_d3_n2_q3.pdf

#compute.and.analyze.properties.over.F2
poly_orbits_d3_n3_q2_F2.csv: poly_orbits_d3_n3_q2.csv

> $(ORBITER_PATH)orbiter.out -v 4
> -define F:finite_field: q2 -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_q2_analyze: poly_orbits_d3_n3_q2_F2.csv

> $(ORBITER_PATH)orbiter.out -v 4
> -define F:finite_field: q2 -end
> -define P:projective_space: 3 F: end
> -with P: do
> -projective_space_activity
> -cubic_surface_properties_analyze
> > poly_orbits_d3_n3_q2_F2.csv: 2
> -end

> pdf_latex poly_orbits_d3_n3_q2_F2_report.tex
> open poly_orbits_d3_n3_q2_F2_report.pdf

# compute and analyze properties over F4

poly_orbits_d3_n3_q2_F4.csv: poly_orbits_d3_n3_q2.csv

> $(ORBITER_PATH)orbiter.out -v 4
> -define F:finite_field: q4 -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_F4.csv: 2.0
> -end

Dickson_q4_analyze: poly_orbits_d3_n3_q2_F4.csv

> $(ORBITER_PATH)orbiter.out -v 4
> -define F:finite_field: q4 -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
> -end

> pdf_latex poly_orbits_d3_n3_q2_F4_report.tex
> open poly_orbits_d3_n3_q2_F4_report.pdf
# compute-and-analyze-properties over $F_8$

```
poly_orbits_d3_n3_q2_F8.csv:poly_orbits_d3_n3_q2.csv
```

```
$\$(\text{ORBITER PATH})/\text{orbiter.out}.v-4$
```

```
-define $F$-finite_field-$q=8$-end\`
-define $P$-projective_space-3 $F$-end\`
-define $Q$-with $P$-do\`
-define $P$-projective_space_activity\`
```

```
poly_orbits_d3_n3_q2.csv.20
```

```
Dickson_q8.analyze:poly_orbits_d3_n3_q2_F8.csv
```

```
$\$(\text{ORBITER PATH})/\text{orbiter.out}.v-4$
```

```
-define $F$-finite_field-$q=8$-end\`
-define $P$-projective_space-3 $F$-end\`
-define $Q$-with $P$-do\`
-define $P$-projective_space_activity\`
```

```
poly_orbits_d3_n3_q2_F8.csv.20
```

```
Dickson_q16.analyze:poly_orbits_d3_n3_q2_F16.csv
```

```
$\$(\text{ORBITER PATH})/\text{orbiter.out}.v-4$
```

```
-define $F$-finite_field-$q=16$-end\`
-define $P$-projective_space-3 $F$-end\`
-define $Q$-with $P$-do\`
-define $P$-projective_space_activity\`
```

```
poly_orbits_d3_n3_q2_F16.csv.20
```

```
pdflatex poly_orbits_d3_n3_q2_F8_report.tex
```

```
open poly_orbits_d3_n3_q2_F8_report.pdf
```

```
Dickson_q8.analyze:poly_orbits_d3_n3_q2_F16.csv
```

```
pdflatex poly_orbits_d3_n3_q2_F16_report.tex
```

```
open poly_orbits_d3_n3_q2_F16_report.pdf
```

```
```

Section 7.6: Cubic Surfaces

SECTION_CUBIC_SURFACES_ATLAS_AND_TABLES:

MAKE_TABLE_OF_CUBIC_SURFACES=-define P -projective_space 3: F -end \n与 P-do\n-projective_space_activity\n-table_of_cubic_surfaces\n-end

$cubic_surfaces.tables.17:\$
$ORBITER_PATH$oriter.out-v.3\n-define F -finite_field -q 17-end\n$(MAKE_TABLE_OF_CUBIC_SURFACES)

$cubic_surfaces.tables.latex.17:\$
$ORBITER_PATH$oriter.out-v.3-csv_file_latex.1\n-table_of_cubic_surfaces_q17.info.csv

$cubic_surfaces.tables.up_to.17:\$
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 4-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 7-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 8-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 9-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 11-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 13-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 16-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
$ORBITER_PATH$oriter.out-v.3-define F -finite_field -q 17-end$(MAKE_TABLE_OF_CUBIC_SURFACES)
cubic_surfaces.tables.19.37:

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.19--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.23--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.25--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.27--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.29--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.31--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.32--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.37--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

cubic_surfaces.tables.41.and.up:

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.41--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.43--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.47--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.49--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.53--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.59--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.61--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.64--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.67--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.71--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.73--end$(MAKE_TABLE_OF_CUBIC_SURFACES)

$(ORBITER_PATH)orbiter.out-v.3--defineF--finite_field--q.79--end$(MAKE_TABLE_OF_CUBIC_SURFACES)
\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-81.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-83.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-89.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-97.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-101.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-103.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-107.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-109.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-113.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-121.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-127.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

\$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-define-F--finite_field--q-128.-end}\$(\text{MAKE\_TABLE \_OF\_CUBIC\_SURFACES})$

8490

8491 cubic\_surfaces\_tables\_latex::

8492 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_1.test.csv}$

8493 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q4.info.csv}$

8494 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q7.info.csv}$

8495 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q8.info.csv}$

8496 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q9.info.csv}$

8497 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q11.info.csv}$

8498 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q13.info.csv}$

8499 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q16.info.csv}$

8500 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q17.info.csv}$

8501 \$\text{(ORBITER\_PATH)}\text{orbiter.out-v.3.-csv\_file\_latex\_0.table\_of\_cubic\_surfaces\_q19.info.csv}$

8502
cubic_surfaces_tables_latex_big:

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q23\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q25\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q27\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q29\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q31\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q32\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q37\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q41\_info.csv

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -csv\_file\_latex\_0\-table\_of\_cubic\_surfaces\_q43\_info.csv

#$(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

surface_table:

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -make\_table\_of\_surfaces

pdflatex-surfaces_report.tex

open-surfaces_report.pdf

surface_atlas:

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -create\_surface\_atlas\_97

~/bin/tth-surface_atlas.tex

surface_reports:

$(ORBITER_PATH)orbiter.out\ -v\ 3\ -orbiter-path$(ORBITER_PATH)-create_surface_reports\_4,7,8,9,11
8542
8543
8544
8545
8546
8547
8548 Surface_Orb0_q4:
8549  \$\text{(ORBITER~PATH)} orbiter.out -v 3 \$
8550  \>$linear\_group -PGL\_4\_4 -wedge -end$
8551  \>$group\_theoretic\_activity$
8552  \>$control\_six\_arcs -end$
8553  \>$define\_surface -label\_txt "Orb0\_q4" -label\_tex "Rank-$\text{(Orb0)}$" -by_rank-$\text{(Orb0)}$ -2 -q 4$
8554  \>$end$
8555  \>$end$
8556
8557
8558
8559
8560
8561
8562
8563
8564 create_dickson_makefiles_q2:
8565  \$\text{(ORBITER~PATH)} orbiter.out -create_files -file_mask dickson_q2_%03d$
8566  \>$ -N 141$
8567  \>$ -line\_numeric "Surface\_Orb%d\_q2:\"
8568  \>$ -line\_numeric \"t \text{D(ORBITER~PATH)}/orbiter.out -v 3 \"
8569  \>$ -line\_numeric \"t t linear\_group -PGL\_4\_2 -wedge -end \"
8570  \>$ -line\_numeric \"t t group\_theoretic\_activity \"
8571  \>$ -line\_numeric \"t t control\_six\_arcs -end \"
8572  \>$ -line\_numeric \"t t define\_surface -label\_txt "Orb%d\_q2 \"
8573  \>$ -label\_tex "Rank -D(0rb%d)"
8574  \>$ -label\_for\_summary \"0rb%d\" -by_rank \text{D(0rb%d) -2 -q 2 \"
8575  \>$ -line\_numeric \"t t end \"
8576  \>$ -line\_numeric \"t t end \"
8577  \>$ -line\_numeric \"\"
8578  \>$ end
8579  \> cat dickson_q2* \> dickson_q2
8580
8581
8582 create_dickson_makefiles_q4:
8583  \$\text{(ORBITER~PATH)} orbiter.out -create_files -file_mask dickson_q4_%03d$
8584  \>$ -N 141$
8585  \>$ -line\_numeric "Surface\_Orb%d\_q4:\"
8586  \>$ -line\_numeric \"t \text{D(ORBITER~PATH)}/orbiter.out -v 3 \"
8587  \>$ -line\_numeric \"t t linear\_group -PGL\_4\_4 -wedge -end \"

565
create_dickson_makefiles.q8:

$\$(\text{ORBITER\_PATH})\text{orbiter.out--create\_files--file\_mask\_dickson\_q8\_\%03d\}\$\$

create_dickson_makefiles.q16:

$\$(\text{ORBITER\_PATH})\text{orbiter.out--create\_files--file\_mask\_dickson\_q16\_\%03d\}\$\$

create_dickson_makefiles.q32:

$\$(\text{ORBITER\_PATH})\text{orbiter.out--create\_files--file\_mask\_dickson\_q32\_\%03d\}\$\$

566
create_dickson_makefiles_q64:
$(ORBITER_PATH)orbiter.out --create_files --file_mask dickson_q64_%03d
create_dickson_makefiles_top_line_q2:
$(ORBITER_PATH)orbiter.out --create_files --file_mask dickson_q2_top
create_dickson_makefiles_top_line_q4:
$(ORBITER_PATH)orbiter.out --create_files --file_mask dickson_q4_top
create_dickson_makefiles_top_line_q8:
$(ORBITER_PATH)orbiter.out --create_files --file_mask dickson_q8_top
create_dickson_makefiles_top_line_q16:
$(ORBITER_PATH)orbiter.out --create_files --file_mask dickson_q16_top
create_dickson_makefiles_top_line_q32:
create_dickson_makefiles_top_line_q64:
surfaces_q4_join:
create_dickson_makefiles_join_q2:
create_dickson_makefiles_join_q4:
dickson_q2_latex:
create_dickson_atlas:
quartic_curve_tables_23:
> $(ORBITER_PATH)orbetter.out -v.3\ 
> -define F -finite_field -q.23 -end\ 
> -define P -projective_space 2 F -end\ 
> -with P -do\ 
> -projective_space_activity\ 
> -table of quartic curves\ 
> -end
quartic_curve_tables:
> $(ORBITER_PATH)orbetter.out -v.3\ 
> -define F -finite_field -q.9 -end\ 
> -define P -projective_space 2 F -end\ 
> -with P -do\ 
> -projective_space_activity\ 
> -table of quartic curves\ 
> -end
quartic_curve_tables\ latex:
> $(ORBITER_PATH)orbetter.out -v.3\ 
> -csv file latex\ 1 test.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q9 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q13 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q17 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q19 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q25 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q27 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv filelatex 0 quartic curves q29 info.csv
> $(ORBITER_PATH)orbetter.out -v.3 -csv file latex 0 quartic curves q31 info.csv
# Chapter 8: Applications

## Section 8.1: Number Theory

### inverse_mod_a:

```bash
$($ORBITER_PATH)orbiter.out-v.2-inverse_mod.18059241-58014043
```

### jacobi_35_41:

```bash
$($ORBITER_PATH)orbiter.out-v.5-jacobi.35-41
```
\begin{verbatim}
sqrt_mod_7817:
> $(ORBITER_PATH)oriter.out-\v.2-square_root_mod-2217-8821

#Section 8.2: Combinatorics
SECTION_COMBINATORICS:
>
Char_Sym_4:
> $(ORBITER_PATH)oriter.out-\v.2-character_table_symmetric_group.4

Char_Sym_5:
> $(ORBITER_PATH)oriter.out-\v.2-character_table_symmetric_group.5

Char_Sym_6:
> $(ORBITER_PATH)oriter.out-\v.2-character_table_symmetric_group.6

Walsh_matrix_4:
> $(ORBITER_PATH)oriter.out-\v.3-
> -define F-\texttt{finite_field}-q2-end-
> -with F-do-\texttt{finite_field_activity}-
> -Walsh_matrix-4-end
> $(ORBITER_PATH)oriter.out-\v.2-draw_matrix-
> -input_csv_file-Walsh_01_4.csv-
> -box_width-10-bit_depth-24-partition-3:16:16-end
> #pdflatex-GF_2.tex
> #open-GF_2.pdf

Dedekind_10_10:
> $(ORBITER_PATH)oriter.out-\v.3-Dedekind_numbers:2:10:2:10

Dedekind_30_2:
\end{verbatim}
8866  $(ORBITER_PATH) orbiter.out -v 3 -Dedekind_numbers 2 30 2 2
8867
8868
8869  Dedekind_100_2:
8870  $(ORBITER_PATH) orbiter.out -v 3 -Dedekind_numbers 2 100 2 2
8871
8872
8873
8874  elementary_symmetric_functions_4:
8875  $(ORBITER_PATH) orbiter.out -make_elementary_symmetric_functions 4 4
8876
8877  elementary_symmetric_functions_8:
8878  $(ORBITER_PATH) orbiter.out -make_elementary_symmetric_functions 8 8
8879
8880
8881
8882
8883
8884
8885
8886
8887  # large sets:
8888
8889  GENERATORS_H5="1,2,3,4,0,6,7,8,9,5,10,11,12"
8890  # (0,1,2,3,4)(5,6,7,8,9)
8891
8892  GENERATORS_C13="11,0,10,12,5,3,7,4,2,8,6,9,1"
8893  #(0,11,9,8,2,10,6,7,4,5,3,12,1)
8894
8895
8896
8897  LARGE_SET_S0="0,1,2,3,4,5,6,7,8,9,10,11,12"
8898  # identity
8899
8900  LARGE_SET_S1="6,8,9,2,7,10,1,11,0,3,5,4,12"
8901
8902  #\dots (0,6,1,8)(2,9,3)(4,7,11)(5,10),
8903
8904
8905
8906  LARGE_SET_S2="2,0,1,6,3,4,11,5,7,8,10,9,12"
8907  #\dots (0,2,1)(3,6,11,9,8,7,5,4),
8908
8909
8910  LARGE_SET_S3="12,5,6,11,3,7,10,8,9,1,4,2,0"
8911  #\dots (0,12)(1,5,7,8,9)(2,6,10,4,3,11),
8912

572
Large set $S_0$ = "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 $S_1$ = "10,11,0,7,12,2,3,1,4,5,8,6,9"
#.....(0,10,8,4,12,9,5,2)(1,11,6,3,7), \ 

Large set $S_2$ = "3,4,1,9,5,6,8,2,7,11,12,10"
#.....(0,3,9,11,10,12)(1,4,5,6,8,7,2), \ 

Large set $S_3$ = "9,11,0,6,1,3,5,4,2,12,8,7,10"
#.....(0,9,12,10,8,2)(1,11,7,4)(3,6,5), \ 

Large set $S_4$ = "10,2,12,8,0,3,4,1,5,6,9,7,11"
#.....(0,10,9,6,4)(1,2,12,11,7)(3,8,5), \ 

Large set $S_5$ = "1,3,4,10,5,6,9,7,8,11,0,12,2"
#.....(0,1,3,10)(2,4,5,6,9,11,12), \ 

Large set $S_6$ = "7,12,1,6,0,4,5,2,3,10,9,8,11"
#.....(0,7,2,1,12,11,8,3,6,5,4)(9,10).

file_S:

echo ROW,0 "$\{(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)\}"

> S.csv

Large set $H_5$:

$(\text{ORBITER\_PATH}) oral\_out -v 10 \$

$\text{define G} - permutation\_group - symmetric\_group 13 \$

$\text{define G} - subgroup\_by\_generators H_5 5 1 \$

$\text{define G} - do \$

$\text{define G} - group\_theoretic\_activity \$

$\text{define G} - report \$

$\text{define G} - end \$

$\text{define G} - do \$

$\text{define G} - group\_theoretic\_activity \$

$\text{define G} - save\_elements\_csv H5\_elts.csv \$

$\text{define G} - end \$

$pdflatex\_Perm13\_Subgroup\_H5\_5\_report.tex$
Large_set_C13:

```plaintext
dollar $(ORBITER_PATH) orbiter.out -v 10 \  
define G permutation_group symmetric_group 13 \  
subgroup_byGenerators C13 13 1 $(GENERATORS_C13) -end \  
with G -do \  
group_theoretic_activity \  
report -end \  
with G -do \  
group_theoretic_activity \  
with G -do \  
group_theoretic_activity \  
report -end \  
group_theoretic_activity \  
multiply_elements_csv_column_major_ordering C13 elts.csv S.csv C13xS.csv \  
end \  
report -end \  
group_theoretic_activity \  
multiply_elements_csv_column_major_ordering C13 elts.csv S.csv C13xS.csv \  
end
```

## the following lines were created using -export_orbiter:

```
GeneratorsPerm13_Subgroup_C13_13 = "11,0,10,12,5,3,7,4,2,8,6,9,1"
Perm13_Subgroup_C13_13:
```

Large_set_mult_C13xS:

```
report -end
```

Large_set_mult_C13xSxH5:

```
report -end
```
representations of degree 3:

representation_on_polynomials_of_degree_3:

representation_tetrahedral_group_on_polynomials_of_degree_3:
Section 8.4: Cryptography

EC_add:

EC_cyclic_subgroup:

EC_points_13:
EC_points_199:
$\texttt{\$(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

 $\texttt{\$(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

 EC_Koblitz_encoding:

 $\texttt{\$(ORBITER\_PATH)orbiter.out-v.6-seed:17\$

 define $F$-finite_field-q.199-end

 with $F$-do

 finite_field_activity

 EC_Koblitz_encoding:5-7.67."147,164"."DEADBEEF"

 end

 EC_bsgs:

 $\texttt{\$(ORBITER\_PATH)orbiter.out-v.2\$

 define $F$-finite_field-q.199-end

 with $F$-do

 finite_field_activity

 EC_bsgs:5-7."147,164".212

 "172,158,45,195,50,22,10,103,55,33,50,22,145,105,31,74,73,155,67,60,25,6"

 end

 EC_bsgs_decode:

 $\texttt{\$(ORBITER\_PATH)orbiter.out-v.2\$

 define $F$-finite_field-q.199-end

 with $F$-do

 finite_field_activity

 EC_bsgs_decode:5-7."129,176".212

 "127,188,51,141,85,29,106,90,41,105,179,71,171,2,16,197,183,72,27,129,37,10"

 "50,179,169,13,153,169,115,116,188,110,176"

 end

 NTRU_N=7

 NTRU_P=3
NTRU Q=41
NTRU D=2
NTRUE XN1="-1,0,0,0,0,0,0,1,"
# D·plus·ones·and·D·minus·ones
ALICE_PRIVATE_F="-1,0,1,1,-1,0,1"
# D·plus·ones·and·D·minus·ones
ALICE_PRIVATE_G="0,-1,-1,0,1,0,1"

ALICE PRIVATE F="-1,0,1,1,-1,0,1"
# D·plus·ones·and·D·minus·ones
ALICE PRIVATE G="0,-1,-1,0,1,0,1"

ALICE PRIVATE FQ="37,2,40,21,31,26,8"

ALICE private:

Alice1:

⊿ $(ORBITER PATH)orbiter.out -v:2 \n-define F·finite_field=q $(NTRUE Q) -end \n-with F·do \n-finite_field_activity \n-extended_gcd_for_polynomials $(NTRUE XN1) $(ALICE PRIVATE F) -end \n
#F q(x) =-8X^6 +26X^5 +31X^4 +21X^3 +40X^2 +2X +37
ALICE PRIVATE FQ="37,2,40,21,31,26,8"

Alice2:

⊿ $(ORBITER PATH)orbiter.out -v:2 \n-define F·finite_field=q $(NTRUE P) -end \n-with F·do \n-finite_field_activity \n-extended_gcd_for_polynomials $(NTRUE XN1) $(ALICE PRIVATE F) -end \n
#F p(x) =X^6 +2X^5 +X^3 +X^2 +X +1
ALICE PRIVATE FP="1,1,1,0,2,1"

Alice public key:

⊿ $(ORBITER PATH)orbiter.out -v:2 \n-define F·finite_field=q $(NTRUE Q) -end \n-with F·do \n-finite_field_activity \n-polynomial_mult_mod $(ALICE PRIVATE F) \n-extended_gcd_for_polynomials $(NTRUE XN1) $(ALICE PRIVATE F) -end \n
#C(X)=20X^6 +40X^5 +2X^4 +38X^3 +8X^2 +26X +30
ALICE_PUBLIC_KEY="30,26,8,38,2,40,20"

BOB MESSAGE="1,-1,1,1,0,-1"
BOB ONE_TIME KEY="-1,1,0,0,0,-1,1"

NTRU encrypt:

⊿ $(ORBITER PATH)orbiter.out -v:2 \n
NTRU decrypt1:
$(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

BOB ENCRYPT ="25,3,40,2,4,19,31"

#E(X) = 31X^6 + 19X^5 + 4X^4 + 2X^3 + 40X^2 + 3X + 25
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_mult_mod $(ALICE_PRIVATE_F) -
$(BOB_ENCRYPT) $(NTRUE_XN1) -
end

#A(X) = X^6 + 10X^5 + 8X^4 - X^3 - X^2 + 1
ALICE C2 ="-1,1,-1,-1,-8,10,1"

NTRU decrypt3:
$(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 + X^4 + X^3 + 2X^2 + X + 2
ALICE C3 ="2,1,2,1,1,1"
\begin{verbatim}
9234 \>
9235 \>
9236 \>
9237 #C(X)=2X^5+X^3+X^2+2X+1
9238 ALICE_C4="1,2,1,1,0,2"
9239

9240 NTRU_decrypt5:
9241 \>
9242 \>
9243 \>
9244 \>
9245 \>
9246 \>
9247 \>
9248 #A(X)=-X^5+X^3+X^2-X+1
9249 #plaintext-BOB_MESSAGE
9250
9251
9252
9253
9254 ++++
9255
9256
9257 inv_59_mod:
9258 \>
9259 \>
9260 #the-inverse-of-59-mod-10200-is-2939
9261
9262
9263
9264 RSA_e:
9265 \>
9266 \>
9267 \>
9268 RSA_d:
9269 \>
9270 \>
9271
9272 im1:
9273 \>
9274 \>
9275 #the-inverse-of-869-mod-1843488-is-386093
9276
9277 #FUNFACTOR:
9278 \>
9279 \>
9280
\end{verbatim}
### RSA_d1:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -RSA-869-1846303.3:"1248407,345776,317846"
```

### im1061:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -inverse_mod.1061.25320204
```

### RSA_e2:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -RSA_encrypt_text.2076209.25330309.3-creamcheese
```

### RSA_d2:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -RSA.1061.25330309.3:"19019931,1619805,740498,2671344"
```

### im3:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -inverse_mod.2909.59248840
```

### RSA_e3:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -RSA_encrypt_text.2909.59264263.3-encrypted
```

### RSA_d3:

```bash
> $(ORBITER_PATH)orbiter.out -v.2 -RSA.4358629.59264263.3:"35270141,9642524,49058157"
```
im4: $(ORBITER\_PATH) orbiter.out \(-v\;2\;\-inverse\_mod\;583\;62236200$
# the inverse of $583\;62236200$ is $32559247$

im5:
# the inverse of $173\;52504368$ is $38543669$

RSA\_e4:
> $(ORBITER\_PATH) orbiter.out \(-v\;2\;\-RSA\_encrypt\_text\;583\;62251979\;3\;venividivici$
# $\text{RSA\_encrypt\_text}\;583\;62251979\;\text{venividivici}$
# $40513610,53979973,56449676,35068535$

RSA\_d4:
> $(ORBITER\_PATH) orbiter.out \(-v\;2\;\-RSA\;32559247\;62251979\;\text{"40513610,53979973,56449676,35068535"}$
$\text{(ORBITER\_PATH)orbiter.out-v.2-RSA}\text{173-52518863-"31526751,8962078,51045732,51894467"}$

RSA_d6:

$\text{(ORBITER\_PATH)orbiter.out-v.2-RSA}\text{47177497-55040413-"28702119,48926559"}$

smooth:

$\text{(ORBITER\_PATH)orbiter.out-v.2-sift_smooth\_100000\_100\_2,3,5,7,11,13,17,19"}$

im7:

$\text{(ORBITER\_PATH)orbiter.out-v.2-inverse_mod\_3221\_15796188}$

#the_inverse_of_3221_mod_15796188_is_10048553

im8:

$\text{(ORBITER\_PATH)orbiter.out-v.2-inverse_mod\_9017\_60240544}$

#the_inverse_of_9017_mod_60240544_is_14430473

RSA_e7:

$\text{(ORBITER\_PATH)orbiter.out-v.2-RSA_encrypt_text\_10048553\_15806093\_3-beachandfun}$

sqrt_big:

$\text{(ORBITER\_PATH)orbiter.out-v.2-square_root\_1002001}$

sqrt_mod_33_41:
quadratic_sieve:

```
$\$(ORBITER\_PATH)\texttt{orbiter.out\textasciitilde v\textasciitilde quadratic\_sieve\textasciitilde 31\textasciitilde 500\textasciitilde 1}
```

```
# 
```

pseudoprime3:

```
$\$(ORBITER\_PATH)\texttt{orbiter.out\textasciitilde v\textasciitilde seed\textasciitilde 2531011\textasciitilde find\_pseudoprime\textasciitilde 3\textasciitilde 5\textasciitilde 0\textasciitilde 0}
```

```
$\$(ORBITER\_PATH)\texttt{pdflatex\textasciitilde pseudoprime\_3\textasciitilde tex}
```

```
$\$(ORBITER\_PATH)\texttt{open\textasciitilde pseudoprime\_3\textasciitilde pdf}
```

pseudoprime10:

```
$\$(ORBITER\_PATH)\texttt{orbiter.out\textasciitilde v\textasciitilde seed\textasciitilde 2531011\textasciitilde find\_pseudoprime\textasciitilde 10\textasciitilde 5\textasciitilde 5\textasciitilde 5}
```

```
$\$(ORBITER\_PATH)\texttt{pdflatex\textasciitilde pseudoprime\_10\textasciitilde tex}
```

```
$\$(ORBITER\_PATH)\texttt{open\textasciitilde pseudoprime\_10\textasciitilde pdf}
```

```
# 4460190157
```

pseudoprime11:

```
$\$(ORBITER\_PATH)\texttt{orbiter.out\textasciitilde v\textasciitilde seed\textasciitilde 2531011\textasciitilde find\_pseudoprime\textasciitilde 11\textasciitilde 5\textasciitilde 5\textasciitilde 5}
```

```
$\$(ORBITER\_PATH)\texttt{pdflatex\textasciitilde pseudoprime\_11\textasciitilde tex}
```

```
$\$(ORBITER\_PATH)\texttt{open\textasciitilde pseudoprime\_11\textasciitilde pdf}
```

```
# 6381463367
```

```
# product\textasciitilde is\textasciitilde 284625399616057168619
```

pseudoprime20:

```
$\$(ORBITER\_PATH)\texttt{orbiter.out\textasciitilde v\textasciitilde seed\textasciitilde 2531011\textasciitilde find\_pseudoprime\textasciitilde 20\textasciitilde 5\textasciitilde 5\textasciitilde 5}
```

```
$\$(ORBITER\_PATH)\texttt{pdflatex\textasciitilde pseudoprime\_20\textasciitilde tex}
```

```
$\$(ORBITER\_PATH)\texttt{open\textasciitilde pseudoprime\_20\textasciitilde pdf}
```

```
# 584
```
PR10:

\$(\text{ORBITER\ PATH})\text{orbiter.out-v-5.-primitive\ root-4460190157}

# mistake! long integer overflow
# a primitive root modulo 165222861 is 1293

pseudoprime50:

\$(\text{ORBITER\ PATH})\text{orbiter.out-v-5.-seed-2531011-find\ pseudoprime-50-5-0-0}

\text{pdfflatex pseudoprime_50.tex}

\text{open-pseudoprime_50.pdf}

# 91322792878581218181431392170986926262336688354473

pseudoprime51:

\$(\text{ORBITER\ PATH})\text{orbiter.out-v-5.-seed-2531011-find\ pseudoprime-51-5-5-5}

\text{pdfflatex pseudoprime_51.tex}

\text{open-pseudoprime_51.pdf}

# 754600727746834470214089702490004944659715367045417

# product 6891224596605081960619999423264315732335295324400658436661744403244049

# 57291409437904326661586100241

pseudoprime30:

\$(\text{ORBITER\ PATH})\text{orbiter.out-v-5.-seed-2531011-find\ pseudoprime-30-5-5-5}

\text{pdfflatex pseudoprime_30.tex}

\text{open-pseudoprime_30.pdf}

# 286525565474504516914595596387

pseudoprime31:

\$(\text{ORBITER\ PATH})\text{orbiter.out-v-5.-seed-2531011-find\ pseudoprime-31-5-5-5}

\text{pdfflatex pseudoprime_31.tex}

\text{open-pseudoprime_31.pdf}

# 8777266765422645523724129853331

# 251491132328298698837184692002835573476743643265896783515097

# maybe 2 seconds

pseudoprime33:
#371674199498295345543363004459891
pseudoprime33:
#9309708224110488378214945245346817
#·3460178351758962531912872979731874528849142238619677890786061016947
#·18·sec
pseudoprime34:
#81329557792505271120435930267561599139
#·1322619338309310524537919350220354135219441323641636665484262532145217
#factoring·takes·46·seconds
MATH360_hw2:
#162624680891993404333363207561599139
#1322619338309310524537919350220354135219441323641636665484262532145217
#with·F·do·finite_field_activity·
#parse_and_evaluate·"test"·"a+b"·"a=8,b=14"·end
#with·F·do·finite_field_activity·
#parse_and_evaluate·"test"·"a*b"·"a=9,b=13"·end
#with·F·do·finite_field_activity·
9555 ▶ ▶ -parse_and_evaluate"test":"a*a*a*a"."a=9".-end
9556 ▶ $(ORBITER_PATH)orbiter.out-v.3-define F--finite_field--q 16.-end\ 
9557 ▶ -with F--do--finite_field_activity\ 
9558 ▶ -parse_and_evaluate"test":"(a+b)*(a+b)"."a=5,b=7".-end
9559 ▶ $(ORBITER_PATH)orbiter.out-v.3-define F--finite_field--q 16.-end\ 
9560 ▶ -with F--do--finite_field_activity\ 
9561 ▶ -parse_and_evaluate"test":"a*a+b*b"."a=5,b=7".-end
9562
9563
9564 F_256_Rijndahl:
9565 ▶ $(ORBITER_PATH)orbiter.out-v.3\  
9566 ▶ -define F--finite_field--q 256--override_polynomial 283.-end\ 
9567 ▶ -with F--do--finite_field_activity--cheat_sheet GF.-end
9568
9569
9570
9571
9572 ntt:
9573 ▶ $(ORBITER_PATH)orbiter.out-v.2--NTT 4 17 ntt4.cpp
9574
9575
9576
9577 # $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--input_csv_file PG 2 4_singer_inc ma_subgroup_index 3.csv--box_width 20--bit_depth 24--partition 3 7 7 7 7 7.--en d
9578
9579 # problem:
9580
9581 FGDTP:
9582 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_F k4.csv 20 8
9583 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_Fv k4.csv 20 8
9584 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_AAv k4.csv 20 8
9585 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_G k4.csv 20 8
9586 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_D k4.csv 20 8
9587 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_T k4.csv 20 8
9588 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_Tv k4.csv 20 8
9589 ▶
9590 no:
9591 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_P k4.csv 20 8
9592 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_F k3.csv 20 8
9593 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_Gr k3.csv 20 8
9594 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_Dr k3.csv 20 8
9595 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_Tr k3.csv 20 8
9596 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_Pr k3.csv 20 8
9597 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_G k3.csv 20 8
9598 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_D k3.csv 20 8
9599 ▶ $(ORBITER_PATH)orbiter.out-v.2--draw_matrix--ntt_T k3.csv 20 8
n4.csv

function

matrix

mod

n

k2.csv

mod

n

al

t

9

588
SECTION CODING THEORY INTRODUCTION:

Hamming space 4 2 distance matrix:

```bash
$(ORBITER_PATH)orbiter.out--Hamming_space_distance_matrix 4 2
```

- input_csv_file: Hamming_n4_q2.csv
- box_width: 20 - bit_depth: 24 - partition: 4 16 16 - end

Open Hamming_n4_q2_draw.bmp

Code 5 2 3 diagram:

```bash
$(ORBITER_PATH)orbiter.out--v 2--code_diagram "code 5 2 3" 
$(CODE 5 2 3_CODEWORDS)5--metric_balls 1
```

- input_csv_file: code 5 2 3_diagram_01 5 4.csv
- box_width: 25 - bit_depth: 24 - partition: 4 8 4 - end

Hamming 5 2 graph:

```bash
$(ORBITER_PATH)orbiter.out--v 2 
define G--graph: Hamming 5 2 -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

```bash
$(ORBITER_PATH)orbiter.out--v 2--draw_matrix 
```

- input_csv_file: Hamming 5 2.csv
- box_width: 8 - bit_depth: 24 - partition: 4 32 32 - end

dot--Tpng: Hamming 5 2.gv > Hamming 5 2.png
Hamming 5\_2\_with\_5\_2\_3\_code:

\begin{verbatim}
> $(ORBITER\_PATH)orbiter.out -v 2:
  > -define G -graph Hamming 5\_2:
  > -subset "\_code\_5\_2\_3"."\_code\_5\_2\_3\_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

code_6:

\end{verbatim}

# linear code with generator matrix

\begin{verbatim}
> $(ORBITER\_PATH)orbiter.out -v 2:
  > -general_code_binary 6: "0,60,50,41,14,21,27,39"
  > $(ORBITER\_PATH)orbiter.out -v 2 -draw_matrix:
  > -input_csv_file code_matrix 8\_8.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
  > pdflatex code_6\_8.tex
  > open code_6\_8.pdf
  > open code_matrix 8\_8\_draw.bmp

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes

# Section 9.2: Hamming codes
SECTION CODING THEORY HAMMING CODES:

Hamming generator:

```latex
\$ (ORBITER\_PATH) orbiter.out -v.2 \$
\$ -define F -finite_field -q.2 -end \$
```

Hamming code words:

```latex
\$ (ORBITER\_PATH) orbiter.out -v.2 \$
\$ -define v -vector -dense \$(HAMMING\_CODE\_ROWS\_IN\_BINARY\_RANKS) -end -linear code through basis 7\$
```

Hamming weight enumerator:

```latex
\$ (ORBITER\_PATH) orbiter.out -v.2 \$
```

Hamming code diagram:

```latex
\$ -$(ORBITER\_PATH) orbiter.out -v.2 -code_diagram "Hamming 7\_4" \$
```
\$\text{(HAMMING\_CODE\_CODEWORDS)}\cdot 7 - \text{metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-draw\_matrix}\cdot \text{-end} \\
\text{-input\_csv\_file}\cdot \text{Hamming\_7\_4\_diagram\_01\_7\_16.csv} \\
\text{-box\_width\_25\cdot -bit\_depth\_24\cdot -partition\_4\_16\_8\cdot -end} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-draw\_matrix}\cdot \text{-end} \\
\text{-input\_csv\_file}\cdot \text{Hamming\_7\_4\_diagram\_7\_16.csv} \\
\text{-box\_width\_25\cdot -bit\_depth\_14\cdot -partition\_4\_16\_8\cdot -end} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_0\"}\cdot 0\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_1\"}\cdot 67\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_2\"}\cdot 37\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_3\"}\cdot 102\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_4\"}\cdot 22\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_5\"}\cdot 85\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_6\"}\cdot 51\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_7\"}\cdot 112\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_8\"}\cdot 15\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_9\"}\cdot 76\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_10\"}\cdot 42\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_11\"}\cdot 105\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_12\"}\cdot 25\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_13\"}\cdot 90\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_14\"}\cdot 60\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-code\_diagram}\cdot \text{"Hamming\_7\_4\_word\_15\"}\cdot 127\cdot 7\cdot \text{-metric\_balls:1} \\
\$\text{(ORBITER\_PATH)}\text{orbiter\_out}\cdot v\cdot 2\cdot \text{-loop\_L\_0\_16\_1\cdot -draw\_matrix}\cdot \text{-end} \\
\text{-input\_csv\_file}\cdot \text{Hamming\_7\_4\_word\_2\_L\_diagram\_7\_1.csv} \\
\text{-box\_width\_25\cdot -bit\_depth\_8\cdot -partition\_4\_16\_8\cdot -end\cdot -end\_loop}
Hamming systematic:

define v -vector -dense $(HAMMING_CODE_ROWS_IN_BINARY_RANKS) -end 

linear_code_through_basis 7 \cdot v

$\text{HAMMING CODE ROWS IN BINARY RANKS}$

$\text{HAMMING CODE GENERATOR}$

finite_field activity

finite_field

nullspace

finite_field

Hamming_long:

```
9881  $(ORBITER_PATH)orbiter.out-v:2-long_code:7:4\n9882  "0,5,6."
9883  "1,4,6."
9884  "2,4,5."
9885  "3,4,5,6."
9886  $(ORBITER_PATH)orbiter.out-v:2-loop:L:0:16:1-draw_matrix\n9887  -input_csv_file-long_code_genma_n7_k4_codeword_%L.csv\n9888  -box_width:25-bit_depth:8-partition:3:4:2\n9889  -end_loop
```

Hamming_code_macwilliams:

```
9897  $(ORBITER_PATH)orbiter.out-v:2-make_macwilliams_system:7:4:2
9898  pdflatex-MacWilliams_n7_k4_q2.tex
9899  open-MacWilliams_n7_k4_q2.pdf
```

Hamming_singer:

```
9903  -define-G-linear_group-PGL:3:2-singer:1-end\n9905  -group_theoretic_activity\n9906  -report\n9907  -orbits_on_points\n```
Hamming cyclic generator:

```
$($ORBITER_PATH)orbiter.out-v.2-
define F -finite_field -q.2-
\end
```

```
define v -vector -format 3 -field F -
\end
```

```
dense $(SIMPLEX_CODE_GENMA_CYCLIC) -
\end
```

```
with F -do -finite_field_activity -
nullspace v -
\end
```

```
$(ORBITER_PATH)orbiter.out-v.2-nullspace.3_7.tex
open-nullspace.3_7.pdf
```

Hamming cyclic long:

```
$($ORBITER_PATH)orbiter.out-v.2-long_code.7.4-
\end
```

```
"0,4,6" -
"1,4,5,6" -
"2,4,5" -
"3,5,6" -
```

```
$(ORBITER_PATH)orbiter.out-v.2-loop-L.0:16.1-draw_matrix-
\end
```

```
-input_csv_file-long_code_genma_n7_k4_codeword_%L.csv-
\end
```

```
-box_width 25 -bit_depth 8 -partition 3.4.2 -
\end_loop
```

Hamming cyclic:

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

```
define v -vector -dense "69,39,22,11" -
\end
```

```
-linear_code_through_basis 7.v
```

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

```
-input_csv_file_code_matrix_16.8.csv-
```

```
-box_width 25-bit_depth 8 -partition 2.16.8-
\end
```

```
open_code_matrix_16.8_draw.bmp
```

```
pdflatex-code_7_16.tex
```

```
open-code_7_16.pdf
```

Hamming cyclic clean:

```
\begin{verbatim}
#11111111 = 255
#01010101 = 170
#00110011 = 204
#00001111 = 240
\end{verbatim}
```
Golay23 code words:

$\text{ORBITER\ PATH}\text{orbiter.out}-v.2$

-define-v-\text{vector}-dense\text{(GOLAY\ COLUMN\ RANKS\ PROJECTIVELY)}-end$

-\text{linear\ code\ through\ columns\ of\ parity\ check\ projectively}\cdot 12\ v$

pdflatex code\_n23\_k12\_q2.tex

open-code\_n23\_k12\_q2.pdf

Golay23 code diagram:

$\text{ORBITER\ PATH}\text{orbiter.out}-v.2$

-code\_diagram\_from\_file\text{"Golay\ 23"}$

codewords\_n23\_k12\_q2.csv\cdot 23\ -\text{enhance}\cdot 4$

d-rule\ -\text{metric\ balls}\cdot 3$

Golay23 code diagram draw:

$\text{ORBITER\ PATH}\text{orbiter.out}-v.2$

draw\_matrix$

-input\_csv\_file\text{Golay\ 23\ diagram\_01\_23\_4096}\.csv$

-box\_width\cdot 4\ -\text{bit\ depth}\cdot 8$

-partition\cdot 20\cdot 4096\cdot 2048$

d-end

Golay23 code diagram draw:

$\text{ORBITER\ PATH}\text{orbiter.out}-v.2$

draw\_matrix$

-input\_csv\_file\text{Golay\ 23\ diagram\_01\_23\_4096}\.csv$

-box\_width\cdot 4\ -\text{bit\ depth}\cdot 8$

-partition\cdot 20\cdot 4096\cdot 2048$

d-end

Golay23 code diagram draw:

$\text{ORBITER\ PATH}\text{orbiter.out}-v.2$

draw\_matrix$

-input\_csv\_file\text{Golay\ 23\ diagram\_01\_23\_4096}\.csv$

-box\_width\cdot 4\ -\text{bit\ depth}\cdot 8$

-partition\cdot 20\cdot 4096\cdot 2048$

d-end

#-Section:9.3::Coding:Theory--Golay\-codes
encode_text_5bits:

encode_text_5bits_check:

encode_constitution_field_induction:

encode_text_5bits_1error:
encode_text_5bits_1error:
  $(ORBITER_PATH)orbiter.out-encode_text_5bits-
"Hithere"."text.csv"

encode_text_5bits_2error:
  $(ORBITER_PATH)orbiter.out-encode_text_5bits-
"text.csv"

encode_text_5bits_error_0_7:
  $(ORBITER_PATH)orbiter.out-encode_text_5bits-
"text.csv"
10131 \pdflatex-polynomial\_division\_file\_13.tex
10132 \open-polynomial\_division\_file\_13.pdf
10133
10134
10135
10136
10137
10138
10139 \texttt{CRC\_2\_4\_3:}
10140 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde v.3-define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10141 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.4.3-\textasciitilde\textasciitilde end}
10142
10143 \#algebra\_global::find\_CRC\_polynomials\_info=4\_check=3\_nb\_sol=3
10144 \#0::1101
10145 \#1::1011
10146 \#2::1111
10147
10148
10149 \texttt{CRC\_2\_4\_4:}
10150 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10151 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.4.4-\textasciitilde\textasciitilde end}
10152
10153 \texttt{CRC\_2\_5\_4:}
10154 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10155 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.5.4-\textasciitilde\textasciitilde end}
10156
10157 \texttt{CRC\_2\_6\_4:}
10158 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10159 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.6.4-\textasciitilde\textasciitilde end}
10160
10161 \texttt{CRC\_2\_7\_4:}
10162 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10163 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.7.4-\textasciitilde\textasciitilde end}
10164
10165 \texttt{CRC\_2\_8\_4:}
10166 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10167 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.8.4-\textasciitilde\textasciitilde end}
10168
10169 \texttt{CRC\_2\_9\_4:}
10170 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10171 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.9.4-\textasciitilde\textasciitilde end}
10172
10173 \texttt{CRC\_2\_9\_5:}
10174 \texttt{\$ORBITER\_PATH\textbackslash orbiter.out-\textasciitilde\textasciitilde define F-\textasciitilde\textasciitilde finite\_field-\textasciitilde\textasciitilde q.2-\textasciitilde\textasciitilde end\textbackslash \}
10175 \texttt{\textasciitilde\textasciitilde -with F-\textasciitilde\textasciitilde do-\textasciitilde\textasciitilde finite\_field\_activity-\textasciitilde\textasciitilde find\_CRC\_polynomials:2.9.5-\textasciitilde\textasciitilde end}
10176
10177 \texttt{CRC\_2\_10\_5:}
 pinnacle $\circ (\text{PATH})orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:10:5 -end
CRC 2_32:6:
  $(\text{PATH})orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:32:6 -end
CRC 2_64:8:
  $(\text{PATH})orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:64:8 -end
CRC 2_128:8:
  $(\text{PATH})orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:128:8 -end
CRC 2_128:16:
  $(\text{PATH}) orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:128:16 -end
CRC 3_128:8:
  $(\text{PATH}) orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :3:128:8 -end
CRC 3_128:10:
  $(\text{PATH}) orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :3:128:10 -end
CRC 2_256:8:
  $(\text{PATH}) orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:256:8 -end
CRC 2_256:10:
  $(\text{PATH}) orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:256:10 -end
CRC 2_256:12:
  $(\text{PATH}) orbiter.out$ define F finite_field q 2 -end:\
  -with F do -finite_field_activity -find_CRC_polynomials :2:256:10 -end
  -with F do -finite_field_activity -find_CRC_polynomials :2:256:12 -end
10225  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.1\cdot\text{define}\cdot\text{finite}\_\text{field}\cdot q.2\cdot\text{end}\ \backslash
10226  \triangleright \ -\text{with}\cdot\text{F}\cdot\text{do}\cdot\text{finite}\_\text{field}\_\text{activity}\cdot\text{find}\_\text{CRC}\_\text{polynomials}\cdot 2.256.12\cdot\text{end}
10227
10228
10229  \#\text{algebra}\_\text{global}::\text{find}\_\text{CRC}\_\text{polynomials}\_\text{info} = 256\cdot\text{check} = 12\cdot\text{nb}\_\text{sol} = 2654
10230  \#\text{User}\_\text{time}::2:0
10231
10232  \text{CRC\_3.256.10}: 
10233  $(\text{ORBITER\_PATH})\text{orbiter.out}-\text{define}\cdot\text{F}\cdot\text{finite}\_\text{field}\cdot q.2\cdot\text{end}\ \backslash
10234  \triangleright \ -\text{with}\cdot\text{F}\cdot\text{do}\cdot\text{finite}\_\text{field}\_\text{activity}\cdot\text{find}\_\text{CRC}\_\text{polynomials}\cdot 3.256.10\cdot\text{end}
10235
10236
10237  \#\text{algebra}\_\text{global}::\text{find}\_\text{CRC}\_\text{polynomials}\_\text{info} = 256\cdot\text{check} = 10\cdot\text{nb}\_\text{sol} = 140
10238  \#\text{Orbiter}\_\text{session}\_\text{finished}.
10239  \#\text{User}\_\text{time}::8:7
10240
10241  \text{CRC\_3.256.16}: 
10242  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.1\cdot\text{define}\cdot\text{F}\cdot\text{finite}\_\text{field}\cdot q.2\cdot\text{end}\ \backslash
10243  \triangleright \ -\text{with}\cdot\text{F}\cdot\text{do}\cdot\text{finite}\_\text{field}\_\text{activity}\cdot\text{find}\_\text{CRC}\_\text{polynomials}\cdot 3.256.16\cdot\text{end}
10244
10245
10246  \text{CRC\_3.8.4}: 
10247  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.2\cdot\text{define}\cdot\text{F}\cdot\text{finite}\_\text{field}\cdot q.2\cdot\text{end}\ \backslash
10248  \triangleright \ -\text{with}\cdot\text{F}\cdot\text{do}\cdot\text{finite}\_\text{field}\_\text{activity}\cdot\text{find}\_\text{CRC}\_\text{polynomials}\cdot 3.8.4\cdot\text{end}
10249
10250
10251  \text{CRC\_3.64.8}: 
10252  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.1\cdot\text{define}\cdot\text{F}\cdot\text{finite}\_\text{field}\cdot q.2\cdot\text{end}\ \backslash
10253  \triangleright \ -\text{with}\cdot\text{F}\cdot\text{do}\cdot\text{finite}\_\text{field}\_\text{activity}\cdot\text{find}\_\text{CRC}\_\text{polynomials}\cdot 3.64.8\cdot\text{end}
10254
10255
10256  \text{CRC\_poly.3.128.10}: 
10257  $(\text{ORBITER\_PATH})\text{orbiter.out}-v.1\cdot\text{define}\cdot\text{F}\cdot\text{finite}\_\text{field}\cdot q.2\cdot\text{end}\ \backslash
10258  \triangleright \ -\text{with}\cdot\text{F}\cdot\text{do}\cdot\text{finite}\_\text{field}\_\text{activity}\cdot\text{find}\_\text{CRC}\_\text{polynomials}\cdot 3.128.10\cdot\text{end}
10259
10260
10261  \#\cdot12/26/2020::243\cdot\text{polynomials}\_\text{in}::0:56\cdot\text{minutes}\_\text{on}\_\text{Mac}
SECTION CODING THEORY REED MULLER CODES:

```
RM_3_1_code_words:
  $(ORBITER_PATH)orbiter.out-v.2-
  -define-v-vector-dense$(REED_MULLER_3_1_BASIS_IN_BINARY)-end-
  -linear_code_through_basis:8.v
  pdflatex-code_n8_k4_q2.tex
  open-code_n8_k4_q2.pdf

#Codewords: (0,255,170,85,204,51,102,153,240,15,90,165,60,195,150,105)

RM_3_1_Hamming_space_diagram:
  $(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_split:
  $(ORBITER_PATH)orbiter.out-split_by_values:RM_3_1_holes_8_16.csv-

RM_3_1_holes_draw:
  $(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-
```
$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+X0*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*X5*X8^5+
X0*X1*X6*X8^5+X0*X1*X7*X8^5+X0*X2*X3*X8^5+X0*X2*X4*X8^5+X0*X2*X5*X8^5+
X0*X2*X6*X8^5+X0*X2*X7*X8^5+X0*X3*X4*X8^5+X0*X3*X5*X8^5+X0*X3*X6*X8^5+
X0*X3*X7*X8^5+X0*X4*X5*X8^5+X0*X4*X6*X8^5+X0*X4*X7*X8^5+X0*X5*X6*X8^5+
X0*X5*X7*X8^5+X0*X6*X7*X8^5+X0*X6*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+X0*X1*X2*X8^4+X0*X1*X2*X7*X8^4+X0*X1*X2*X6*X8^4+
X0*X1*X2*X5*X8^4+X0*X1*X2*X4*X8^4+X0*X1*X2*X3*X8^4+X0*X1*X2*X2*X8^4+
X0*X1*X2*X1*X8^4+X0*X1*X2*X0*X8^4+X0*X2*X3*X8^4+X0*X2*X3*X7*X8^4+X0*X2*X3*
X6*X8^4+X0*X2*X3*X5*X8^4+X0*X2*X3*X4*X8^4+X0*X2*X3*X3*X8^4+X0*X2*X3*X2*X8^4+
X0*X2*X3*X1*X8^4+X0*X2*X3*X0*X8^4+X0*X2*X2*X8^4+X0*X2*X2*X7*X8^4+X0*X2*X2*
X6*X8^4+X0*X2*X2*X5*X8^4+X0*X2*X2*X4*X8^4+X0*X2*X2*X3*X8^4+X0*X2*X2*X2*X8^4+
X0*X2*X2*X1*X8^4+X0*X2*X2*X0*X8^4+X0*X2*X1*X8^4+X0*X2*X1*X7*X8^4+X0*X2*X1*
X6*X8^4+X0*X2*X1*X5*X8^4+X0*X2*X1*X4*X8^4+X0*X2*X1*X3*X8^4+X0*X2*X1*X2*X8^4+
X0*X2*X1*X1*X8^4+X0*X2*X1*X0*X8^4+X0*X2*X0*X8^4+X0*X1*X2*X8^4+X0*X1*X2*X7*
X8^4+X0*X1*X2*X6*X8^4+X0*X1*X2*X5*X8^4+X0*X1*X2*X4*X8^4+X0*X1*X2*X3*X8^4+
X0*X1*X2*X2*X8^4+X0*X1*X2*X1*X8^4+X0*X1*X2*X0*X8^4+X0*X1*X1*X8^4+X0*X1*X1*
X7*X8^4+X0*X1*X1*X6*X8^4+X0*X1*X1*X5*X8^4+X0*X1*X1*X4*X8^4+X0*X1*X1*X3*X8^4+
X0*X1*X1*X2*X8^4+X0*X1*X1*X1*X8^4+X0*X1*X1*X0*X8^4+X0*X1*X0*X8^4+X0*X1*X0*X7*
X8^4+X0*X1*X0*X6*X8^4+X0*X1*X0*X5*X8^4+X0*X1*X0*X4*X8^4+X0*X1*X0*X3*X8^4+
X0*X1*X0*X2*X8^4+X0*X1*X0*X1*X8^4+X0*X1*X0*X0*X8^4+X0*X0*X8^4+X0*X0*X7*X8^4+
X0*X0*X6*X8^4+X0*X0*X5*X8^4+X0*X0*X4*X8^4+X0*X0*X3*X8^4+X0*X0*X2*X8^4+X0*
X0*X2*X7*X8^4+X0*X0*X2*X6*X8^4+X0*X0*X2*X5*X8^4+X0*X0*X2*X4*X8^4+X0*X0*X2*
X3*X8^4+X0*X0*X2*X2*X8^4+X0*X0*X2*X1*X8^4+X0*X0*X2*X0*X8^4+X0*X0*X1*X8^4+
X0*X0*X1*X7*X8^4+X0*X0*X1*X6*X8^4+X0*X0*X1*X5*X8^4+X0*X0*X1*X4*X8^4+X0*X0*
X1*X3*X8^4+X0*X0*X1*X2*X8^4+X0*X0*X1*X1*X8^4+X0*X0*X1*X0*X8^4+X0*X0*X0*X8^4+
X0*X0*X0*X7*X8^4+X0*X0*X0*X6*X8^4+X0*X0*X0*X5*X8^4+X0*X0*X0*X4*X8^4+X0*X0*
X0*X0*X3*X8^4+X0*X0*X0*X2*X8^4+X0*X0*X0*X1*X8^4+X0*X0*X0*X0*X8^4+X0*X0*
X8^4 + X0*X2*X3*X7*X8^4 + X0*X2*X4*X5*X8^4 + X0*X2*X4*X7*X8^4 + X0*X2*X5*X6*X8^4 + X0*X2*X6*X7*X8^4 + X0*X3*X4*X5*X8^4 + X0*X3*X4*X6*X8^4 + X0*X3*X5*X7*X8^4 + X0*X3*X5*X6*X8^4 + X0*X4*X5*X6*X7*X8^4 + X0*X4*X6*X7*X8^4 + X0*X5*X6*X7*X8^4 + X0*X5*X6*X8^4 + X1*X2*X3*X4*X5*X8^4 + X1*X2*X4*X5*X6*X7*X8^4 + X1*X2*X5*X6*X7*X8^4 + X1*X3*X4*X5*X6*X8^4 + X1*X3*X5*X6*X7*X8^4 + X1*X4*X5*X6*X7*X8^4 + X1*X5*X6*X7*X8^4 + X2*X3*X4*X5*X6*X7*X8^4 + X2*X3*X4*X5*X6*X8^4 + X2*X4*X5*X6*X7*X8^4 + X2*X4*X5*X6*X8^4 + X2*X5*X6*X7*X8^4 + X2*X5*X6*X8^4 + X2*X6*X7*X8^4 + X2*X6*X8^4 + X2*X7*X8^4 + X2*X8^4 + X3*X4*X5*X6*X7*X8^4 + X3*X4*X5*X6*X8^4 + X3*X4*X5*X7*X8^4 + X3*X4*X5*X8^4 + X3*X4*X6*X7*X8^4 + X3*X4*X6*X8^4 + X3*X4*X7*X8^4 + X3*X4*X8^4 + X3*X5*X6*X7*X8^4 + X3*X5*X6*X8^4 + X3*X5*X7*X8^4 + X3*X5*X8^4 + X3*X6*X7*X8^4 + X3*X6*X8^4 + X3*X7*X8^4 + X3*X8^4 + X4*X5*X6*X7*X8^4 + X4*X5*X6*X8^4 + X4*X5*X7*X8^4 + X4*X5*X8^4 + X4*X6*X7*X8^4 + X4*X6*X8^4 + X4*X7*X8^4 + X4*X8^4 + X5*X6*X7*X8^4 + X5*X6*X8^4 + X5*X7*X8^4 + X5*X8^4 + X6*X7*X8^4 + X6*X8^4 + X7*X8^4 + X8^4

10346
10347 \#-E_2+-E_3+-E_4
10348
10349
10350 RM_4_1:
10351 \>$\$(ORBITER_PATH)orbiter.out-v.2:\$
10352 \>$-\linear_code_through_columns_of_parity_check-5:\$
10353 \>$\$(REED_MULLER_4_1_COLUMNS_OF_PARITY_CHECK)\$
10354 \>$pdflatex\$code_n16_k5_q2.tex$
10355 \>$open-code_n16_k5_q2.pdf$
10356
10357
10358 \#-codewords_n16_k5_q2.csv
10359
10360
10361
10362 RM_4_1_diagram:
10363 \>$\$(ORBITER_PATH)orbiter.out-v.2:\$
10364 \>$-\code_diagram_from_file-\"RM_4_1\"-\$
10365 \>$\$codewords_n16_k5_q2.csv-16:\$
10366 \>$\#$-enhance-4:
10367 \>$\#$-metric_balls-3
10368
10369 RM_4_1_diagram_draw:
10370 \>$\$(ORBITER_PATH)orbiter.out-v.2-draw_matrix-\$
10371 \>$\$-input_csv_file-RM_4_1_diagram_01_16_32.csv-\$
10372 \>$\$-box_width-25-bit_depth-8-\$partition-10:256:256-end$
10373 \>$open-RM_4_1_diagram_01_16_32.draw.bmp$
10374
10375
10376 RM_4_1_split:
10377 \>$\$(ORBITER_PATH)orbiter.out-split_by_values-RM_4_1_holes_16_32.csv$
10378
10379 RM_4_1_diagram_draw_holes:
10380 \>$\$(ORBITER_PATH)orbiter.out-v.2-draw_matrix-\$
10381 \>$\$-input_csv_file-RM_4_1_holes_16_32.csv-\$
10382 \>$\$-box_width-25-bit_depth-8-\$partition-10:256:256-end$

605
10427
10428
10429
10430 RM6:
10431 > $(ORBITER_PATH)orbiter.out -v 2 -long_code 64 7 \\
10432 > $(RM_6_GENERATOR_1) \\
10433 > $(RM_6_GENERATOR_2) \\
10434 > $(RM_6_GENERATOR_3) \\
10435 > $(RM_6_GENERATOR_4) \\
10436 > $(RM_6_GENERATOR_5) \\
10437 > $(RM_6_GENERATOR_6) \\
10438 > $(RM_6_GENERATOR_7) \\
10439 > $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix \\
10440 > -input_csv_file long_code_genma_n64_k7.csv \\
10441 > -box_width 25 -bit_depth 8 -partition 3 7 64 -end \\
10442 > $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix \\
10443 > -input_csv_file long_code_genma_n64_k7_codeword_0.csv \\
10444 > -box_width 25 -bit_depth 8 -partition 3 8 8 -end \\
10445 > $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix \\
10446 > -input_csv_file long_code_genma_n64_k7_codeword_1.csv \\
10447 > -box_width 25 -bit_depth 8 -partition 3 8 8 -end \\
10448 > $(ORBITER_PATH)orbiter.out -v 2 -draw_matrix \\
10449 > -input_csv_file long_code_genma_n64_k7_codeword_2.csv \\
10450 > -box_width 25 -bit_depth 8 -partition 3 8 8 -end \\
10451
10452
10453
10454
10455 RM6words:
10456 > -mkdir-RM6 \\
10457 > #$\$(ORBITER_PATH)orbiter.out -v 2 -draw_matrix -input_csv_file long_code_genma_n \\
64_k7_codeword_0.csv -box_width 25 -bit_depth 8 -partition 4 8 8 -end \\
10458 > $(ORBITER_PATH)orbiter.out -v 2 -loop L 0 128 1 \\
10459 > -draw_matrix -input_csv_file long_code_genma_n64_k7_codeword %L.csv \\
10460 > -box_width 25 -bit_depth 8 -partition 4 8 8 -end \\
10461 > -mv long_code_genma_n64_k7_codeword %L_draw.bmp RM6/RM_6_1_codeword %L.bmp \\
10462 > -end_loop \\
10463
10464
10465
10466 RM6.convert:
10467 > -mkdir-RM6.png \\
10468 > convert RM6/RM_6_1_codeword_0.bmp -frame 8 RM6.png/000.png \\
10469 > convert RM6/RM_6_1_codeword_1.bmp -frame 8 RM6.png/001.png \\
10470 > convert RM6/RM_6_1_codeword_2.bmp -frame 8 RM6.png/002.png \\
10471 > convert RM6/RM_6_1_codeword_3.bmp -frame 8 RM6.png/003.png \\
10472 > convert RM6/RM_6_1_codeword_4.bmp -frame 8 RM6.png/004.png
### Section 9.6: Coding Theory

#### F2 BCH Code n21:

```bash
$(ORBITER_PATH)orbiter.out -v 3
```

```bash
  -define F -finite_field -q 2 -end:
  -with F -do -finite_field_activity -make_BCH_code 21 3 -end
```

#### F7 RS Code n6:

```bash
$(ORBITER_PATH)orbiter.out -v 30
```

```bash
  -define F -finite_field -q 7 -end:
  -with F -do -finite_field_activity -make_BCH_code 6 3 -end
```
BCH_255_5_evaluate_elementary_symmetric_functions_1:

BCH_255_5_evaluate_elementary_symmetric_functions_2:

BCH15:

BCH11:
10683  $(ORBITER\_PATH)\$orbiter.out--BCH:11:2:3.
10684  $(ORBITER\_PATH)\$orbiter.out--BCH:11:2:5.
10685  
10686  
10687  BCH13:
10688  $(ORBITER\_PATH)\$orbiter.out--BCH:13:2:3.
10689  $(ORBITER\_PATH)\$orbiter.out--BCH:13:2:5.
10690  
10691  
10692  BCH7:
10693  $(ORBITER\_PATH)\$orbiter.out--BCH:7:2:3.
10694  $(ORBITER\_PATH)\$orbiter.out--BCH:7:2:5.
10695  
10696  
10697  BCH21:
10698  $(ORBITER\_PATH)\$orbiter.out--BCH:21:2:3.
10699  $(ORBITER\_PATH)\$orbiter.out--BCH:21:2:5.
10701  
10702  
10703  
10704  BCH93:
10705  $(ORBITER\_PATH)\$orbiter.out--BCH:93:2:3
10706  
10707  BCH255:
10708  $(ORBITER\_PATH)\$orbiter.out--BCH:255:2:4
10709  $(ORBITER\_PATH)\$orbiter.out--v.2--draw_matrix-
10710  -input_csv_file-BCH_255_4.csv-
10711  -box_width:40.-bit_depth:24.-partition:10."239"."255".--end
10712  
10713  #BCH_255_4.csv
10714  
10715  
10716  BCH273:
10717  $(ORBITER\_PATH)\$orbiter.out--BCH:273:2:4
10718  
10719  
10720  
10721  
10722  
10723  draw_mod:31:
10724  $(ORBITER\_PATH)\$orbiter.out--v.2-
10725  -draw_options--embedded--end-
10726  -draw_mod_n:31:mod:31--draw_mod_n_power_cycle:2
10727  pdflatex:mod_31_draw.tex
10728  open:mod_31_draw.pdf
10729  
613
PR127: $(\text{ORBITER PATH})\text{orbiter.out}$ · -v · \text{-primitive_root:127}

draw_mod_127_power:

$\text{draw_options: scale 0.4: -embedded: -end:}$

$\text{draw_mod_n:127-mod_127:-draw_mod_n_power_cycle:3:}$

$\text{pdflatex-mod_127_drawn.tex}$

open-mod_127_drawn.pdf

draw_mod_251:

$\text{draw_options: nodes_empty: -radius:10: -embedded: -end:}$

$\text{draw_mod_n:251-mod_251:}$

$\text{pdflatex-mod_251_drawn.tex}$

open-mod_251_drawn.pdf

#-draw_mod_n_inverse

draw_mod_255_cyclotomic_1:

$\text{draw_options: nodes_empty: -radius:10:}$

$\text{draw_mod_n:n:255:-file:mod_255_cyclotomic_1:}$

$\text{pdflatex-mod_255_cyclotomic_1_drawn.tex}$

open-mod_255_cyclotomic_1_drawn.pdf

draw_mod_255_cyclotomic_3:

$\text{draw_options: nodes_empty: -radius:10:}$

$\text{draw_mod_n:n:255:-file:mod_255_cyclotomic_3:}$

$\text{pdflatex-mod_255_cyclotomic_3_drawn.tex}$

open-mod_255_cyclotomic_3_drawn.pdf

draw_mod_255_cyclotomic_1_and_3:

$\text{draw_options: nodes_empty: -radius:10:}$

$\text{draw_mod_n:n:255:-file:mod_255_cyclotomic_1_and_3:}$

$\text{pdflatex-mod_255_cyclotomic_1_and_3_drawn.tex}$
draw_mod_63_4_cyclotomic_3_6:

$\text{(ORBITER\ PATH)}\ orbiter.out -v.2 -draw_mod_n-n.63 -n.63$-mod-63.4-cyclotomic.3_6-
$\text{-draw_options -radius.20 \ line_width.0.1 -embedded \ -end}$
$\text{-cyclotomic_sets psychotic.3_6}$
$\text{-cyclotomic_sets thickness psychotic.50}$
$\text{-end}$
$\text{pdflatex\ mod_63_4_cyclotomic_3_6\ draw.pdf}$

BCH_F.64:
$\text{(ORBITER\ PATH)}\ orbiter.out -v.3 -define\ F\ finite_field\ q.64\ end$
$\text{-with\ F\ do\ finite_field activity cheat_sheet GF \end}$
$\text{pdflatex GF.64.tex}$

BCH_elementary_symmetric_functions.3:
$\text{(ORBITER\ PATH)}\ orbiter.out -make elementary_symmetric_functions.3 -3$
$\text{pdflatex mod_63_4_cyclotomic_3_6 draw.pdf}$

$\text{open-mod_63_4_cyclotomic_3_6 draw.pdf}$

10811 #The values of the formulae are:
10812 #0: 57
10813 #1: 0
10814 #2: 1
10815
10819 BCH_63_4_evaluate_elementary_symmetric_functions.2:
$\text{(ORBITER\ PATH)}\ orbiter.out -v.3 -define\ F\ finite_field\ q.64\ end$
$\text{-define\ e1 -formula \$(ELEMENTARY\ SYMMETRIC.3.1) \end}$
$\text{-define\ e2 -formula \$(ELEMENTARY\ SYMMETRIC.3.2) \end}$
$\text{-define\ e3 -formula \$(ELEMENTARY\ SYMMETRIC.3.3) \end}$
$\text{-with\ F\ do\ finite_field activity \end}$
$\text{-evaluate E3：x0=8, x1=62, x2=15 \end}$
define \( E_3 \) -collection: \( "e_1, e_2, e_3" \)

with \( F \) -do -finite_field_activity:

- evaluate \( E_3 \): \( x_0=33, x_1=45, x_2=52 \)

The values of the formulae are:

0: \( 56 \)

1: \( 0 \)

2: \( 1 \)

\( \text{poly}: 1,0,3,1 \)

BCH_21_poly_mult_mod_F4:

\( \text{poly}: 1,0,1,0,1,1,1 \)

BCH_21_poly_division_a:

\( \text{poly:} 1,0,1,0,1,1,1 \)

BCH_21_poly_division_b:

\( \text{poly:} 1,0,1,0,1,1,1 \)

BCH_21_poly_division_ab:
BCH\_21 generator matrix:

\[ \text{(ORBITER PATH)} \text{orbiter.out} \]

BCH\_21\_15 weight enumerator:

\[ \text{(ORBITER PATH)} \text{orbiter.out} \]

BCH\_21\_6 weight enumerator:
\begin{verbatim}
}y^3
10914
10915 \( \#(-1,0,0,0,0,0,0,0,0,0,0,0,0.294,0.0.756,0,0.1890,0,0.1092,0,0,0,0) \)
10916
10917
10918
10919 BCH_21_64_macwilliams:
10920 \>$\$(ORBITER_PATH)orbiter.out-v.2-make_macwilliams_system-21.6.4$
10921 \>$pdflatex-MacWilliams_n21_k6_q4.tex$
10922 \>$open-MacWilliams_n21_k6_q4.pdf$
10923
10924
10925
10926 \$ww:=[1,0,0,-84,152,1575,10080,58032,-319662,-1411116,5133744,15282792,$
10927
10928
10929 BCH_21_15_4_field_reduction:
10930 \>$\$(ORBITER_PATH)orbiter.out-v.2-$
10931 \>$-define F-finite_field-q.4-end$
10932 \>$-with F-do$
10933 \>$-finite_field_activity$
10934 \>$-field_reduction\"BCH_21_15_4\"-2.15.21-$\$(BCH_21_15)\$-end$
10935 \>$\$(ORBITER_PATH)orbiter.out-v.2-$
10936 \>$-draw_matrix-input_csv_file-BCH_21_15_4.csv$
10937 \>$-box_width.20-bit_depth.24$
10938 \>$-partition.4\"30\".42-end$
10939 \>$pdflatex-field_reduction_Q4_q2_15_21.tex$
10940 \>$open-BCH_21_15_4_draw.bmp$
10941 \>$#open-field_reduction_Q4_q2_15_21.pdf$
10942
10943 \$poly_of_degree.12:1,0,1,0,0,0,0,1,0,0,0,1$
10944
10945 BCH_21_poly_division_c:
10946 \>$\$(ORBITER_PATH)orbiter.out-v.2-$
10947 \>$-define F-finite_field-q.2-end$
10948 \>$-with F-do$
10949 \>$-finite_field_activity$
10950 \>$-polynomial_division\"1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1\".1,0,1,0,1,0,0,0,1,0,0,0,1$-end
10951
10952
10953 F256_roots_771:
10954 \>$\$(ORBITER_PATH)orbiter.out-v.3-$
10955 \>$-define F-finite_field-q.256-end$
10956 \>$-with F-do-finite_field_activity-nth_roots.771-end$
\end{verbatim}
10957
10958
10959
10960  F256_BCH_code_d16:
10961  \$(${\text{ORBITER}}_{\text{PATH}})\text{orbiter.out}\ -v.3\
10962  \triangleright \  -\text{define}\ -F:\ -\text{finite_field}\ -q\ :256:\ -end\
10963  \triangleright \  -\text{with}\ -F:\ -do:\ -\text{finite_field}\ -activity:\ -\text{make_BCH_code}\ -771\ -16:\ -end\
10964
10965  \#\text{generator\-polynomial: is }X^\{30\} + 253 X^\{29\} + 174 X^\{28\} + 109 X^\{27\} + 97 X^\{26\} + 144 X^\{25\} + 112 X^\{24\} + 212 X^\{23\} + 192 X^\{22\} + 169 X^\{21\} + 24 X^\{20\} + 150 X^\{19\} + 110 X^\{18\} + 248 X^\{17\} + 3 X^\{16\} + 193 X^\{15\} + 194 X^\{14\} + 205 X^\{13\} + 9 X^\{12\} + 56 X^\{11\} + 95 X^\{10\} + 199 X^\{9\} + 108 X^\{8\} + 58 X^\{7\} + 160 X^\{6\} + 148 X^\{5\} + 138 X^\{4\} + 24 X^\{3\} + 210 X^\{2\} + 26 X + 1
10966
10967
10968
10969
10970
10971  F256_BCH_code_d16\_division:
10972  \$(${\text{ORBITER}}_{\text{PATH}})\text{orbiter.out}\ -v.2\
10973  \triangleright \  -\text{define}\ -F:\ -\text{finite_field}\ -q\ :256:\ -end\
10974  \triangleright \  -\text{define}\ -A:\ -\text{vector}\ -\text{field}\ F:\ -\text{sparse} \ 771,1,0\ -\text{end}\
10975  \triangleright \  -\text{define}\ -B:\ -\text{vector}\ -\text{field}\ F:\ -\text{dense} \ (POLY\_Q256\_DEG30\_DENSE)\ -\text{end}\
10976  \triangleright \  -\text{with}\ -F:\ -do\
10977  \triangleright \  -\text{finite_field}\ -activity\
10978  \triangleright \  -\text{polynomial}\ -\text{division}\ -A\ -B\ -\text{end}\
10979
10980  F256_BCH_write\_code\_for\_division\_d16:
10981  \$(${\text{ORBITER}}_{\text{PATH}})\text{orbiter.out}\ -v.2\
10982  \triangleright \  -\text{define}\ -F:\ -\text{finite_field}\ -q\ :256:\ -end\
10983  \triangleright \  -\text{define}\ -A:\ -\text{vector}\ -\text{field}\ F:\ -\text{sparse} \ 771,1,0\ -\text{end}\
10984  \triangleright \  -\text{define}\ -B:\ -\text{vector}\ -\text{field}\ F:\ -\text{dense} \ (POLY\_Q256\_DEG30\_DENSE)\ -\text{end}\
10985  \triangleright \  -\text{with}\ -F:\ -do\
10986  \triangleright \  -\text{finite_field}\ -activity\
10987  \triangleright \  -\text{write}\ -\text{code}\ -\text{for}\ -\text{division}\ -\text{check} \ q256\_n771\_r30.\cpp\ -A\ -B\ -\text{end}\
10988  \$g++ -check\ q256\_n771\_r30.\cpp\ -o\ -check\ q256\_n771\_r30.\out
10989  ./check\ q256\_n771\_r30.\out
10990
10991
10992  F256_BCH_code_d16\_error:
10993  \$(${\text{ORBITER}}_{\text{PATH}})\text{orbiter.out}\ -v.2\
10994  \triangleright \  -\text{define}\ -F:\ -\text{finite_field}\ -q\ :256:\ -end\
10995  \triangleright \  -\text{define}\ -A:\ -\text{vector}\ -\text{field}\ F:\ -\text{sparse} \ 771,2,3,31,55,770\ -\text{end}\
10996  \triangleright \  -\text{define}\ -B:\ -\text{vector}\ -\text{field}\ F:\ -\text{dense} \ (POLY\_Q256\_DEG30\_DENSE)\ -\text{end}\
10997  \triangleright \  -\text{with}\ -F:\ -do\
10998  \triangleright \  -\text{finite_field}\ -activity\
10999  \triangleright \  -\text{polynomial}\ -\text{division}\ -A\ -B\ -\text{end}\

619
### Section 19.7: Coding Theory -- Reed-Solomon Codes

#### SECTION CODING THEORY REED SOLOMON CODES:

**Code RS 6**:  
```plaintext
$\text{RREF}_\text{RS}_6$ 
```

**weight enumerator:**
```plaintext
\text{define } F = \text{finite field } q = 7 \text{ -end}\n
\text{define } v = \text{vector } -\text{format } 4 -\text{field } F \text{-with } F \text{-do}\n
\text{weight enumerator v -end}
```

Code RS 11:  
```plaintext
$\text{RREF}_\text{RS}_11$ 
```

**weight enumerator:**
```plaintext
\text{define } F = \text{finite field } q = 11 \text{ -end}\n
\text{define } v = \text{vector } -\text{format } 8 -\text{field } F \text{-with } F \text{-do}\n
\text{weight enumerator v -end}
```
11045  ▶  ▶  ▶  -compact\$(CODE_RS_11_RREF)\$
11046  ▶  ▶  -end\$
11047  ▶  ▶  -with-F\-do\$
11048  ▶  ▶  -finite_field_activity\$
11049  ▶  ▶  ▶  -weight Enumerator\v$
11050  ▶  ▶  -end
11051
11052
11053  #1*y^\(10\)-1.200*x^3*y^7-1.6800*x^4*y^6-2.09160*x^5*y^5+1.734600*x^6*y^4-0.991
8000*x^7*y^3+3.7189800*x^8*y^2+8.2644700*x^9*y+8.2644620*x\^\(10\)
11054
11055
11056  RREF RS_8_weight Enumerator:
11057  ▶  \$(ORBITER_PATH)orbiter.out\-v\-2\$
11058  ▶  ▶  -define-F\-finite_field\-q\-8\-end\$
11059  ▶  ▶  -define-v\-vector\-format\-5\-field-F$
11060  ▶  ▶  ▶  -compact\$(CODE_RS_8)$\$
11061  ▶  ▶  -end\$
11062  ▶  ▶  -with-F\-do\$
11063  ▶  ▶  -finite_field_activity\$
11064  ▶  ▶  ▶  -weight Enumerator\v$
11065  ▶  ▶  -end
11066
11067
11068  #the group cannot be computed
11069
11070  RS_8_field_reduction:
11071  ▶  \$(ORBITER_PATH)orbiter.out\-v\-2\$
11072  ▶  ▶  -define-F\-finite_field\-q\-8\-end\$
11073  ▶  -with-F\-do\$
11074  ▶  -finite_field_activity\$
11075  ▶  -field_reduction\"RS_8_red_2\"$
11076  ▶  ▶  ▶  2\cdot 7\$(CODE_RS_8)$\$
11077  ▶  ▶  -end
11078  ▶  \$(ORBITER_PATH)orbiter.out\-v\-2\$
11079  ▶  ▶  -draw_matrix\-input_csv_file\RS_8_red_2.csv$
11080  ▶  ▶  -box_width 40\-bit_depth 24\$
11081  ▶  ▶  -partition\":4\"3,3,3,3\"\"3,3,3,3,3,3\"\-end
11082  ▶  pdflatex field_reduction_Q8_q\_2\_5\_7.tex
11083  ▶  open field_reduction_Q8_q\_2\_5\_7.pdf
11084
11085
11086  RREF RS_8_reduced weight Enumerator:
11087  ▶  \$(ORBITER_PATH)orbiter.out\-v\-2\$
11088  ▶  ▶  -define-F\-finite_field\-q\-2\-end\$
11089  ▶  ▶  -define-v\-vector\-format\-15\-field-F$
11090  ▶  ▶  ▶  -compact\$(RS_8_reduced)$\$

621
CODE_21_15_4_store:
$\text{(ORBITER PATH)}$ orbiter.out-v-2:\n\> -store_as_csv_file:"code_21_15_4.csv":\n\> 15:21-$\text{(CODE_21_15_4)}$.
\> $(\text{ORBITER PATH)}$ orbiter.out-v-2-draw_matrix:\n\> -input_csv_file:code_21_15_4.csv:\n\> -box_width:40-bit_depth:24:\n\> -partition-4:"15":"21":\n\> -end
\> -end\n
CODE_21_15_4_weight Enumerator:
$\text{(ORBITER PATH)}$ orbiter.out-v-2:\n\> -define-F-finite_field-q:2-end\n\> -define-v-vector-format:15-field:F\n\> -compact-$\text{(CODE_21_15_4)}$\n\> -end\n\> -with-F-do\n\> -finite_field_activity:\n\> -weight Enumerator-v\n\> -end

Reed solomon F8 work:
$\text{(ORBITER PATH)}$ orbiter.out-v:3\n\> -define-F-finite_field-q:8-end\n\> -with-F-do-finite_field_activity\n\> -parse_and_evaluate:"test".""(t-a)*(t-a*a)"."a=2"-end

bounds for d given n6,k4,q7:
\> $\text{(ORBITER PATH)}$ orbiter.out-v-2:\n\> -make_bounds_for_d_given_n_and_k_and_q:6:4:7

SECTION_CODING_THEORY_BOUNDS:

# Section 9.8: Coding Theory - Bounds

bounds for d given n6,k4,q7:
bounds_for_d_given_n15_k6_q2:

```
$ (ORBITER_PATH) orbiter.out -v 2 \n  -make_bounds_for_d_given_n_and_k_and_q 15 6 2
```

```
# n = 15 k = 6 q = 2
# d_{GV} = 5
# d_{singleton} = 10
# d_{hamming} = 6
# d_{plotkin} = 7
# d_{griesmer} = 6
```

```
coding_theory_bounds_q2:
```

```
$ (ORBITER_PATH) orbiter.out -v 2 -table_of_bounds 20 2
```

```
# produces table_of_bounds_n20_q2.csv
```

```
coding_theory_bounds_q8:
```

```
$ (ORBITER_PATH) orbiter.out -v 2 -table_of_bounds 20 8
```

```
GV_n15_k6_d5:
```

```
$ (ORBITER_PATH) orbiter.out -v 2 \n  -define F finite_field q 2 -end \n  -define P projective_space 8 F -end \n  -with P do \n  -projective_space_activity make_gilbert_varshamov_code 15 5 -end
```

```
# [15,6] code created
```

```
bounds_for_d_given_n12_k4_q13:
```

```
$ (ORBITER_PATH) orbiter.out -v 2 \n  -make_bounds_for_d_given_n_and_k_and_q 12 4 13
```

```
GV_n15_k6_d5.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```

623
# finite_field_activity

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

```bash
$ define-F-finite_field -q:2 -end
```

```bash
$ define-v-vector -format:6 -field:F -end
```

```bash
$ compact $(CODE_15_6_6_A) -end
```

```bash
$ weight Enumerator
```

```bash
$ compact $(CODE_15_6_6_A) -end
```

```bash
$ with-F-do -end
```

```bash
$ finite_field_activity -weight Enumerator -v -end
```

```bash
$ weight Enumerator
```

```bash
$ compact $(CODE_15_6_6_A) -end
```

```bash
$ with-F-do -end
```

```bash
$ pdflatex RREF_example_q2_6_15.tex
```

```bash
$ open RREF_example_q2_6_15.pdf
```

```bash
$ pdflatex RREF_example_q2_9_15.tex
```

624
Section 9.9: Coding Theory -- Classification

# code-classification:
codes_8_4_4:

$(ORBITER_PATH)orbiter.out
-orbiter_path:$(ORBITER_PATH)"-define G"
-linear_group:-PGL4:2-"-with G\-do"
group_theoretic_activity:
-poset_classification_control:
-problem_label:codes_8_4_4:
draw_poset:
draw_options:--embedded--radius250:
-line_width1.0--spanning_tree--end:
-report--end:

-end
-linear_codes:3\-8:
-end

pdflatex codes_8_4_4_poset_lvl_8.tex
open-codes_8_4_4_poset_lvl_8.pdf
pdflatex codes_8_4_4_poset.tex
open-codes_8_4_4_poset.pdf

codes_8_4_4_draw:

$(ORBITER_PATH)orbiter.out
-draw_layered_graph:codes_8_4_4_poset_lvl_8.layered_graph:
-radius250--embedded--line_width1.0:
-y_stretch1.0--scale0.5:
-end

pdflatex codes_8_4_4_poset_lvl_8_draw.tex
open-codes_8_4_4_poset_lvl_8_draw.pdf
codes_14_4_9_3:

$(\text{ORBITER\_PATH})\texttt{orbiter.out} -v 6 -define G -linear_group -PGL 10 3 -end -with G -do -group_theoretic_activity -poset_classification_control -problem_label codes_14_4_9_3 -draw poset -draw_options -embedded -radius 250 -end -linear codes 9 14 -end

pdflatex codes_14_4_9_3 poset lvl 13.tex
open codes_14_4_9_3 poset lvl 13.pdf


codes_15_6_6_2:

$(\text{ORBITER\_PATH})\texttt{orbiter.out} -v 6 -define G -linear_group -PGL 9 2 -end -with G -do -group_theoretic_activity -poset_classification_control -problem_label codes_15_6_6_2 -draw poset -draw_options -embedded -radius 250 -end -linear codes 6 15 -end

pdflatex codes_15_6_6_2 poset lvl 15.tex
open codes_15_6_6_2 poset lvl 15.pdf


codes_16_5_9_3:

$(\text{ORBITER\_PATH})\texttt{orbiter.out} -v 6 -codes classify -n 16 -k 5 -q 3 -d 9 -w W -lex -draw poset -end

# 5/31/2020: 28 min 22 sec on Mac

0::1 orbits

1::1 orbits
codes_d4:
  \$\text{(ORBITER\_PATH)\orbiter.out-\text{-v.3}}\$
  \$\text{-define-G-\text{-linear_group-\text{-PGL-4\cdot2-\text{-end}}}\}$
  \$\text{-with-G-\text{-do}}\$
  \$\text{-group_theoretic_activity}\$
  \$\text{-poset\_classification\_control-\text{-W} \text{-end}}\$
  \$\text{-problem\_label\_codes_r4\_d4\_\text{-draw\_poset}}\$
  \$\text{-embedded\_\text{-end\_linear\_codes-4\cdot100}}\$
  \$\text{-end}\$
  \$\text{-end}\$

codes_24_12_8:
  \$\text{(ORBITER\_PATH)\orbiter.out-\text{-v.6}}\$
  \$\text{-orbiter\_path-\text{(ORBITER\_PATH)}}\$
  \$\text{-define-G}\$
  \$\text{-linear\_group-\text{-PGL-12\cdot2-\text{-end}}}\$
  \$\text{-with-G-\text{-do}}\$
  \$\text{-group\_theoretic\_activity}\$
  \$\text{-poset\_classification\_control-\text{-problem\_label\_codes-24_12_8}}\$
  \$\text{-draw\_poset} \text{-draw\_options-\text{-embedded-\text{-radius-250}}}\$
  \$\text{-line\_width-1.0-\text{-spanning\_tree-\text{-end}}}\$
  \$\text{-report-\text{-end}}\$
  \$\text{-end}\$
  \$\text{-end}\$
  \$\text{pdflatex\_codes-24_12_8\_poset\_tex}$
  \$\text{open\_codes-24_12_8\_poset\_pdf}$

627
11371
11372 #codes_24_12_8_poset_lvl_24.layered_graph
11373
codes_24_12_8.draw:
11375 > $(ORBITER_PATH)orbiter.out-v.3\  
11376 | > -draw.layered_graph-codes_24_12_8_poset_lvl_24.layered_graph\ 
11377 | > -radius.100.-spanning_tree.-embedded-\ 
11378 | > -line_width.0.5.-x_stretch.1.4.-scale.0.25.-nodes_empty-\ 
11379 | > -end-\ 
11380 | > pdflatex-codes_24_12_8_poset_lvl_24.draw.tex
11381 | > open-codes_24_12_8_poset_lvl_24_draw.pdf
11382
glynn_arc:
11384  > $(ORBITER_PATH)orbiter.out-v.5-\ 
11385  | > -orbiter_path$(ORBITER_PATH)-\ 
11386  | > -define-G-\ 
11387  | > -linear_group-PGGL.5.9.-end-\ 
11388  | > -with-G-do-\ 
11389  | > -group.theoretic_activity-\ 
11390  | > -poset_classification_control.-problem_label-glynn_arc-\ 
11391  | > -draw_options.-embedded.-radius.250-\ 
11392  | > -line_width.1.0.-spanning_tree.-end-\ 
11393  | > -draw_poset-\ 
11394  | > -report.-end-\ 
11395  | > -end-\ 
11396  | > -linear_codes.6.10-\ 
11397  | > -end-\ 
11398  | > pdflatex_glynn_arc_poset.tex
11399  | > open-glynn_arc_poset.pdf
11400
11401
11402
11403
11404
five_points_in_general:
11406  > $(ORBITER_PATH)orbiter.out-v.5-\ 
11407  | > -orbiter_path$(ORBITER_PATH)-\ 
11408  | > -define-G-\ 
11409  | > -linear_group-PGL.4.2.-end-\ 
11410  | > -with-G-do-\ 
11411  | > -group.theoretic_activity-\ 
11412  | > -poset_classification_control-\ 
11413  | > -problem_label-five_points_in_general-\ 
11414  | > -draw_options.-embedded.-radius.250-\ 
11415  | > -line_width.1.0.-spanning_tree.-end-\ 
11416  | > -draw_poset-\ 
11417  | > -report.-end-\ 

628
Chapter 10 - Incidence Geometry

Section 10.1: Diophantine Systems

part10:

$(\text{ORBITER\_PATH})\text{orbiter.out}-v.4$

-define D-diophant-label-part10
octic_monomials:
  $\text{(ORBITER PATH)orbiter.out}\ -v\ -4\$
  -define\ D\ =\ \text{diophant}\ .\ \text{label}\ .\ octic\_monomials\$
  coefficient\ matrix\ =\ 1\ 6\ \{21,15,10,6,3,1\}\$
  \text{RHS} =\ \{21,21,1\}\$
  -x_{\text{min}}\ =\ 0\$
  -x_{\text{max}}\ =\ 21\$
  -end$
  with\ D\ -do$
  diophant\ .\ activity\ -solve\_mckay$
  -end$

sort\ -r\ .\ octic\_monomials\ .sol\ >\ octic\_monomials\ .sorted\ .txt

#Found\ 165\ solutions\ with\ 210\ backtrack\ steps
# 165 = \text{binomial}(11,3)

# Section 10.2: Combinatorial Linear Spaces
SECTION COMBINATORIAL LINEAR SPACES:
linsp7:
  $\text{(ORBITER PATH)orbiter.out}\ -v\ -4\$
  -define\ D\ =\ \text{diophant}\ .\ \text{label}\ .\ linsp7\$
  coefficient\ matrix\ =\ 1\ 6\ \{21,15,10,6,3,1\}\$
  \text{RHS} =\ \{21,21,1\}\$
  -x_{\text{min}}\ =\ 0\$
  -x_{\text{max}}\ =\ 21\$
  -end$
11508 ◁ -with-D-do-\n11509 ◁ ◁ -diophant_activity-solve_mckay-\n11510 ◁ ◁ -end\n11511 ◁
11512
11513 #32-solutions-in-45-backtrack-steps
11514
11515
11516 linsp6:
11517 ◁ $(ORBITER_PATH)orbit.out-\v-4-\n11518 ◁ ◁ -define-D-diophant-label-lin\n11519 ◁ ◁ -coefficient_matrix-1.5-15,10,6,3,1-\n11520 ◁ ◁ -RHS-15,15,1-\n11521 ◁ ◁ -x_min_global-0-\n11522 ◁ ◁ -x_max_global-15-\n11523 ◁ ◁ -end-\n11524 ◁ ◁ -with-D-do-\n11525 ◁ ◁ ◁ -diophant_activity-solve_mckay-\n11526 ◁ ◁ -end\n11527 ◁
11528 #Found-15-solutions-with-22-backtrack-steps
11529
11530
11531
11532
11533
11534 linsp30_pt_types:
11535 ◁ $(ORBITER_PATH)orbit.out-\v-4-\n11536 ◁ ◁ -define-D-\n11537 ◁ ◁ -diophant-label-lin\n11538 ◁ ◁ -coefficient_matrix-1.3-6,4,3-\n11539 ◁ ◁ -RHS-29,29,1-x_bounds-0,1,0,27,0,24-\n11540 ◁ ◁ -end-\n11541 ◁ ◁ -with-D-do-\n11542 ◁ ◁ ◁ -diophant_activity-solve_mckay-\n11543 ◁ ◁ -end\n11544
11545
11546
11547
11548
11549
11550
11551
11552
11553
11554
11555
631
Section 10.3: Combinatorial Linear Spaces

geometry.builder:

geo_petersen:

$\text{(ORBITER\_PATH)orbiter.out-v.8}\$

-define Test_lines -set -loop 3.11.1 -end

-define Geo -geometry_builder

-define V -7 -B -15 -TDO -3 -fuse -1

-fname GEO -petersen -girth 5

-search_tree

test Test_lines FFFF

-define Geo -geometry_builder

-V -10 -B -15 -TDO -3 -fuse -1

-fname GEO -petersen -girth 5

-search_tree

test Test_lines FFFF

end

geo_7_3:

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

-define Test_lines -set -loop 3.8.1 -end

-define Geo -geometry_builder

-V -7 -B -7 -TDO -3

-fuse -1 -fname GEO -7_3

-test Test_lines FFFF

-end

geo_8_3:

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

-define Test_lines -set -loop 3.9.1 -end

-define Geo -geometry_builder

-V -8 -B -8 -TDO -3

-fuse -1 -fname GEO -8_3

-test Test_lines FFFF

-end

-print_at_line 4

#1:geo:0.11:18:29:30:38:44:54
geo_9_3:

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

-define Test_lines-set-loop-3.10.1-end
-define Geo-geometry_builder
-V.9-B.9-TDO.3
-fuse.1-fname_GEO.9.3
-test Test_lines-FFFF

-define Geo-geometry_builder
-V.9-B.9-TDO.3
-fuse.1-fname_GEO.10.3
-test Test_lines-FFFF
-f_orderly
-end
```

test

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

-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
-test Test_lines-FFFF
```

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

-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-FFFF
-end
```

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

-draw_options-embedded-radius 50
-xin-100000-yin-10000
-xout-1000000-yout-500000
-nodes_empty
-scale 0.5-line_width 0.3
```
11648 \-end\n11649 \-tree_draw.10.3_tree.txt
11650 pdflatex:10.3_tree_draw.tex
11651 open:10.3_tree_draw.pdf
11652
11653 #-sideways-
11654
11655
11656
11657
11658
11659
11660 geo_11.3:
11661 \$(ORBITER_PATH)orbiner.out-v.2\n11662 \-define-Test_lines-set-loop4.12.1.-end\n11663 \-define-Geo-geometry_builder\n11664 \-V.11.-B.11.-TD0.3\n11665 \-fuse.1.-fname_GEO.11.3\n11666 \-test-Test_lines-FFFF.\n11667 \-end
11668
11669 #.31.geos
11670 #8/26/2021:0.00.on.Mac
11671
11672 geo_12.3:
11673 \$(ORBITER_PATH)orbiner.out-v.2\n11674 \-define-Test_lines-set-loop4.13.1.-end\n11675 \-define-Geo-geometry_builder\n11676 \-V.12.-B.12.-TD0.3\n11677 \-fuse.1.-fname_GEO.12.3\n11678 \-test-Test_lines-FFFF.\n11679 \-end
11680
11681 #.229.geos
11682 #User.time:-0.45.of.a.second,dt=45.tps=100
11683 #nb_calls_to_densenauty=24586
11684
11685
11686
11687
11688
11689 geo_13.3:
11690 \$(ORBITER_PATH)orbiner.out-v.2\n11691 \-define-Test_lines-set-loop4.14.1.-end\n11692 \-define-Geo-geometry_builder\n11693 \-V.13.-B.13.-TD0.3\n11694 \-fuse.1.-fname_GEO.13.3\n
634
11695 \> \> \> -test-Test_lines-FFFF-\n11696 \> \> -end
11697 11698 #2036:geos,96,39,13,12^4,8^3,6^16,4^30,3^20,2^190,1^1770
11699 #User-time:0:4
11700 #nb_calls_to_densenauty=216777
11701 11702 11703 geo_14_3:
11704 \> $(ORBITER_PATH)orbiter.out--v:2\n11705 \> \> -define-Test_lines-set-loop4:15:1-end\n11706 \> \> -define-Geo-geometry_builder\n11707 \> \> \> -V:14-B:14-TDO:3\n11708 \> \> \> -fuse1-fname GEO:14_3\n11709 \> \> \> -test-Test_lines-FFFF\n11710 \> \> -end
11711 11712 #21399:geos,56448,128,24^2,16^3,14^3,12^7,8^15,7,6^12,4^91,3^19,2^91
6,1^20328
11713 #User-time:0:55
11714 #nb_calls_to_densenauty=2089344
11715 11716 11717 15_3.inc:
11718 15_3.inc:
11719 \> $(ORBITER_PATH)orbiter.out--v:2\n11720 \> \> -define-Test_lines-set-loop4:16:1-end\n11721 \> \> -define-Geo-geometry_builder\n11722 \> \> \> -V:15-B:15-TDO:3\n11723 \> \> \> -fuse1-fname GEO:15_3\n11724 \> \> \> -test-Test_lines-FFFF\n11725 \> \> -end
11726 11727 #245342:geos,8064,720,192^2,128,72,48^6,32,30^2,24^2,20^2,18,16^10,15^2,12^11,10^3,8^34,6^59,5^5,4^180,3^69,2^3709,1^241240
11728 #8-min:11-sec-on-Mac
11729 #running:out-of-memory
11730 11731 11732 geo_15_3_g4:
11733 \> $(ORBITER_PATH)orbiter.out--v:2\n11734 \> \> -define-Test_lines-set-loop4:16:1-end\n11735 \> \> -define-Geo-geometry_builder\n11736 \> \> \> -V:15-B:15-TDO:3\n11737 \> \> \> -fuse1-fname GEO:15_3_g4\n11738 \> \> \> -girth4\n11739 \> \> \> -search_tree\n
635
# The unique Cremona-Richmond configuration with group of order 720

User time: 0.15 of a second, dt=15 tps=100

nb_calls_to_densenauty=6767

# 4 objects

User time: 0.15 of a second, dt=15 tps=100

nb_calls_to_densenauty=6767

4 objects
-girth 4 \  
-test.Test_lines.FFFF \  
-end  

#none  

40.4_g4.inc:  
  $(ORBITER_PATH)orbiter.out-v.2 \  
  -define:Test_lines-set:-loop-0.41:1-end \  
  -define:Geo:geometry_builder \  
  -V-40-B-40-TDO-4 \  
  -fuse:fname_GEO-40.4_g4 \  
  -girth 4 \  
  -search_tree \  
  -test.Test_lines.FFFF \  
  -end  

$(ORBITER_PATH)orbiter.out-v.2 \  
-draw_options:embedded:radius:50 \  
-xin:10000-yin:10000 \  
-xout:1000000-yout:1000000 \  
-nodes_empty \  
-scale:0.5-line_width:0.3-end \  
-tree_draw-40.4_g4_tree.txt \  
pdflatex-40.4_g4_tree_draw.tex \  
open-40.4_g4_tree_draw.pdf  

#2:geos,-ago=51840^2 \  
#User time:0.18 of a second, dt=18 tps=.100 \  
#nb_calls_to_densenaught=1065  

geo_63_3_g6:  
  $(ORBITER_PATH)orbiter.out-v.2 \  
  -define:Test_lines-set:-loop-4.64:1-end \  
  -define:Geo:geometry_builder \  
  -V-63-B-63-TDO-3 \  
  -fuse:fname_GEO-63.3_g6 \  
  -girth 6 \  
  -test.Test_lines.FFFF \  
  -end  

g6
SECTION DESIGN THEORY:

design PG_2.3:
$\$(ORBITER_PATH)\orbiter.out-v.8:\$
- create_design-q.3-family-PG_2.q-end

design PG_2.4:
$\$(ORBITER_PATH)\orbiter.out-v.8:\$
- create_design-q.4-family-PG_2.q-end

generators H5="1,2,3,4,0,6,7,8,9,5,10,11,12"
generators N5="0,1,2,3,4,9,5,6,7,8,10,11,12,0,4,3,2,1,5,9,8,7,6,10,11,12,0,2,4,1,3,5,7,9,6,8,10,11,12,0,1,2,3,4,5,6,7,8,9,11,10,12,1,2,3,4,0,6,7,8,9,5,10,11,12,5,9,8,7,6,0,4,3,2,1,10,11,12"

design PG_2.3_table_create:
$\$(ORBITER_PATH)\orbiter.out-v.2:\$
- define-D-design-q.3-family-PG_2.q-end
- define-Sym13-permutation_group-symmetric_group.13-end
- define-T-design_table-D."PG_2.13".Sym13-end

#written_file PG_2.13_design_table.csv

#User_time: 7:30

design PG_2.3_group.5:
$\$(ORBITER_PATH)\orbiter.out-v.2:\$
- define-D-design-q.3-family-PG_2.q-end
- define-T-design_table-D."PG_2.13".Sym13-end
- define-LSW-large_set_with_symmetry_assumption-T-"5".$(GENERATORS_H5)\$
- define-LSW-large_set_with_symmetry_assumption-T-"1200".$(GENERATORS_N5)\$
- prefix."H5"\$
- selected_orbit_length.5"
design PG2_3_group5_sol_0:

$\text{design\_pg2_3\_group5\_sol_0}$

Define linear group PGGL 2 32 with G:

$\text{define\_G\_linear\_group\_PGGL2_32}$

Group theoretic activity:

$\text{group\_theoretic\_activity}$

Problem label KM_PGGL2_32-W-depth 8:

$\text{problem\_label\_KM\_PGGL2_32}$

Kramer Mesner matrix 7 8:

$\text{Kramer\_Mesner\_matrix}$

Draw options:

$\text{draw\_options}$

-orbits-on-subsets 8:

$\text{-orbits\_on\_subsets}$

-depth 5:

$\text{-depth}$

-linear-size 8:

$\text{-linear\_size}$

-invariant-under-PGGL2_32:

$\text{-invariant\_under}$

User-time: 2:39

-orbits 678

-run: 2:39
\texttt{11926 \triangleright \triangleright -end}
\texttt{11927 \triangleright \$(\text{ORBITER\_PATH})\text{orbiter.out--v.2--draw_matrix}\backslash}
\texttt{11928 \triangleright \indent \text{-input\_csv\_file\_KM\_PGGL\_2\_32\_KM\_7\_8.csv}}
\texttt{11929 \triangleright \indent \text{-box\_width.20--bit\_depth.24\backslash}
\texttt{11930 \triangleright \indent \text{-partition.3\_32.97--end}}
\texttt{11931 \indent \text{pdflatex\_KM\_PGGL\_2\_32\_poset\_lvl.8.tex}}
\texttt{11932 \indent \text{open\_KM\_PGGL\_2\_32\_poset\_lvl.8.pdf}}
\texttt{11933 \indent \text{open\_KM\_PGGL\_2\_32\_KM\_7\_8\_draw.bmp}}
\texttt{11934 \indent \text{\$\text{\textbackslash ORBITER\_PATH})\text{orbiter.out--v.4\backslash}}
\texttt{11935 \indent \text{\indent \text{-define-D--diophant\backslash}}
\texttt{11936 \indent \text{\indent \text{-label."KM\_PGGL\_2\_32\_KM\_7\_8\_system"\backslash}}
\texttt{11937 \indent \text{\indent \text{-coefficient\_matrix\_csv-KM\_PGGL\_2\_32\_KM\_7\_8.csv\backslash}}
\texttt{11938 \indent \text{\indent \text{-RHS\_constant."10,10,1"\backslash}}
\texttt{11939 \indent \text{\indent \text{-x\_min\_global.0--x\_max\_global.1\backslash}}
\texttt{11940 \indent \text{\indent \text{-end\backslash}}
\texttt{11941 \indent \text{\indent \text{-with-D--do\backslash}}
\texttt{11942 \indent \text{\indent \text{\indent -diophant\_activity--solve\_mckay\backslash}}
\texttt{11943 \indent \text{\indent \text{\indent -end}}
\texttt{11944 \indent \text{\indent \text{-end}}
\texttt{11945 \indent \text{\indent \text{-end}}
\texttt{11946 \text{\indent \text{\textbackslash}}}
\texttt{11947 \text{\indent \text{\textbackslash}}}
\texttt{11948 \text{\indent \text{KM\_PGSL\_3\_5:\text\textbackslash}}}
\texttt{11949 \indent \text{\indent \text{-define\_G--linear\_group--PSL\_3\_5--end}}
\texttt{11950 \indent \text{\indent \text{-with\_G--do\text\textbackslash}}}
\texttt{11951 \text{\indent \text{\indent -group\_theoretic\_activity\text\textbackslash}}}
\texttt{11952 \text{\indent \text{\indent -poset\_classification\_control\text\textbackslash}}}
\texttt{11953 \text{\indent \text{\indent -problem\_label\_KM\_PSL\_3\_5--W--depth.10\text\textbackslash}}}
\texttt{11954 \text{\indent \text{\indent -Kramer\_Mesner\_matrix.8\_10\text\textbackslash}}}
\texttt{11955 \text{\indent \text{\indent -draw\_poset\text\textbackslash}}}
\texttt{11956 \text{\indent \text{\indent -draw\_options--embedded--sideways\text\textbackslash}}}
\texttt{11957 \text{\indent \text{\indent -radius.50--scale.0.5--line\_width.0.3--end}}
\texttt{11958 \text{\indent \text{\indent -orbits\_on\_subsets.10\text\textbackslash}}}
\texttt{11959 \text{\indent \text{\indent -end}}
\texttt{11960 \text{\indent \text{\indent -end}}
\texttt{11961 \text{\indent \text{\textbackslash}}}
\texttt{11962 \indent \text{-define\_D--diophant\text\textbackslash}}
\texttt{11963 \indent \text{-label."KM\_PSL\_3\_5\_KM\_8\_10\_system"\text\textbackslash}}
\texttt{11964 \indent \text{-coefficient\_matrix\_csv-KM\_PSL\_3\_5\_KM\_8\_10.csv\text\textbackslash}}
\texttt{11965 \indent \text{-RHS\_constant."93,93,1"\text\textbackslash}}
\texttt{11966 \indent \text{-x\_min\_global.0--x\_max\_global.1\text\textbackslash}}
\texttt{11967 \indent \text{-end\text\textbackslash}}
\texttt{11968 \indent \text{-with\_D--do\text\textbackslash}}
\texttt{11969 \indent \text{\indent -diophant\_activity--solve\_mckay\text\textbackslash}}
#the sorted ranks of all subsets are: (9, 116, 140, 205, 278, 308, 375, 389, 455, 482, 577, 640, 656)

#the sorted ranks of all subsets are: (47, 87, 111, 148, 246, 270, 339, 414, 426, 557, 625, 651, 714)

#the sorted ranks of all subsets are: (22, 58, 179, 199, 237, 298, 371, 444, 499, 511, 559, 612, 662)

#the sorted ranks of all subsets are: (37, 90, 102, 184, 253, 297, 303, 393, 447, 582, 620, 664, 683)

#191704, 9, 116, 140, 205, 278, 308, 375, 389, 455, 482, 577, 640, 656

#964499, 47, 87, 111, 148, 246, 270, 339, 414, 426, 557, 625, 651, 714

#445252, 22, 58, 179, 199, 237, 298, 371, 444, 499, 511, 559, 612, 662

#761894, 37, 90, 102, 184, 253, 297, 303, 393, 447, 582, 620, 664, 683

#191704, 445252, 761894, 964499

#191704, 445252, 761894, 964499

#191704, 9, 116, 140, 205, 278, 308, 375, 389, 455, 482, 577, 640, 656

#964499, 47, 87, 111, 148, 246, 270, 339, 414, 426, 557, 625, 651, 714

#445252, 22, 58, 179, 199, 237, 298, 371, 444, 499, 511, 559, 612, 662

#761894, 37, 90, 102, 184, 253, 297, 303, 393, 447, 582, 620, 664, 683

#191704, 445252, 761894, 964499

#design large set rank_k subsets a:

#design large set rank_k subsets b:

#design large set rank_k subsets c:

#design large set rank_k subsets d:

#################################################################################

# design large set rank_k subsets

# $(ORBITER\ PATH)\ orbiter.out -v -2 -rank_k_subsets 13 -4 $(PLANE_1)

# $(ORBITER\ PATH)\ orbiter.out -v -2 -rank_k_subsets 13 -4 $(PLANE_2)

# $(ORBITER\ PATH)\ orbiter.out -v -2 -rank_k_subsets 13 -4 $(PLANE_3)

# $(ORBITER\ PATH)\ orbiter.out -v -2 -rank_k_subsets 13 -4 $(PLANE_4)
AG_2_3.inc:

```
12019 # (ORBITER_PATH)orbit.out -v 2
12020   -define Geo -geometry_builder
12021     -V 9 -B 12
12022       -TDO 4 -fuse 1
12023         -fname GEO AG_2_3
12024           -test 3, 4, 5, 6, 7, 8, 9, FFF
12025               -end
12026
12027 #9-12-3
12029 # -1-1
12030 #432
12031
12032
12033
12034
12035
12036
12037 LS_AG_2_3.design_table_create:
12038 # (ORBITER_PATH)orbit.out -v 20
12039   -define D -design -list_of_blocks
12040     9-3 $(AG_2_3_BLOCKS) -end
12041       -define Sym9 -permutation_group -symmetric_group 9 -end
12042           -define T -design_table D "AG_2_3" Sym9
12043
12044 # creates AG_2_3.design_table.csv
12045   # and AG_2_3.makefile
12046
12047 #0, 0, 13, 22, 27, 35, 41, 47, 53, 55, 59, 71, 76
12048 # is the first design in AG_2_3.design_table.csv
12049 .
12050 # poset_orbit_node::init_root_node storing strong generators for a group of order 362880
12051 # stabilizer order 432
12052 # 840 designs
12053
12054 # User time: 0.13 of a second, dt=13 tps=100
12055
12056
```
AG_2_3_on_designs:
$\$(\textit{ORBITER\ PATH})\texttt{orbiter.out-\(v\cdot2\)}$
$\$ -\texttt{define-gens}\ -\texttt{vector}\ -\texttt{file}\ AG_2_3\_gens.csv\ -\texttt{end} $
$\$ -\texttt{define-G}\ -\texttt{permutation}\ -\texttt{group} $
$\$ -\texttt{bsgs-AG_2_3}\"\ AG_2_3\"\ -840\cdot362880\"0,1,2,3,4,5,6,7\"\ -\texttt{gens}\ -\texttt{end} $
$\$ -\texttt{with-G}\ -\texttt{do} $
$\$ -\texttt{group}\_\texttt{theoretic}\_\texttt{activity} $
$\$ -\texttt{orbits}\_\texttt{on}\_\texttt{points} $
$\$ -\texttt{stabilizer}\_\texttt{of}\_\texttt{orbit}\_\texttt{rep}0\texttt{)}$
$\$ -\texttt{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:
$\$(\textit{ORBITER\ PATH})\texttt{orbiter.out-\(v\cdot2\)}$
$\$ -\texttt{define-gens}\ -\texttt{vector}\ -\texttt{file}\ AG_2_3\_stab\_orb\_0\_gens.csv\ -\texttt{end} $
$\$ -\texttt{define-G}\ -\texttt{permutation}\ -\texttt{group} $
$\$ -\texttt{bsgs-AG_2_3\_stab\_orb\_0}\"AG_2_3\_stab\_orb\_0\"\ -840\cdot432\"0,1,2,3,4,5,6,7,8\"\ -\texttt{gens}\ -\texttt{end} $
$\$ -\texttt{with-G}\ -\texttt{do} $
$\$ -\texttt{group}\_\texttt{theoretic}\_\texttt{activity} $
$\$ -\texttt{export}\_\texttt{orbiter} $
$\$ -\texttt{end} $

AG_2_3\_stab\_orb\_0\_Perm840\_res192:
$\$(\textit{ORBITER\ PATH})\texttt{orbiter.out-\(v\cdot2\)}$
$\$ -\texttt{define-gens}\ -\texttt{vector}\ -\texttt{file}\ Perm840\_res192\_gens.csv\ -\texttt{end} $
$\$ -\texttt{define-G}\ -\texttt{permutation}\ -\texttt{group} $
$\$ -\texttt{bsgs-Perm840\_res192}\"Perm840\cdot\{\texttt{rm}\cdot res192\}\"\ -840\cdot432\"0,1,2,3,4,5,6,7,8\"\ -\texttt{gens}\ -\texttt{end} $
$\$ -\texttt{with-G}\ -\texttt{do} $
$\$ -\texttt{group}\_\texttt{theoretic}\_\texttt{activity} $
$\$ -\texttt{report} $
$\$ -\texttt{end} $

\texttt{pdflatex}\ Perm840\_res192\_report.tex
\texttt{open}\ Perm840\_res192\_report.pdf

643
LS_AG_2_3.disjoint_sets_graph_and_cliques:

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

\begin{verbatim}
define-Gamma-graph-
define-disjoint_sets_graph-
AG_2_3_design_table.csv-
end-
with-Gamma-do-
define-graph_theoretic_activity-
define-save-
end-
with-Gamma-do-
define-graph_theoretic_activity-
don-define-cliques-target_size=7-end-
end-
print_symbols
end

#AG_2_3_design_table_disjoint_sets.colored_graph
User-time: 0.66 of a second, \(dt=66\text{ tps}=100\)

#AG_2_3_design_table_disjoint_sets_sol.txt
#AG_2_3_design_table_disjoint_sets_sol.csv

#15360-solutions

LS_AG_2_3.disjoint_sets_split:

$(ORBITER\_PATH)orbiter.out-\text{-v.4} $

\begin{verbatim}
define-Gamma-graph-load-
AG_2_3_design_table_disjoint_sets.colored_graph-
end-
with-Gamma-do-
define-graph_theoretic_activity-
don-split_by_clique=0.0-end-
end
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-\text{-v.20} $

\begin{verbatim}
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-
\end{verbatim}

644
12150 ▷ ▷ -with-D-do-
12151 ▷ ▷ -design_activity-
12152 ▷ ▷ ▷ -extract_solutions_by_index:"AG_2_3".Sym9-
12153 ▷ ▷ ▷ ▷ AG_2_3_design_table_disjoint_sets_sol.csv-
12154 ▷ ▷ ▷ ▷ solutions.csv-
12155 ▷ ▷ ▷ ▷ ""-
12156 ▷ ▷ ▷ -end
12157
12158 #User-time: 0.39 of a second, dt=39 tps=100
12159 #solutions.csv
12160
12161
12162
12163
12164
12165 #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12166 #SECTION 10.6: Design Theory -- Delandtsheer-Doyen
12167
12168 SECTION DESIGN THEORY DELANDTSHEER DOYEN:
12169
12170
12171 DD_PP4:
12172 ▷ $(ORBITER_PATH)orbiter.out-v.6-
12173 ▷ ▷ -Delandtsheer-Doyen-$\text{PP4}$.$\text{PP4\_GROUP1}$.$\text{PP4\_MASK1}$-
12174 ▷ ▷ ▷ -end-
12175
12176 DD_PP4_system:
12177 ▷ $(ORBITER_PATH)orbiter.out-v.4-
12178 ▷ ▷ -define-D-diophant-label_PP4-
12179 ▷ ▷ -problem_of_Steiner_type_10-PP4_pair_covering.csv-
12180 ▷ ▷ ▷ -has_sum_1-
12181 ▷ ▷ ▷ -end-
12182 ▷ ▷ ▷ -with-D-do-
12183 ▷ ▷ ▷ -diophant_activity-solve_mckay-
12184 ▷ ▷ ▷ -end
12185
12186
12187
12188
12189
12190 DD_CC:
12191 ▷ $(ORBITER_PATH)orbiter.out-v.6-
12192 ▷ ▷ -Delandtsheer_Doyen--search_wrt_subgroup-
12193 ▷ ▷ ▷ $(\text{DELANDTSHEER\_DOYEN\_PROBLEM\_COLOBURN\_COLOBURN\_7\_13})-
12194 ▷ ▷ ▷ $(\text{DELANDTSHEER\_DOYEN\_PROBLEM\_COLOBURN\_COLOBURN\_7\_13\_GROUP1})-
12195 ▷ ▷ ▷ $(\text{DELANDTSHEER\_DOYEN\_PROBLEM\_COLOBURN\_COLOBURN\_7\_13\_MASK1})-
12196  ▶ ▶  -end.
12197
12198  #target.level:6
12199  #k2:15
12200  #number of k-orbits at target level: 1774964
12201
12202  #creates DD_CC_7_13_pair_covering.csv
12203
12204
12205  DD_CC_system:
12206  ▶  $(ORBITER_PATH)orbiter.out -v 4 \n12207  ▶  ◄ defines D -diophant: -label: DD_CC_7_13 \n12208  ▶  ◄ problem of Steiner type 45-DD_CC_7_13_pair_covering.csv \n12209  ▶  ◄ has sum 3 \n12210  ▶  ◄ -end \n12211  ▶  ◄ with D -do \n12212  ▶  ◄ diophant_activity -solve_mckay \n12213  ▶  ◄ -end \n12214  ▶
12215
12216
12217  #no-solution
12218
12219
12220
12221  # 18603 = 27 * 53 * 13
12222
12223  DD_M1_G1:
12224  ▶  $(ORBITER_PATH)orbiter.out -v 4 \n12225  ▶  ◄ Delandtsheer Doyen \n12226  ▶  ◄ ◄ $(DELANDTSHEER_DOYEN_PROBLEM_27_53) \n12227  ▶  ◄ ◄ $(DELANDTSHEER_DOYEN_PROBLEM_27_53_GROUP1) \n12228  ▶  ◄ ◄ $(DELANDTSHEER_DOYEN_PROBLEM_27_53_MASK1) \n12229  ▶  ◄ ◄ -end \n12230
12231  DD_M1_G1_S:
12232  ▶  $(ORBITER_PATH)orbiter.out -v 4 \n12233  ▶  ◄ Delandtsheer Doyen \n12234  ▶  ◄ ◄ $(DELANDTSHEER_DOYEN_PROBLEM_27_53) \n12235  ▶  ◄ ◄ $(DELANDTSHEER_DOYEN_PROBLEM_27_53_GROUP1) \n12236  ▶  ◄ ◄ $(DELANDTSHEER_DOYEN_PROBLEM_27_53_MASK1) \n12237  ▶  ◄ ◄ -singletons \n12238  ▶  ◄ ◄ -end \n12239
12240
12241  DD_PG_2_4_M1_G1:
12242  ▶  $(ORBITER_PATH)orbiter.out -v 4 \n
PG_2.27_special:
$\$(ORBITER\ PATH)\ orbiter.out-v.2\$

$\define-F-\text{finite\ field}\ -q.27\ -\text{override\ polynomial}46\ -\text{end}$.}

$\with-F-\text{do}\$

$\text{-finite\ field\ activity}$

$\text{-cheat\ sheet\ PG}2\ -\text{end}$

$\text{pdflatex\ PG}2.27.tex$

$\text{open\ PG}2.27.pdf$

# Section 10.7::Tactical-Decompositions

SECTION_TACTICAL_DECOMPOSITIONS:

max_arc_16_4_start:

$\$(ORBITER\ PATH)\ orbiter.out-v.4\ -\text{maximal\ arc\ parameters}16-4$

max_arc_16_4_convert_stack.tdo:

$\$(ORBITER\ PATH)\ orbiter.out-v.4\ -\text{convert\ stack\ to\ tdo}\ max\ arc\ q16\ r4\ .stack$

max_arc_16_4_refine:

$\$(ORBITER\ PATH)\ orbiter.out-v.4\ -\text{tdo\ refinement}$

$\text{-input\ file}\ max\ arc\ q16\ r4\ .tdo\ -\text{dual\ is\ linear\ space}\ -\text{end}$

max_arc_16_4r_print:

$\$(ORBITER\ PATH)\ orbiter.out-v.4\ -\text{tdo\ print}\ max\ arc\ q16\ r4\ r.tdo$

# Section 10.8::Spreads
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

-cheat_sheet_desarguesian_spread-2-end

pdflatex Desarguesian_Spread_5_2.tex

open Desarguesian_Spread_5_2.pdf

desarguesian_spread_in_PG_3_4:

$\$(ORBITER\_PATH)orbiter.out\ -v\ 3\$

-define FQ -finite_field -q 16 -end

-define Fq -finite_field -q 4 -end

-with FQ -and Fq -do -finite_field_activity

-cheat_sheet_desarguesian_spread-2-end

pdflatex Desarguesian_Spread_3_4.tex

open Desarguesian_Spread_3_4.pdf

desarguesian_spread_in_PG_3_5:

$\$(ORBITER\_PATH)orbiter.out\ -v\ 3\$

-define FQ -finite_field -q 25 -end

-define Fq -finite_field -q 5 -end

-with FQ -and Fq -do -finite_field_activity

-cheat_sheet_desarguesian_spread-2-end

pdflatex Desarguesian_Spread_3_5.tex

open Desarguesian_Spread_3_5.pdf

spreads4:

-mkdir SPREADS_4

-rm live_points.txt

$\$(ORBITER\_PATH)orbiter.out\ -v\ 10\$

-define F -finite_field -q 2 -end

-define P -projective_space -3 F -end

-with P -do
12335 $\triangleright$ $\triangleright$ -projective_space_activity\\
12336 $\triangleright$ $\triangleright$ -spread_classify:2-problem_label-spreads_2,2-depth:5-draw_poset\\
12337 $\triangleright$ $\triangleright$ -end
12338
12339
12340 spreads16.4:
12341 $\triangleright$ $(\text{ORBITER\_PATH})$orbiter.out-v.6\\
12342 $\triangleright$ $\triangleright$ -orbiter_path:$(\text{ORBITER\_PATH})\\
12343 $\triangleright$ $\triangleright$ -define-F-finite_field-q.4-end\\
12344 $\triangleright$ $\triangleright$ -define-P-projective_space:3-F-end\\
12345 $\triangleright$ $\triangleright$ -with-P-do\\
12346 $\triangleright$ $\triangleright$ -projective_space_activity\\
12347 $\triangleright$ $\triangleright$ -spread_classify:2-problem_label-spreads_4,2\\
12348 $\triangleright$ $\triangleright$ $\triangleright$ -W-depth:17-draw_poset\\
12349 $\triangleright$ $\triangleright$ $\triangleright$ -draw_options:-radius:20\\
12350 $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ -nodes_empty:-line_width:0.2:-embedded\\
12351 $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ -end\\
12352 $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ -report\\
12353 $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ $\triangleright$ -end
12354 $\triangleright$ #pdflatex-spreads_4,2_pose_detailed_lvl_17.tex
12355 $\triangleright$ #open-spreads_4,2_pose_detailed_lvl_17.pdf
12356
12357 #17
12359 aaaaaaaaaaaabaaaaaaaaaaaaaaaaaaaaaaaaaaaaaagaaaaaaaaadmaaaaaaaaaaafaaaaaab
12360 aaaaaaaaaaabaadaadaadaaaaaaaeaaaaagaaaaaaaaffaaaaaaaaaaaaaaaaab
12361 aaaaaaadaaaaaaaaaaaaaaaaaaaabaaaaaaaaaaadmaaaaaaaaaacaijlpahaabahealabecljdoaabdmpbnaeaa
12362 aamanajmpnaa
12364 aaaaaaaaaaaabaaaaaaaaaaaaaaaaaaaaaaaaaaaagaaaaaaaadmaaaaaaaaaaaadaaaaaaaadaaaaabaa
12365 aaaaaaaaabaabaabfakamaababaeldibaacijaipbocaappbnnjebraajccgpciaabbeagcllaa
12366 #1:3-1126-in.0-of.a.second,d=0.tps:=.100
12367 #(81600,-1200,-576)-average.is=27792:+0.-3
12368
12369
12370
12371
12372 #Section 10.9: Translation planes

649
SECTION_TRANSLATION_PLANES:

TP_9_0:
-define F: finite_field: q:3: end
-define PGL4: linear_group: PGL.4: F: end
-define PGL5: linear_group: PGL.5: F: end
-with PGL4: and PGL5: do
-group_theoretic_activity:
-Andre_Bruck_Bose_construction: 0: "TP9-0:"
-end

TP_9_1:
-define F: finite_field: q:3: end
-define PGL4: linear_group: PGL.4: F: end
-define PGL5: linear_group: PGL.5: F: end
-with PGL4: and PGL5: do
-group_theoretic_activity:
-Andre_Bruck_Bose_construction: 0: "TP9-1:"
-end

TP_16_4:
-define F: finite_field: q:4: end
-define PGGL4: linear_group: PGGL.4: F: end
-define PGGL5: linear_group: PGGL.5: F: end
-with PGGL4: and PGGL5: do
-group_theoretic_activity:
-Andre_Bruck_Bose_construction: 0: "TP16-4-HALL:"
-end
# Section 10.10: Packings

## SECTION PACKINGS:

```
12456 # Section 10.10: Packings
12458
12459
12460 SECTION PACKINGS:
12461
12462 spread_table_PG_3_4:
12463 `mkdir SPREAD_TABLES_4
12464 `$(ORBITER_PATH)orbiter.out--v.6`
12465 `define F--finite_field--q.4--end`
12466 `define P--projective_space--3--F--end`
```
spreads

12471 #1020 self-dual spreads
12472 #User-time: 56:38 on Mac
12473
12474 spread_table_PG_3_5_regular:
12475 » mkdir -v 3
12476 » $(ORBITER_PATH) orbiter.out -v 6
12477 » -define F -finite_field -q 5 -end
12478 » -define P -projective_space 3 F -end
12479 » -define T -spread_table P 2 "12" "SPREAD_TABLES_5_REG/
12480 » -print_symbols
12481
12482
12483 #12/9/2020: 34 sec on Mac
12484 #12 is the index of the regular spread in the classification of spreads
12485 #155000 spreads
12486
12487
12488 PG_3_5_desarguesian_spread:
12489 » $(ORBITER_PATH) orbiter.out -v 3
12490 » -define FQ -finite_field -q 25 -end
12491 » -define Fq -finite_field -q 5 -end
12492 » -with FQ -and Fq -do
12493 » -finite_field_activity
12494 » -cheat_sheet_desarguesian_spread 2
12495 » -end
12496 » pdflatex Desarguesian_Spread_3_5.tex
12497 » open Desarguesian_Spread_3_5.pdf
12498
12499 #Spread elements by rank: (0, 805, 36, 108, 72, 144, 581, 509, 686, 415, 639, 758,
285, 722, 332, 343, 202, 592, 473, 238, 675, 379, 166, 545, 249, 451)
12500
12501
12502 PG_3_5_element_of_order_31:
12503 » $(ORBITER_PATH) orbiter.out -v 6 -define G
12504 » -linear_group -GL 3 5 -end
12505 » -with G -do
12506 » -group_theoretic_activity
12507 » -raise_to_the_power 0 1 0 0 1 3 0 4 -31
12508 » -end
12509 » pdflatex GL_3_5_power.tex
12510 » open GL_3_5_power.pdf
12511
12512 PG_3_5_element_of_order_31_normalizer:
$(\text{ORBITER\ PATH})\text{orbiter.out\ -v\ 6\ -define\ G\ -}
\text{-linear_group\ -PGL\ 4\ 5\ -end}\$
\text{-with\ G\ -do\ -group\_theoretic\_activity\ -}
\text{-normalizer\_of\_cyclic\_subgroup\ "31"\ -}
\text{"2,0,0,0,0,1,0,0,0,1,0,3,0,4"\ -}
\text{-end\ -mv\ normalizer\ of\ 31\ in\ PGL\ 4\ 5\ .tex\ -normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .tex\ -}
\text{pdflatex\ normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .tex\ -}
\text{open\-normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .pdf\ -}
\text{PG\ 3\ 5\ element\ of\ order\ 31\ GL\ normalizer:\ -}
\text{$(\text{ORBITER\ PATH})\text{orbiter.out\ -v\ 6\ -define\ G\ -}
\text{-linear_group\ -GL\ 4\ 5\ -end}\$
\text{-with\ G\ -do\ -group\_theoretic\_activity\ -}
\text{-normalizer\_of\_cyclic\_subgroup\ "124"\ -}
\text{"2,0,0,0,0,1,0,0,0,1,0,3,0,4"\ -}
\text{-end\ -mv\ normalizer\ of\ 31\ in\ GL\ 4\ 5\ .tex\ -normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .tex\ -}
\text{pdflatex\ normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .tex\ -}
\text{open\-normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .pdf\ -}
\text{PG\ 3\ 5\ element\ of\ order\ 31\ ME\ normalizer:\ -}
\text{$(\text{ORBITER\ PATH})\text{orbiter.out\ -v\ 5\ -}
\text{-define\ G\ -linear_group\ -PGL\ 4\ 5\ -end\ -}
\text{-with\ G\ -do\ -group\_theoretic\_activity\ -}
\text{-normalizer\_of\_cyclic\_subgroup\ "31"\ -}
\text{"1,0,0,0,0,3,4,3,0,3,3,4,0,3,2,3"\ -}
\text{-end\ -mv\ normalizer\ of\ 31\ in\ PGL\ 4\ 5\ .tex\ -normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .tex\ -}
\text{pdflatex\ normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .tex\ -}
\text{open\-normalizer\ of\ 31\ AB\ in\ PGL\ 4\ 5\ .pdf\ -}
\text{PG\ 3\ 5\ assume\ 31\ graph:\ -}
\text{$(\text{ORBITER\ PATH})\text{orbiter.out\ -v\ 5\ -}}$
12560  \$ \text{-define-F-} \text{finite}\_\text{field}\-q\-5\-\text{-end}\$
12561  \$ \text{-define-P-} \text{projective}\_\text{space}\-3\-\text{F}\-\text{-end}\$
12562  \$ \text{-define-T-} \text{spread}\_\text{table}\-P\-2\-\text{12}\-\text{SPREAD}\_\text{TABLES}\-5\-\text{REG}/\$
12563  \$ \text{-define-PW-} \text{packing}\_\text{with}\_\text{symmetry}\_\text{assumption}\-T\$
12564  \$ \text{-define-} \text{H}\-\text{"H31\"} \text{\$(PGL}\_\text{4}\_\text{5}\_\text{SUBGROUP}\_\text{31}\_\text{ME}\-\text{-end}\$
12565  \$ \text{-define-N-} \text{"N31\"} \text{\$(PGL}\_\text{4}\_\text{5}\_\text{SUBGROUP}\_\text{31}\_\text{ME}\_\text{NORMALIZER}\-\text{-end}\$
12566  \$ \text{-end}\$
12567  \$ \text{-define-PWF-} \text{packing}\_\text{choose}\_\text{fixed}\_\text{points}\-P\-0\-\text{-end}\$
12568  \$ \text{-define-L-} \text{packing}\_\text{long}\_\text{orbits}\-PWF\$
12569  \$ \text{-define-} \text{orbit}\_\text{length}\-31\-\text{-} \text{clique}\_\text{size}\-1\-\text{-create}\_\text{graphs}\-\text{-end}\$
12570  \$ \text{-print}\_\text{symbols}\$
12571  \text{pdflatex}\-H31\_\text{reduced}\_\text{spread}\_\text{orbits}\_\text{orbits}\_\text{report}\_\text{tex}$
12572  \text{open-H31\_\text{reduced}\_\text{spread}\_\text{orbits}\_\text{orbits}\_\text{report}\_\text{pdf}$
12573  \text{pdflatex}\-H31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}\_\text{tex}$
12574  \text{open-H31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf}$
12575  \text{pdflatex}\-H31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf$
12576  \text{open-H31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf}$
12577  \text{pdflatex}\-N31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}.tex$
12578  \text{open-N31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf}$
12579  \text{pdflatex}\-H31\_\text{point}\_\text{orbits}\_\text{orbits}\_\text{report}.tex$
12580  \text{open-H31\_\text{point}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf}$
12581  \text{pdflatex}\-N31\_\text{point}\_\text{orbits}\_\text{orbits}\_\text{report}.tex$
12582  \text{open-N31\_\text{point}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf}$
12583  \text{#pdflatex}\-H31\_\text{spread}\_\text{orbits}\_\text{orbits}\_\text{report}.tex$
12584  \text{open-H31\_\text{spread}\_\text{orbits}\_\text{orbits}\_\text{report}.pdf}$
12586  \text{H31\_\text{line}\_\text{orbits}\_\text{orbits}.bin}$
12587  \text{H31\_\text{line}\_\text{orbits}\_\text{orbits}\_\text{report}.tex}$
12588  \text{H\_\text{spread}\_\text{orbits}\_\text{orbits}\_\text{types}\_\text{report}.tex}$
12589  \text{H31\_\text{spread}\_\text{orbits}\_\text{orbits}.bin}$
12590  \text{H31\_\text{good}\_\text{orbits}}$
12591  \text{H31\_\text{spread}\_\text{types}\_\text{reduced}\_\text{orbits}\_\text{types}\_\text{report}.tex}$
12592  \text{H31\_\text{reduced}\_\text{spread}\_\text{orbits}\_\text{orbits}.bin}$
12593  \text{H31\_\text{fpc0}\_\text{lo}\_\text{graph}}$
12594$
12595$
12596  \text{PG}\_\text{3}\_\text{5}\_\text{assume}\_\text{31}\_\text{fpc0}\_\text{lo}\_\text{cliques}}$
12597  \$ \text{\$(ORBITER}\_\text{PATH})$orbiter.out\-v\-2\$
12598  \$ \text{-define-G-} \text{graph}\-load\-H31\_\text{fpc0}\_\text{lo}\_\text{graph}\-\text{-end}\$
12599  \$ \text{-with-G-} \text{do}\$
12600  \$ \text{-graph}\_\text{theoretic}\_\text{activity}\$
12601  \$ \text{-find}\_\text{cliques}\-\text{-target}\_\text{size}\-1\-\text{-end}\$
12602  \$ \text{-print}\_\text{symbols}$
12603$
12604  \text{H31\_\text{fpc0}\_\text{lo}\_\text{sol}.txt}$
12605  \text{H31\_\text{fpc0}\_\text{lo}\_\text{sol}.csv}$
# ToDo: problem when computing the orbits of the normalizer:

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:
```

```
> -define H "H31" $(PGL_4 5_SUBGROUP_31_ME) -end:
```

```
> -define N "H31" $(PGL_4 5_SUBGROUP_31_ME) -end:
```

```
> -define PWF -packing -choose_fixed_points PW 0 -end:
```

```
> -define L -packing_long_orbits PWF:
```

```
> -orbit_length 31 -clique_size 1:
```

```
> -read_solutions:
```

```
> -end:
```

```
> # writes H31_packings.csv
```

```
> # the two packings (ignore the first number)
```

```
> #0, 444, 43313, 154402, 46682, 108254, 75363, 27729, 32139, 5244, 60442, 142811, 111115, 94209,
```

```
> 120678, 89533, 13798, 103994, 129953, 82168, 136838, 19253, 23017, 145985, 134996, 54705, 36
```

```
> 267, 55066, 117542, 96699, 69154, 72460
```

```
> #1, 616, 42728, 152655, 48576, 105431, 79607, 28634, 32817, 9799, 62356, 141176, 110085, 92557,
```

```
> 122136, 86312, 13975, 101942, 126869, 81478, 139352, 18028, 24325, 147284, 130370, 52074, 36
```

```
> 843, 55602, 118454, 95973, 69642, 74036
```

PG_3.5_packing_0_dualize:

```
> $(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/":
```

```
> -with T -do -
```

```
> -spread_table_activity:
```

```
> -dualize Packing $(PENTILLA_WILLIAMS_PRINCE_REG_PACKING_0):
```

```
> -end:
```

PG_3.5_assume_3:

655
12650  ➾  -rm N3B_ME_fixp_cliques.csv
12651  ➾  $(ORBITER_PATH)orbiter.out -v.20 \
12652  ➾  -define F=finite_field -q.5 -end \ 
12653  ➾  -define P=projective_space -3 F- -end \ 
12654  ➾  -define T=spread_table -P.2:"12"."SPREAD_TABLES.5_REG/". \ 
12655  ➾  -define PW=packing_with_symmetry_assumption T \ 
12656  ➾  -define -H."H3B_ME". $(PGL.4_5_SUBGROUP_3B_ME) -end \ 
12657  ➾  -define -N."N3B_ME". $(PGL.4_5_SUBGROUP_3B_ME_NORMALIZER) -end \ 
12658  ➾  -end \ 
12659  ➾  -define PWF=packing_choose_fixed_points PW \ 
12660  ➾  -W=problem label N3B_ME_fixp_cliques \ 
12661  ➾  -preferred_choice 0:0:2 -end \ 
12662  ➾  -print_symbols \ 
12663 \ 
12664  PG.3_5_3B_create_graph_on_long_orbits: \ 
12665  ➾  $(ORBITER_PATH)orbiter.out -v.5 \ 
12666  ➾  -define F=finite_field -q.5 -end \ 
12667  ➾  -define P=projective_space -3 F- -end \ 
12668  ➾  -define T=spread_table -P.2:"12"."SPREAD_TABLES.5_REG/". \ 
12669  ➾  -define PW=packing_with_symmetry_assumption T \ 
12670  ➾  -define -H."H3B_ME". $(PGL.4_5_SUBGROUP_3B_ME) -end \ 
12671  ➾  -define -N."N3B_ME". $(PGL.4_5_SUBGROUP_3B_ME_NORMALIZER) -end \ 
12672  ➾  -end \ 
12673  ➾  -define PWF=packing_choose_fixed_points PW \ 
12674  ➾  -W=problem label N3B_ME_fixp_cliques \ 
12675  ➾  -define L=packing_long_orbits PW \ 
12676  ➾  -end \ 
12677  ➾  -clique_size 10 \ 
12678  ➾  -list_of_cases_from_file N3B_ME_fixp_cliques.csv \ 
12679  ➾  -create_graphs \ 
12680  ➾  -end \ 
12681  ➾  -print_symbols \ 
12682 \ 
12683  # 16120 vertices \ 
12684  # creates H3B_ME_fpc0_lo.graph \ 
12685 \ 
12686  PG.3_5_assume_3B_fpc0_lo_cliques: \ 
12687  ➾  $(ORBITER_PATH)orbiter.out -v.2 \ 
12688  ➾  -define G=graph -load H3B_ME_fpc0_lo.graph -end \ 
12689  ➾  -with G=do \ 
12690  ➾  -graph_theoretic_activity -find cliques \ 
12691  ➾  -target_size 10 -end -end \ 
12692  ➾  -print_symbols \ 
12693 \ 
12694  # 768 solutions \ 
12695  # User time: 8:16 \ 
12696  

656
12697  PG_3.5_assume_3B_long_read:
12698  $(ORBITER_PATH)orbit.out-v.5:\
12699  \> \> -define-F-finite_field-q.5:-end\n12700  \> \> -define-P-projective_space-3-F:-end\n12701  \> \> -define-T-spread_table-P.2:"12":"SPREAD_TABLES.5_REG/"\n12702  \> \> -define-PW-packing_with_symmetry_assumption-T:\n12703  \> \> \> -H."H3B_ME"$(PGL_4.5_SUBGROUP_3B_ME):-end\n12704  \> \> \> -N."N3B_ME"$(PGL_4.5_SUBGROUP_3B_ME_NORMALIZER):-end\n12705  \> \> -end\n12706  \> \> -define-PWF-packing_choose_fixed_points-PW:\n12707  \> \> \> 1-W-problem_label-N3B_ME_fixp_cliques:-end\n12708  \> \> -define-L-packing_long_orbits-PWF:\n12709  \> \> \> -orbit_length-3-clique_size-10\n12710  \> \> \> -list_of_cases_from_file-N3B_ME_fixp_cliques.csv:\n12711  \> \> \> -read_solutions\n12712  \> \> -end\n12713  \> -print_symbols
12714
12715
12716  #total_number_of_packings=768
12717  #written_file-N3B_ME_fixp_cliques_count.csv.of.size-38
12718  #packing_long_orbits::list_of_cases_from_file_before_save_packings_by_case
12719  #packing_long_orbits::save_packings_by_case
12720  #written_file-H3B_ME_packings.csv.of.size-150540
12721
12722
12723
12724
12725  PG_3.5_packing0_print:
12726  $(ORBITER_PATH)orbit.out-v.5:\
12727  \> \> -define-F-finite_field-q.5:-end\n12728  \> \> -define-P-projective_space-3-F:-end\n12729  \> \> -define-T-spread_table-P.2:"12":"SPREAD_TABLES.5_REG/"\n12730  \> \> -define-PW-packing_with_symmetry_assumption-T:\n12731  \> \> \> -H."H31_ME"$(PGL_4.5_SUBGROUP_31_ME):-end\n12732  \> \> \> -N."N31_ME"$(PGL_4.5_SUBGROUP_31_ME_NORMALIZER):-end\n12733  \> \> -end\n12734  \> \> -define-PWF-packing_choose_fixed_points-PW:\n12735  \> \> \> 0-W-problem_label-N3B_ME_fixp_cliques:-end\n12736  \> \> \> -with-PWF-do-packing_fixed_points_activity:\n12737  \> \> \> -print_packing$(PG_3.5_PACKING_0_WITH_AGO3_FIXP444)\n12738  \> \> -end
12739
12740
12741
12742
12743  #H-in-the-action-on-point-has-6-orbits-of-length-1-and-50-orbits-of-length-3
12744  #the fixed points make up a line. It is the first of the two special lines.
12745  #line orbits 2 and 3 make up the second of the special line.
12746
12747  #H in the action on lines has 270 orbits.
12748  #There are 2 orbits of length 1 and 268 orbits of length 3.
12749  #the two orbits of length one are the special lines.
12750  #the first line is fixed pointwise.
12751
12752
12753  #spread orbits of length 1:
12754  #Orbit 0:
12755  #$$
12756  #0=(\ldots,36,\ldots,72,108,144,157,193,229,265,301,314,350,386,422,458,466,\ldots,
12757  502,538,574,610,623,659,695,731,767,805,\ldots)
12758
12759
12760  # H3-orbit_of_length_1:
12761
12762
12763
12764  PG_3_5_H3_orbit_of_length_1:
12765  $(\text{orбит}_{\text{orbits}}.\text{out})-v.5:\$
12766  >>\>$\text{-define F-finite field-}q.5\text{-end}\$
12767  >>\>$\text{-define P-projective space-}3\text{-F-end}\$
12768  >>\>$\text{-define T-spread table-P-2}\cdot12\cdot\text{SPREAD\_TABLES.5\_REG/}\cdot$
12769  >>\>$\text{-with T-do}\$
12770  >>\>$\text{-spread table activity}\$
12771  >>\>$\text{-find spread and dualize-}\$(\text{ORBIT\_OF\_LENGTH\_1})\cdot$
12772  >>\>$\text{-end}\$
12773
12774  # The given spread has index 0 in the spread table.
12775  # The dual spread has index 1 in the spread table.
12776
12777
12778  H31\_ORBIT\_OF\_SPREAD\_0=\"0,44137,153432,45323,109781,77407,29412,32522,6582,\ldots\ldots,146811,132608,53487,36011,55803,116998,99446,69752,7329,2\"
12785
12786 PG_3.5.packings_compare:
12787  ▷ $(ORBITER_PATH)orbiter.out-v.5:\n12788  ▷ ▷ _define-F-_finite_field-_q.5-_end:\n12789  ▷ ▷ _define-P-_projective_space-3-F-_end:\n12790  ▷ ▷ _define-T-_spread_table-P-2.12."SPREAD_TABLES.5_REG/".\n12791  ▷ ▷ _define-PW-_packing_with_symmetry_assumption-T:\n12792  ▷ ▷ ▷ ▷ _define-N."H31_ME"$_(PGL_4.5_SUBGROUP_31_ME)._end:\n12793  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12794  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12795  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12796  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12797  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12798  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12799  ▷ ▷ ▷ ▷ _define-PW-_packing_choose_fixed_points-PW:\n12800
12801 PG_3.5.packings_sort_each_row:
12802  ▷ $(ORBITER_PATH)orbiter.out-v.5-_csv_file_sort_each_row-H31_packings.csv
12803  ▷ $(ORBITER_PATH)orbiter.out-v.5-_csv_file_sort_each_row-H3B_ME_packings.csv
12804
12805 # H31_packings_sorted.csv
12806
12807
12808
12809
12810 PG_3.5_assume_31_classify:
12811  ▷ $(ORBITER_PATH)orbiter.out-v.2:\n12812  ▷ ▷ _define-C-_combinatorial_objects:\n12813  ▷ ▷ ▷ _file_of_packings_through_spread_table:\n12814  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12815  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12816  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12817  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12818  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12819  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12820  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12821  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12822  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12823  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12824  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12825  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12826  ▷ ▷ ▷ ▷ _H31_packings.csv:\n12827  ▷ ▷ ▷ ▷ _H31_packings_classification.tex
12828  ▷ ▷ ▷ ▷ _H31_packings_classification.pdf
12829
12830 # ToDo:
12831
659
12832 PG_3_5_assume_3B_classify:
12833 ↳ $(ORBITER\_PATH)oribter.out-v.2:\n12834 ↳ ‐define\-F\-finite\_field\-q\-5\‐end:\n12835 ↳ ‐define\-F\‐projective\_space\-3\-F\‐end:\n12836 ↳ ‐with\-F\‐do\:\n12837 ↳ ‐projective\_space\‐activity:\n12838 ↳ › ‐canonical\_form\_PG:\n12839 ↳ › › › ‐input\-file\_of\_packings\_through\_spread\_table:\n12840 ↳ › › › ‐define\F\f\finite\field\‐q\-5\‐end:\n12841 ↳ › › › ‐define\O\o\orthogonal\space\0\-5\-F\‐end:\n12842 ↳ › › › ‐with\-0\‐do\‐orthogonal\_space\_activity\:\n12843 ↳ › › › ‐create\_BLT\_set\‐catalogue\-1\‐end:\n12844 ↳ › › › ‐end
12845 ↳ ‐save\_classification\_cl\_class:\n12846 ↳ › › › ‐save\_canonical\_labeling:\n12847 ↳ › › › ‐save\_ago:\n12848 ↳ › › › ‐save\_cumulative\_canonical\_labeling\_cl\_cl:\n12849 ↳ › › › ‐save\_cumulative\_ago\_cl\_ago:\n12850 ↳ › › › ‐save\_cumulative\_data\_cl\_data:\n12851 ↳ › › › ‐save\_fibration\_cl\_fib:\n12852 ↳ › › › ‐report\_H3B\_packings\:\n12853 ↳ › › › ‐end
12854 ↳ ‐open\_H3B\_packings\_classification\_pdf
12855
12856
12857
12858
12859
12860 ###########################################################################
12861 #Section 10.11: BLT-sets
12862
12863 SECTION_BLT_SETS:
12864
12865 BLT_5_1:
12866 ↳ $(ORBITER\_PATH)oribter.out-v.2:\n12867 ↳ ‐define\-F\-finite\_field\-q\-5\‐end:\n12868 ↳ ‐define\-0\‐orthogonal\_space\-0\-5\-F\‐end:\n12869 ↳ ‐with\-0\‐do\‐orthogonal\_space\_activity\:\n12870 ↳ ‐create\_BLT\_set\‐catalogue\-1\‐end:\n12871 ↳ › ‐end
12872 ↳ ‐open\_catalogue\_q5\_isol1\_pdf
12873
12874
12875 BLT_5_Linear:
12876 ↳ $(ORBITER\_PATH)oribter.out-v.2:\n12877 ↳ ‐define\-F\-finite\_field\-q\-5\‐end:\n
660
12878  \[\text{-define 0-orthogonal_space 0.5 F-end}\]
12879  \[\text{-with 0-do-orthogonal_space_activity}\]
12880  \[\text{-create.BLT_set-family:Linear-end}\]
12881  \[\text{-end}\]
12882  pdflatex-\text{BLT Linear q5.tex}
12883  open-\text{BLT Linear q5.pdf}
12884
12885 BLT_{9, K1}:
12886  \$ (\text{ORBITER_PATH})orbiter.out -v 2 \$
12887  \[\text{-define F-finite_field q.9 -end}\]
12888  \[\text{-define 0-orthogonal_space 0.5 F-end}\]
12889  \[\text{-with 0-do-orthogonal_space_activity}\]
12890  \[\text{-create.BLT_set-family:K1-end}\]
12891  \[\text{-end}\]
12892  pdflatex-\text{BLT K1 q9.tex}
12893  open-\text{BLT K1 q9.pdf}
12894
12895
12896
12897
12898 BLT_{11, 0}:
12899  \$ (\text{ORBITER_PATH})orbiter.out -v 2 \$
12900  \[\text{-define F-finite_field q.11 -end}\]
12901  \[\text{-define 0-orthogonal_space 0.5 F-end}\]
12902  \[\text{-with 0-do-orthogonal_space_activity}\]
12903  \[\text{-create.BLT_set-catalogue 0 -end}\]
12904  \[\text{-end}\]
12905  \#pdflatex-0_1_6_2_report.tex
12906  \#open-0_1_6_2_report.pdf
12907
12908
12909 BLT_{11, Fisher}:
12910  \$ (\text{ORBITER_PATH})orbiter.out -v 2 \$
12911  \[\text{-define F-finite_field q.11 -end}\]
12912  \[\text{-define 0-orthogonal_space 0.5 F-end}\]
12913  \[\text{-with 0-do-orthogonal_space_activity}\]
12914  \[\text{-create.BLT_set-family:Fisher-end}\]
12915  \[\text{-end}\]
12916  pdflatex-\text{BLT Fisher q11.tex}
12917  open-\text{BLT Fisher q11.pdf}
12918
12919 BLT_{11, Mondello}:
12920  \$ (\text{ORBITER_PATH})orbiter.out -v 2 \$
12921  \[\text{-define F-finite_field q.11 -end}\]
12922  \[\text{-define 0-orthogonal_space 0.5 F-end}\]
12923  \[\text{-with 0-do-orthogonal_space_activity}\]
12924  \[\text{-create.BLT_set-family:Mondello-end}\]
BLT_13_FTWKB:
\$(ORBITER\_PATH)\texttt{orbiter.out-v.2}:
\begin{verbatim}
--define F=finite_field-q.11-end
--define 0-orthogonal_space-0.5:F=-end
--with 0-do-orthogonal_space_activity
--create_BLT_set-family:"FTWKB"-end
\end{verbatim}
\begin{verbatim}
--end
\end{verbatim}
\begin{verbatim}
pdflatex \texttt{BLT\_FTWKB_q11.tex}
\end{verbatim}
\begin{verbatim}
open-BLT\_FTWKB.q11.pdf
\end{verbatim}

# for K2, q must be congruent to 2 or 3 mod 5
BLT_13_K2:
\$(ORBITER\_PATH)\texttt{orbiter.out-v.2}:
\begin{verbatim}
--define F=finite_field-q.13-end
--define 0-orthogonal_space-0.5:F=-end
--with 0-do-orthogonal_space_activity
--create_BLT_set-family:"Kantor2"-end
\end{verbatim}
\begin{verbatim}
--end
\end{verbatim}
\begin{verbatim}
pdflatex \texttt{BLT\_K2_q13.tex}
\end{verbatim}
\begin{verbatim}
open-BLT\_K2.q13.pdf
\end{verbatim}

BLT_13_starter_5:
\$(ORBITER\_PATH)\texttt{orbiter.out-v.2}:
\begin{verbatim}
--define F=finite_field-q.13-end
--define 0-orthogonal_space-0.5:F=-end
--with 0-do-orthogonal_space_activity
--create_BLT_set-starter-5-problem_label:BLT\_q13-W-depth:5-end
\end{verbatim}
\begin{verbatim}
--end
\end{verbatim}
\begin{verbatim}
pdflatex \texttt{BLT\_q13.tex}
\end{verbatim}
\begin{verbatim}
open-BLT\_q13.pdf
\end{verbatim}

# BLT_13_lvl_5
BLT_13_deep_14:
\$(ORBITER\_PATH)\texttt{orbiter.out-v.2}:
\begin{verbatim}
--define F=finite_field-q.13-end
--define 0-orthogonal_space-0.5:F=-end
--with 0-do-orthogonal_space_activity
--create_BLT_set-starter-14-problem_label:BLT\_q13-W-depth:14-end
\end{verbatim}
\begin{verbatim}
--end
\end{verbatim}
BLT_13_graphs:

```bash
$ (ORBITER_PATH) orbiter.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_graphs 5 0 1 \ 
- end
```

BLT_13_cliques:

```bash
$ (ORBITER_PATH) orbiter.out -v 2 \
- loop L 0 38 1 \
define G graph - load BLT_q13_graph_5_\%L.bin - end \
with G do \
- graph_theoretic_activity - find cliques - - rainbow - target_size 9 - end - end \ 
- end_loop
```

SECTION_CREATING_GRAPHS:

```bash
Cycle_13:

$ (ORBITER_PATH) orbiter.out -v 2 \
define Gamma graph \
cycle 13 \ 
- end
```

triangle_graph:

```bash
$ (ORBITER_PATH) orbiter.out -v 6 \
```
Chain_232:

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

```bash
 DEFINE P1 -VECTOR -DENSE 2,3,2\n```

```bash
 DEFINE P2 -VECTOR -DENSE 2,3,2\n```

```bash
 DEFINE Gamma -GRAPH\n```

```bash
 define -chain_graph_P1-P2\n```

```bash
 -end\n```

```bash
 with Gamma -do\n```

```bash
 graph_theoretic_activity -export_csv\n```

```bash
 -end\n```

```bash
 with Gamma -do\n```

```bash
 graph_theoretic_activity -properties\n```

```bash
 -end\n```

```bash
 Cycle_13.draw:
```

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

```bash
 DEFINE Gamma -GRAPH -CYCLE 13 -end\n```

```bash
 with Gamma -do\n```

```bash
 graph_theoretic_activity -export_csv -end\n```

```bash
 -end\n```

```bash
 with Gamma -do\n```

```bash
 graph_theoretic_activity -export_graphviz -end\n```

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

```bash
 input_csv_file: Cycle_13.csv\n```

```bash
 box_width 20 -bit_depth 8 -partition 4 13 13 -end\n```

```bash
 #dot -Tpng Cycle_13.gv > Cycle_13.png\n```

```bash
 #twopi -Tpng Cycle_13.gv > Cycle_13.png\n```

```bash
 #open Cycle_13_draw.bmp\n```

```bash
 #pdflatex Cycle_13_report.tex\n```

```
```
Cycle_9_eigenvalues:

```
$\text{(ORBITER PATH)orbiter.out -v.2 -define Gamma -graph -cycle 9 -end -with Gamma -do -graph_theoretic_activity -eigenvalues -end}
```

```
pdflatex Cycle_9_eigenvalues.tex
```

```
open Cycle_9_eigenvalues.pdf
```

Paley_13_graph:

```
$\text{(ORBITER PATH)orbiter.out -v.2 -define Gamma -graph Paley 13 -end -with Gamma -do -graph_theoretic_activity -export csv -end -with Gamma -do -graph_theoretic_activity -export graphviz -end}$
```

```
pdflatex Paley_13_eigenvalues.tex
```

```
open Paley_13_eigenvalues.pdf
```

Paley_13_eigenvalues:

```
$\text{(ORBITER PATH)orbiter.out -v.2 -define Gamma -graph Paley 13 -end -with Gamma -do -graph_theoretic_activity -eigenvalues -end}
```

```
pdflatex Paley_13_eigenvalues.tex
```

```
open Paley_13_eigenvalues.pdf
```

trihedral_pair_graph:

```
$\text{(ORBITER PATH)orbiter.out -v.2 -define Gamma -graph trihedral_pair_disjointness_graph -end -with Gamma -do -graph_theoretic_activity -export csv -end}$
```

```
pdflatex trihedral_pair_disjointness.csv
```

```
box width 20 -box depth 8 -partition 4 13 13 -end
```

```
dot -Tpng trihedral_pair_disjointness.gv > trihedral_pair_disjointness.png
```

```
opentrihedral_pair_disjointness.bmp
```

665
open-triheital_pair_disjointness_draw.bmp

small_graph:

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

$\text{-define G-graph-edges_as_pairs 5:}^{0,1,0,2,0,3,0,4,1,3,1,4,2,4}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-exports_csv-}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-exports_graphviz-}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-saves-}:-\text{end}\\$

$\{(\text{ORBITER\ PATH})\text{orbiter.out-v.2-draw_matrix}\\$

$\text{-box_width 40-bit_depth 24-partition 4:}^{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}:-\text{end}\\$

$\text{dot-Tpng-}^{\text{graph_v5_e7.gv-}graph_v5_e7.png}\\$

$\text{#creates graph_v5_e7.csv}\\$

$\text{#creates graph_v5_e7.colored_graph}\\$

petersen:

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

$\text{-define G-graph-Johnson 6:}^{2,0}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-exports_csv-}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-exports_graphviz-}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-saves-}:-\text{end}\\$

$\{(\text{ORBITER\ PATH})\text{orbiter.out-v.2-draw_matrix}\\$

$\text{-input_csv_file-Johnson 6:2,0}:-\text{csv}\\$

$\text{-box_width 40-bit_depth 24-partition 4:}^{10,10,10,10,10,10,10,10}:-\text{end}\\$

$\text{dot-Tpng-Johnson 6:2,0.gv-Johnson 6:2,0.png}\\$

Johnson 6:2,0:

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

$\text{-define G-graph-Johnson 6:2,0}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-exports_csv-}:-\text{end}\\$

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

$\text{-graph_theoretic_activity-exports_graphviz-}:-\text{end}\\$
Hamming graph 3:
$(ORBITER PATH)orbiter.out -v 2 -draw_matrix -with G do -graph theoretic activity -save -end

Hamming graph 7:
$(ORBITER PATH)orbiter.out -v 2 -draw_matrix -with G do -graph theoretic activity -save -end

HJ properties:
$(ORBITER PATH)orbiter.out -v 6 -define G -graph -load csv no border
13203 ▶ ▶ ▶ halljanko315.csv
13204 ▶ ▶ -end
13205 ▶ ▶ -with-G-do
13206 ▶ ▶ ▶ -graph_theoretic_activity-properties
13207 ▶ ▶ -end
13208
13209 #Degree-type:(10^{-315})
13210
13211
13212
13213 HJ_d2_properties:
13214 ▶ $(ORBITER_PATH)orbiter.out-v.6
13215 ▶ ▶ -define-G-graph
13216 ▶ ▶ ▶ -load_csv_no_border
13217 ▶ ▶ ▶ halljanko315.csv
13218 ▶ ▶ ▶ -distance_2
13219 ▶ ▶ -end
13220 ▶ ▶ -with-G-do
13221 ▶ ▶ ▶ -graph_theoretic_activity
13222 ▶ ▶ ▶ -properties
13223 ▶ ▶ -end
13224
13225
13226 #Degree-type:(80^{-315})
13227
13228
13229 HJ_d2_c5:
13230 ▶ $(ORBITER_PATH)orbiter.out-v.6
13231 ▶ ▶ -define-G-graph
13232 ▶ ▶ ▶ -load_csv_no_border
13233 ▶ ▶ ▶ halljanko315.csv
13234 ▶ ▶ ▶ -distance_2
13235 ▶ ▶ -end
13236 ▶ ▶ -with-G-do
13237 ▶ ▶ -graph_theoretic_activity
13238 ▶ ▶ ▶ -find_cliques-target_size.5-end
13239 ▶ ▶ -end
13240
13241
13242 PGO_5.2_collinearity_graph:0.5.2.incidence_matrix.csv
13243 ▶ $(ORBITER_PATH)orbiter.out-v.3
13244 ▶ ▶ -define-Inc-vector-file:0.5.2_incidence_matrix.csv-end
13245 ▶ -define-Gamma-graph-collinearity_graph-Inc-end
13246 ▶ -with-Gamma-do
13247 ▶ -graph_theoretic_activity
13248 ▶ ▶ -properties
13249 ▶ ▶ -end

668
Section 11.2: Graphs Theoretic Activities

SECTION_GRAPH_THEORETIC_ACTIVITIES:

Chain_232_eigen:

```latex
\$\text{ORBITER\ PATH}\text{orbiter.out}\text{-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}
\text{-eigenvalues}
\text{-end}
\text{pdflatex\_chain\_graph\_eigenvalues.tex}
\text{open\_chain\_graph\_eigenvalues.pdf}
```

Section 11.3: Graph Theory: Classification

SECTION_GRAPH_THEORY_CLASSIFICATION:

```latex
graph_classify_4:
\$\text{ORBITER\ PATH}\text{orbiter.out}\text{-v.2}\$
\text{-define-GC\-graph\_classification}
\text{-n\-4\-end}
\text{-poset\_classification\_control}
\text{-problem\_label\_graphs\_v4\-depth\-6\-draw\_poset}
\text{-draw\_options\-radius\-250\-embedded\-end}
\text{-end}
\text{-with-GC\-do}
\text{-graph\_classification\_activity}
\text{-draw\_options\-embedded\-radius\-400}
\text{-line\_width\-2\-scale\-0.15\-end}
\text{-draw\_graphs\_at\_level\-3}
```

669
13295 ▷ ▷ -print_symbols
13296 ▷ pdflatex-graphs_v4_rep_3_2.tex
13297 ▷ open-graphs_v4_rep_3_2.pdf
13298 ▷ #pdflatex-graphs_v4_poset_detailed_lvl_6.tex
13299 ▷ #open-graphs_v4_poset_detailed_lvl_6.pdf
13300 ▷ #pdflatex-graphs_v4_poset_lvl_6.tex
13301 ▷ #open-graphs_v4_poset_lvl_6.pdf
13302
13303 tournament_classify_4:
13304 ▷ $(ORBITER_PATH)orbiter.out-v.2\`
13305 ▷ ▷ -define-GC-graph_classification\`
13306 ▷ ▷ ▷ -n.4-tournament\`
13307 ▷ ▷ ▷ -poset_classification_control\`
13308 ▷ ▷ ▷ ▷ -problem_label.tournament.4-depth.6-draw_poset\`
13309 ▷ ▷ ▷ ▷ ▷ -draw_options-radius.250-embedded-end\`
13310 ▷ ▷ ▷ ▷ ■ -end\`
13311 ▷ ▷ ▷ -end\`
13312 ▷ ▷ -end\`
13313 ▷ ▷ -with-GC-do\`
13314 ▷ ▷ -graph_classification_activity\`
13315 ▷ ▷ ▷ -draw_options-embedded-radius.400\`
13316 ▷ ▷ ▷ ▷ -line_width.2-scale.0.15-end\`
13317 ▷ ▷ ▷ ▷ -draw_graphs_at_level.6\`
13318 ▷ ▷ ▷ -end\`
13319 ▷ ▷ -print_symbols
13320 ▷ pdflatex-tournament_4_rep_6_0.tex
13321 ▷ pdflatex-tournament_4_rep_6_1.tex
13322 ▷ pdflatex-tournament_4_rep_6_2.tex
13323 ▷ pdflatex-tournament_4_rep_6_3.tex
13324 ▷ open-tournament_4_rep_6_0.pdf
13325 ▷ open-tournament_4_rep_6_1.pdf
13326 ▷ open-tournament_4_rep_6_2.pdf
13327 ▷ open-tournament_4_rep_6_3.pdf
13328 ▷
13329
13330 graph_classify_8_r3:
13331 ▷ $(ORBITER_PATH)orbiter.out-v.3\`
13332 ▷ ▷ -define-GC-graph_classification\`
13333 ▷ ▷ ▷ -n.8-regular.3\`
13334 ▷ ▷ ▷ -poset_classification_control\`
13335 ▷ ▷ ▷ ▷ -problem_label.graphs_v8_r3-depth.12-draw_poset\`
13336 ▷ ▷ ▷ ▷ ▷ -draw_options-radius.250\`
13337 ▷ ▷ ▷ ▷ ▷ ▷ -line_width.2-embedded-end\`
13338 ▷ ▷ ▷ ▷ ▷ ■ -end\`
13339 ▷ ▷ ▷ ■ -end\`
13340 ▷ ▷ -with-GC-do\`
13341 ▷ ▷ ■ -with-GC-do\`
13342 ▷ ▶ -graph_classification_activity\  
13343 ▷ ▶ ▶ -draw_options: -embedded: -radius: 400\  
13344 ▷ ▶ ▶ ▶ -line_width: 2 -end\  
13345 ▷ ▶ ▶ -draw_graphs_at_level: 12\  
13346 ▷ ▶ -end\  
13347 ▷ ▶ -print_symbols
13348 ▷ pdflatex-graphs_v8_r3_poset_lvl_12.tex
13349 ▷ open-graphs_v8_r3_poset_lvl_12.pdf
13350 ▷ #pdflatex-graphs_v8_r3_rep_12_0.tex
13351 ▷ #open-graphs_v8_r3_rep_12_0.pdf
13352 ▷ #pdflatex-graphs_v8_r3_rep_12_1.tex
13353 ▷ #open-graphs_v8_r3_rep_12_1.pdf
13354 ▷ #pdflatex-graphs_v8_r3_rep_12_2.tex
13355 ▷ #open-graphs_v8_r3_rep_12_2.pdf
13356 ▷ #pdflatex-graphs_v8_r3_rep_12_3.tex
13357 ▷ #open-graphs_v8_r3_rep_12_3.pdf
13358 ▷ #pdflatex-graphs_v8_r3_rep_12_4.tex
13359 ▷ #open-graphs_v8_r3_rep_12_4.pdf
13360 ▷ #pdflatex-graphs_v8_r3_rep_12_5.tex
13361 ▷ #open-graphs_v8_r3_rep_12_5.pdf
13362
13363
13364
13365 # Section 11.4: Graph Theory: Clique-finding
13366
13367
13368
13369 SECTION_GRAPH_THEORY клик FINDING:
13370
13371
13372 small_graph_cliques:
13373 ▷ $(ORBITER_PATH)orbiter.out -v: 10\  
13374 ▷ ▶ -define G -graph -load_graph -v5_e7.colored_graph -end\  
13375 ▷ ▶ -with G -do\  
13376 ▷ ▶ -graph_theoretic_activity -find_cliques -target_size: 3 -end
13377
13378 # nb_sol = 3
13379
13380 small_graph_cliques_Sajeeb:
13381 ▷ $(ORBITER_PATH)orbiter.out -v: 2\  
13382 ▷ ▶ -define G -graph -load_graph -v5_e7.colored_graph -end\  
13383 ▷ ▶ -with G -do\  
13384 ▷ ▶ -graph_theoretic_activity -find_cliques -Sajeeb -target_size: 3 -end
13385
13386 # nb_sol = 3
13387

671
Paley_13.aut:
  $(ORBITER_PATH)orbiter.out-v.2\n  -define-Gamma-graph-Paley_13\n  -with-Gamma\n  -graph_theoretic_activity-automorphism_group\n  -end:

# writes Paley_13_group.makefile
User.time: 0 of a second, dt=0 tps=-100
nb_calls_to_densenauty=1

Paley_13:
  $(ORBITER_PATH)orbiter.out-v.2\n  -define-gens-vector-file-Paley_13.gens.csv\n  -define-G-permutation_group\n  -bsgs-Paley_13:"Paley_13".13.78."0,1".3.gens\n  -end:

Paley_13.cliques_classify:
  $(ORBITER_PATH)orbiter.out-v.4\n  -define-gens-vector-file-Paley_13.gens.csv\n  -define-G-permutation_group\n  -bsgs-Paley_13:"Paley_13".13.78."0,1".3.gens\n  -end:

  -group_theoretic_activity\n  -poset_classification_control\n  -W\n  -problem_label:Paley13.cliques\n  -clique_test:Gamma\n  -depth:5\n  -end:
  -orbits_on_subsets:5\n  -report:1\n  -end

User.time: 0.01 of a second, dt=1 tps=-100

Paley_13.cliques:
  $(ORBITER_PATH)orbiter.out-v.10\n  -define-Gamma-graph-Paley_13\n  -with-Gamma\n  -graph_theoretic_activity-find_cliques-target_size:3\n  -end

#User.time: 0.01 of a second, dt=1 tps=-100

Paley_13.cliques:
PG0_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

HJ_d2_c5_second:

$\{ORBITER\_PATH\}orbiter.out -v 6 $

-define G -graph

-load_csv_no_border

-halljanko315.csv

-distance 2

-end

-with G -do

-graph_theoretic_activity

-find_cliques -target_size 5 -end

-end

#graph_theoretic_activity::perform_activity Gr -> label=halljanko315 nb sol = 26208 0

HJ64_cliques5:

$\{ORBITER\_PATH\}orbiter.out -v 6 $

-define Gamma -graph

-load

-Group_Perms15_Orbital_3.colored_graph

-end

-with Gamma -do

-graph_theoretic_activity

-find_cliques -target_size 5 -end

-end

#graph_theoretic_activity::perform_activity Gr -> label=Group_Perms15_Orbital_3 nb sol = 1008

Group_Perms15_Orbital_3.sol.csv
HJ64_cliques5_classify:

```
$ (ORBITER_PATH) orbiter.out -v 6 \
\> define Gamma graph \n\> load \n\> Group Perm315 Orbital_3 colored graph \n\> -end \n\> define gens vector \n\> -file halljanko315 gens.csv \n\> -end \n\> define G permutation group \n\> bsgs halljanko315 "File\_halljanko315" \n\> 315 1209600 "0,1,42,95" 6 gens -end \n\> -with G do \n\> -group theoretic activity \n\> -poset classification control \n\> -w \n\> -problem label HJ64 cliques \n\> -clique test Gamma \n\> -depth 5 \n\> -end \n\> -orbits on subsets 5 \n\> -report \n\> -end
```

#HJ64_cliques_reps_lvl_5.csv

```
Chapter 12 -- Canonical Forms with Nauty
```

#HJ64_cliques_reps_lvl_5.csv
# Section 12.1: Overview of Canonical Forms

SECTION_OVERVIEW_CANONICAL_FORMS:

# Section 12.2: Objects in Projective Space

SECTION_OBJECTS_IN_PROJECTIVE_SPACE:

```
EC_canon: elliptic_curve_b1_c3_q11.txt
> $(ORBITER_PATH)orbiter.out -v 4
> -define C -combinatorial_objects
> -file_of_points_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
> -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
```

675
Hirschfeld_q4.c: Hirschfeld_surface_q4.txt

$(ORBITER_PATH) orbiter.out -v 40

-define C combinatorial_objects

-define F finite_field q 4

-define P projective_space 3 F

-with C do

-define H set here

-HIRSCHFELD_SURFACE Q4 SET OF POINTS

-end

-define C combinatorial_objects

-set of points H

-end

-define F finite_field q 4

-define P projective_space 3 F

-with C do

-combinatorial_object_activity

-cannonical_form PG P

-classification_prefix Hirschfeld_surface_q4

-save ago

-max TDO depth 10

-report

-show TDO

-end

-Hirschfeld_surface_q4_classification.tex

-open Hirschfeld_surface_q4_classification.pdf

-Hirschfeld_q4_set.c:

$(ORBITER_PATH) orbiter.out -v 40

-define H set here

-HIRSCHFELD_SURFACE Q4 SET OF POINTS

-end

-define C combinatorial_objects

-set of points H

-end

-define F finite_field q 4

-define P projective_space 3 F

-with C do

-combinatorial_object_activity

-cannonical_form PG P

-classification_prefix Hirschfeld_surface_q4

-save ago

-end

-Hirschfeld_surface_q4_classification.tex

-open Hirschfeld_surface_q4_classification.pdf
Dickson_sets_stabilizer:

```
$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 3$
\define-C=\text{combinatorial\_objects}$
\define=\text{set\_of\_points}="0,1,2,5,6"
\define=\text{set\_of\_points}="0,1,2,3,6"
\define=\text{set\_of\_points}="0,1,2,3,4"
\define=\text{set\_of\_points}="0,1,2,3,8"
\define=\text{set\_of\_points}="0,1,2,5,6,7,8"
\define=\text{set\_of\_points}="0,1,2,3,5,6,7"
\define=\text{set\_of\_points}="0,1,2,3,5,6,9"
\define=\text{set\_of\_points}="0,1,2,3,5,6,10"
\define=\text{set\_of\_points}="0,1,2,3,5,6,4"
\define=\text{set\_of\_points}="0,1,2,3,8,11,13"
\define=\text{set\_of\_points}="3,6,9,7,10,12,8,11,13,14,4"
\define=\text{set\_of\_points}="3,5,6,9,7,10,12,11,13,14,4"
\define=\text{set\_of\_points}="0,1,2,3,5,6,9,7,10,12,4"
\define=\text{end}$
\define=\text{combinatorial\_object\_activity}$
\define=\text{canonical\_form\_PG-P}$
\define=\text{classification\_prefix-Dickson\_sets}$
\define=\text{save\_ago}$
\define=\text{end}$
\define=\text{pdflatex-Dickson\_sets\_classification.tex}$
```

```
open-Dickson\_sets\_classification.pdf
```

```
Endrass_7c::Endrass_F7.txt
```

```
$\$(\text{ORBITER\_PATH})\text{orbiter.out}\ -v\ 2$
\define-C=\text{combinatorial\_objects}$
\define=\text{file\_of\_points-Endrass_F7.txt}$
\define=\text{end}$
\define=\text{finite\_field}\ -q\ 7$
\define=\text{projective\_space}\ -3 \text{F}$
\define=\text{with}\ -C$
\define=\text{do}$
```

```
\define=\text{combinatorial\_object\_activity}$
\define=\text{canonical\_form\_PG-P}$
\define=\text{classification\_prefix-Endrass_F7}$
\define=\text{save\_ago}$
```

13664 \texttt{end}\ \\
13665 \texttt{-report}\ \\
13666 \texttt{-end}\ \\
13667 \texttt{pdflatex Endrass\_F7\_classification.tex}\ \\
13668 \texttt{open Endrass\_F7\_classification.pdf}\ \\
13669 \ \\
13670 \ \\
13671 \texttt{# group order is 32}\ \\
13672 \ \\
13673 \ \\
13674 \texttt{hyperoval\_16\_c:\}\ \\
13675 \texttt{$\langle ORBITER\_PATH\rangle\ orbiter\_out-v.2\}\ \\
13676 \texttt{\triangleright define C combinatorial objects}\ \\
13677 \texttt{\triangleright set of points $(HYPEROVAL\_16\_16320)\}\ \\
13678 \texttt{\triangleright set of points $(HYPEROVAL\_16\_144)\}\ \\
13679 \texttt{\triangleright end}\ \\
13680 \texttt{\triangleright define F finite field- q 16 - end}\ \\
13681 \texttt{\triangleright define P projective space 2 F - end}\ \\
13682 \texttt{\triangleright with C do}\ \\
13683 \texttt{\triangleright combinatorial object activity}\ \\
13684 \texttt{\triangleright canonical form PG P}\ \\
13685 \texttt{\triangleright \triangleright \triangleright \texttt{classification prefix hyperoval q16}}\ \\
13686 \texttt{\triangleright \triangleright \triangleright \texttt{label hyperoval q16}}\ \\
13687 \texttt{\triangleright \triangleright \triangleright \texttt{-save ago}}\ \\
13688 \texttt{\triangleright \triangleright \triangleright \texttt{-save transversal}}\ \\
13689 \texttt{\triangleright \triangleright \triangleright \texttt{-max TDO depth 10}}\ \\
13690 \texttt{\triangleright \triangleright \triangleright \texttt{-end}}\ \\
13691 \texttt{\triangleright \triangleright \texttt{-report}}\ \\
13692 \texttt{\triangleright \triangleright \texttt{-prefix hyperoval q16}}\ \\
13693 \texttt{\triangleright \triangleright \texttt{-export flag orbits}}\ \\
13694 \texttt{\triangleright \triangleright \texttt{-show TDO}}\ \\
13695 \texttt{\triangleright \triangleright \texttt{-show TDA}}\ \\
13696 \texttt{\triangleright \triangleright \texttt{-dont show incidence matrices}}\ \\
13697 \texttt{\triangleright \triangleright \texttt{-export group}}\ \\
13698 \texttt{\triangleright \triangleright \texttt{-end}}\ \\
13699 \texttt{\triangleright \texttt{-end}}\ \\
13700 \texttt{\texttt{pdflatex hyperoval\_16\_classification.tex}}\ \\
13701 \texttt{\texttt{open hyperoval\_16\_classification.pdf}}\ \\
13702 \texttt{$\langle ORBITER\_PATH\rangle\ orbiter\_out-v.2\-draw\_matrix}\ \\
13703 \texttt{\triangleright \texttt{-input csv file hyperoval\_16\_object0\_TDA\_flag\_orbits.csv}}\ \\
13704 \texttt{\triangleright \texttt{-secondary input csv file hyperoval\_16\_object0\_TDA.csv}}\ \\
13705 \texttt{\triangleright \texttt{-box width 4 \-bit depth 24}}\ \\
13706 \texttt{\triangleright \texttt{-end}}\ \\
13707 \texttt{\texttt{open hyperoval\_16\_object0\_TDA\_flag\_orbits\_draw.bmp}}\ \\
13708 \texttt{$\langle ORBITER\_PATH\rangle\ orbiter\_out-v.2\-draw\_matrix}\ \\
13709 \texttt{\triangleright \texttt{-input csv file hyperoval\_16\_object1\_TDA\_flag\_orbits.csv}}\ \\
13710 \texttt{\triangleright \texttt{-secondary input csv file hyperoval\_16\_object1\_TDA.csv}}
13711 \> \> -box_width 4 -bit_depth 24 \-
13712 \> \> -end
13713 \> \> open-hyperoval_q16_object1_TDA_flag_orbits_draw.bmp
13714
13715
13716
13717
13718
13719 cubic_curves_PG_2_8.canon:
13720 \> $(ORBITER_PATH)/orbiter.out -v 6 \-
13721 \> \> -define C -combinatorial_objects \-
13722 \> \> \> -set_of_points "2, 3, 28, 46, 51, 61, 40, 71" \-
13723 \> \> -end \-
13724 \> \> -define F -finite_field -q 8 \-
13725 \> \> -define P -projective_space 2 -F \-
13726 \> \> -with C \-
13727 \> \> -combinatorial_object_activity \-
13728 \> \> \> -canonical_form PG-P \-
13729 \> \> \> \> -classification_prefix cc 8 \-
13730 \> \> \> \> -save ago \-
13731 \> \> \> \> -max TDO depth 10 \-
13732 \> \> \> -end \-
13733 \> \> -report \-
13734 \> \> -end
13735 \> pdflatex cc_8_classification.tex
13736 \> open cc_8_classification.pdf
13737
13738
13739 F_alpha_beta_gamma_delta_classify_q7_nauty:
13740 \> $(ORBITER_PATH)/orbiter.out -v 2 \-
13741 \> \> -define C -combinatorial_objects \-
13742 \> \> \> -file_of_points \-
13743 \> \> \> F_alpha_beta_gamma_delta_q7_points.txt \-
13744 \> \> -end \-
13745 \> \> -define F -finite_field -q 7 \-
13746 \> \> -define P -projective_space 3 -F \-
13747 \> \> -with C \-
13748 \> \> -combinatorial_object_activity \-
13749 \> \> \> -canonical_form PG-P \-
13750 \> \> \> -classification_prefix surface 15 lines q7 \-
13751 \> \> \> -save ago \-
13752 \> \> -end
13753 \> #pdflatex surface_15_lines_q7_classification.tex
13754 \> #open surface_15_lines_q7_classification.pdf
13755
13756
13757 \#User: time: 4:12 on Mac
# Section 12.3: Incidence Geometries

SECTION INCIDENCE GEOMETRIES:

geo_7_3.c:

```bash
d $(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:7.21\  
  -end\  
  -with-C-do\  
  -combinatorial_object_activity\  
  -canonical_form\  
  -classification_prefix.7_3\  
  -label.7_3\  
  -save.ago\  
  -save_transversal\  
  -end\  
  -report\  
  -prefix.7_3\  
  -export_flag_orbits\  
  -show_incidence_matrices\  
  -export_group\  
  -end\  
  -end
```

d \( \text{pdf}\text{latex}\_7.3\text{\_classification.tex} \)

d \( \text{open}\_7.3\text{\_classification.pdf} \)

```bash
d $(ORBITER_PATH)orbiter.out-v.2-draw_matrix\  
  -input_csv_file.7_3\_object0\_TDA_flag_orbits.csv\  
  -secondary_input_csv_file.7_3\_object0\_TDA.csv\  
  -box_width-32-bit_depth-24\`
$(ORBITER \ PATH)\ orbiter.\ out\ -v.2\ -draw\ matrix\ \$
-\input\_csv\_file\_7.3\_object0\_INP\_flag\_orbits.csv\$
-\box\_width\_32\ -\bit\_depth\_24\$
-open\_7.3\_object0\_INP\_flag\_orbits\_draw.bmp

ge_10.3.c:
$(ORBITER \ PATH)\ orbiter.\ out\ -v.10\$
-\draw\_incidence\_structure\_description\$
-\width\_60\ -\with\_10.6\ -end$
-\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\_FFFF$
-\-end$
-\define\_C\_combinatorial\_objects$
-\file\_of\_incidence\_geometries\_10.3.\_inc\_10.10.30$
-\-end$
-\with\_C\_do$
-\combinatorial\_object\_activity$
-\canonical\_form$
-\define\_classification\_prefix\_10.3$
-\label\_10.3$
-\save\_ago$
-\save\_transversal$
-\-end$
-\prefix\_10.3$
-\export\_flag\_orbits$
-\show\_incidence\_matrices$
-\export\_group$
-\lex\_least\_Geo$
-\-end$
-pdflatex\_10.3\_classification.tex
-open\_10.3\_classification.pdf
$(ORBITER \ PATH)\ orbiter.\ out\ -v.2\ -draw\ matrix\$
-\input\_csv\_file\_10.3\_object7\_TDA\_flag\_orbits.csv$
-\secondary\_input\_csv\_file\_10.3\_object7\_TDA.csv$
-\box\_width\_16\ -\bit\_depth\_24$
-\-end
$(ORBITER \ PATH)\ orbiter.\ out\ -v.2\ -draw\ matrix\$
-input_csv_file:10.3_object7_INP_flag_orbits.csv
-secondary_input_csv_file:10.3_object7_INP.csv
-box_width:16-bit_depth:24
-end

#10.3_object7_TDA_flag_orbits.csv

#0.1.2.10.13.14.20.25.26.31.33.35.41.44.46.52.53.56.62.67.68.74.77.79.85.88.89.

geo_15.3.c:

13865 $(ORBITER_PATH)orbiter.out-v.2-
13866 -draw_incidence_structure_description-
13867 -width:50-with:10.5-end-
13868 -define-C-combinatorial_objects-
13869 -file_of_incidence_geometries:15.3.inc:15.15.45-
13870 -end-
13871 -with:C-do-
13872 -combinatorial_object_activity-
13873 -canonical_form-
13874 -classification_prefix:10.3-
13875 -label:10.3-
13876 -save_ago-
13877 -end-
13878 TFC_24.3.c:
13879 echo-$FILE_24.3_TFC_INC>-24.3_TFC.inc
13880 $(ORBITER_PATH)orbiter.out-v.6-
13881 -define-C-combinatorial_objects-
13882 -file_of_incidence_geometries:24.3_TFC.inc:24.24.72-
13883 -end-
13884 -with:C-do-
13885 -combinatorial_object_activity-
13886 -canonical_form-
13887 -classification_prefix:24.3_TFC-
13888 -label:24.3_TFC-
13889 -save_ago-
13890 -end-
13891 -report-
13892 -prefix:24.3_TFC-
13893 -export_flag_orbits-
13894 -show_TDO-
13895 -show_incidence_matrices-
13896 -end-
13897 -end
13898 \>
13899 \>
13900 \>
13901 \>
13902 \>
13903 \>
13904 \>
13905 \>
13906 \>
13907 \>
13908 geo_40_4_g4.c:
13909 \>
13910 \>
13911 \>
13912 \>
13913 \>
13914 \>
13915 \>
13916 \>
13917 \>
13918 \>
13919 \>
13920 \>
13921 \>
13922 \>
13923 \>
13924 \>
13925 \>
13926 \>
13927 \>
13928 \>
13929 AG_2_3.c:AG_2_3.inc
13930 \>
13931 \>
13932 \>
13933 \>
13934 \>
13935 \>
13936 \>
13937 \>
13938 \>
13939 \>
13940 \>
13941 \>
13942 \>
13943 \>
13944 \>

683
TDO
3
3
canonical:
points
object0
INP
draw.bmp
space
2
of
classification.pdf
0
csv
orbits.csv
orbits
points
object0
matrices
INP
PATH)orbiter.out
prefix
2
25
depth
3
classification.tex
file
of
activity
25
curve
field
0
space
3
points
of
0
classification.pdf
points
object0
2
0
file
orbits
0
form
PG
0
PATH)orbiter.out
points
3
of
25
2
matrix
INP.csv
2
0
input
0
\begin{verbatim}
13945 \> \> \> \> -export_flag_orbits\\
13946 \> \> \> \> -show_TDO\\
13947 \> \> \> \> -show_TDA\\
13948 \> \> \> \> -show_incidence_matrices\\
13949 \> \> \> \> -end\\
13950 \> \> -end
13951 pdflatex-AG.2_3_classification.tex
13952 open-AG.2_3_classification.pdf
13953 \$(\text{ORBITER\_PATH})/\text{orbiter.out}\ -v\ -2\ -draw\_matrix\\
13954 \> -input\_csv\_file\_AG.2_3\_object0\_INP\_flag\_orbits.csv\\
13955 \> -secondary\_input\_csv\_file\_AG.2_3\_object0\_INP.csv\\
13956 \> -box\_width\ 40\ -bit\_depth\ 24\\
13957 \> -end
13958 open-AG.2_3\_object0\_INP\_flag\_orbits\_draw.bmp
13959
13960
13961
13962 \#\text{ToDo:}
13963
13964 quartic\_curve\_25.0_0\_canonical:\
13965 \$(\text{ORBITER\_PATH})/\text{orbiter.out}\ -v\ -3\\
13966 \> -define\_F\ -finite\_field\ -q\ 25\ -end\\
13967 \> -define\_F\ -projective\_space\ -2\_F\ -end\\
13968 \> -with\_P\ -do\\
13969 \> -projective\_space\_activity\\
13970 \> -canonical\_form\_PG\\
13971 \> -input\\
13972 \> -set\_of\_points:10,11,59,63,124,135,136,170,206,257,275,284,285,367,378,393,433,619,641,644\\
13973 \> -set\_of\_points:9,24,62,67,77,84,87,89,125,130,158,172,197,219,266,271,325,356,391,392,400,429,454,458,470,503,531,553,605,625,627,646\\
13976 \> -set\_of\_points:2,12,48,65,87,120,189,246,305,323,354,375,434,435,455,482,496,557,586,595\\
13977 \> -end\\
13978 \> -classification\_prefix\_quartic\_25.0_0\\
13979 \> -report\\
13980 \> -end\\
13981 \> -end
13982 pdflatex-quartic\_25.0_0\_classification.tex
13983 open-quartic\_25.0_0\_classification.pdf
\end{verbatim}
Section 12.4: Objects from Design Theory

SECTION_OBJECTS_FROM_DESIGN THEORY:

LS_AG_2_3_solutions.classify:

SAVE AG

SOLUTIONS

CLASSIFICATION

PREFIX LS AG 2 3

INPUT CSV FILE

ORBITER PATH orbiter.out

-- draw incidence structure description

--width 20 --width 10 2

--define C --combinatorial objects

--file of designs

--solutions.csv 9 84 3 12

--end

--with C

--combinatorial object activity

--canonical form

--save ago

--save transversal

--classification prefix LS AG 2 3

--label LS AG 2 3

--max TDO depth 10

--end

--report

--prefix LS AG 2 3

--export flag orbits

--show TDO

--end

pdflatex LS AG 2 3_classification.tex

open LS AG 2 3_classification.pdf

$(ORBITER_PATH) orbiter.out --v 2 --draw matrix

--input.csv file LS AG 2 3 object 0_INP.flag.orbits.csv

--secondary input csv file LS AG 2 3 object 0_INP.csv

--box width 12 --bit depth 24

--end

open LS AG 2 3 object 0_INP.flag.orbits draw.bmp

$(ORBITER_PATH) orbiter.out --v 2 --draw matrix

--input csv file LS AG 2 3 object 1_INP.flag.orbits.csv

--end

685
-secondary_input_csv_file:LS_AG_2_3_object1_INP.csv\n-box_width:12-bit_depth:24\n-end\nopen:LS_AG_2_3_object1_INP_flag_orbits.draw.bmp

 iso-type:input-object:ago
 iso:0:0:42
 iso:1:3:54

 design_27c:
 $(ORBITER_PATH)orbiter.out-v.4\n -define:C-combinatorial_objects\n -set_of_points:"2,56,30,112,253,90,440,508."
 -end\n -define:F-finite_field-q=27
 -override_polynomial=46
 -end\n -define:P-projective_space:F=2
 -with:C=do\n -combinatorial_object_activity\n -canonical_form_PG_P\n -classification_prefix-design\n -end\n -end\n -report\n -end

 -combinatorial_object_activity\n -canonical_form_PG_P\n -classification_prefix-design\n -end\n -report\n -end

 pdflatex:design_classification.tex
 open:design_classification.pdf

 Section 12.5: Linear Codes

 SECTION_CANONICAL_FORMS_OF_LINEAR_CODES:
code_3_2.aut:

$\$(ORBITER\ PATH)\ orbiter.out\ -v\ 2.0\$

+ define-F\ -finite_field\ -q\ 2\ -end-

+ define-gemma\ -vector\ -field\ F\ -format\ 2-

+ dense-$(CODE\_N3\_K2\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -1\ -F\ -end-

+ with-P\ -do-

+ projective_space\ -activity-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -2\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -1\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -2\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -1\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -2\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -1\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-

+ define-P\ -projective_space\ -2\ -F\ -end-

+ with-P\ -do-

+ canonical_form_of_code-

+ define-gemma\ -vector\ -field\ F\ -format\ 3-

+ dense-$(CODE\_N6\_K3\_Q2\_GENMA)-

+ end-
14123 \triangleright \triangleright \text{-box_width:16-bit_depth:24}\triangleright \triangleright \text{-end}
14124 \triangleright \text{-open:6_3_object0_TDA_flag.orbits.draw.bmp}
14126
14127 \#group of order:24
14128
14129 RM_3_1 group:
14130 \triangleright \$(\text{ORBITER PATH})\text{orbiter.out}--v:2\triangleright \text{-define:F-finite_field-q:2-end}\triangleright \text{-define-gemma-vector-field:F-format:4}\triangleright \end{-compact:\$(\text{CODE\_RM\_3\_1\_GENMA})}\triangleright \text{-define:P-projective_space:3-F-end}\triangleright \end{with:P-do}\triangleright \end{-projective_space_activity}\triangleright \end{-canonical_form_of_code}\triangleright \$\text{"RM\_3\_1"-genma-save_ago-label:"RM\_3\_1"}\triangleright \end{-classification_prefix:"RM\_3\_1"}\triangleright \end{-end}\triangleright \text{-end}
14144 \text{pdflatex:RM\_3\_1\_classification.tex}
14145 \text{open:RM\_3\_1\_classification.pdf}
14146
14147 \#group order:1344
14148 \text{RM\_3\_1\_object0\_INP\_flag.orbits.csv}
14149
14150 RM_3_1 group and diagram:
14151 \triangleright \$(\text{ORBITER PATH})\text{orbiter.out}--v:2\triangleright \text{-define:F-finite_field-q:2-end}\triangleright \text{-define-gemma-vector-field:F-format:4}\triangleright \end{-compact:\$(\text{CODE\_RM\_3\_1\_GENMA})}\triangleright \text{-define:P-projective_space:3-F-end}\triangleright \end{with:P-do}\triangleright \end{-projective_space_activity}\triangleright \end{-canonical_form_of_code}\triangleright \$\text{"RM\_3\_1"-genma-save_ago-label:"RM\_3\_1"}\triangleright \end{-classification_prefix:"RM\_3\_1"}\triangleright \end{-end}\triangleright \text{-end}
14156 \text{pdflatex:RM\_3\_1\_classification.tex}
14157 \text{open:RM\_3\_1\_classification.pdf}
14158 \triangleright \$(\text{ORBITER PATH})\text{orbiter.out}--v:2--\text{-draw_matrix}\triangleright \text{-input_csv_file:RM\_3\_1\_object0\_INP\_flag.orbits.csv}\triangleright \text{-secondary_input_csv_file:RM\_3\_1\_object0\_INP.csv}\triangleright \text{-box_width:16-bit_depth:24\text{-end:}}
14170 \ > \ > \ -end
14171 \ > \ $(\texttt{ORBITER\_PATH})\texttt{oriber.out\_v.2\_draw_matrix}\$
14172 \ > \ > \ -input\_csv\_file\_RM\_3\_1\_object0\_TDA\_flag\_orbits.csv\$
14173 \ > \ > \ -secondary\_input\_csv\_file\_RM\_3\_1\_object0\_TDA.csv\$
14174 \ > \ > \ -box\_width\_16\_bit\_depth\_24\$
14175 \ > \ > \ -end
14176 \ > \ open\_RM\_3\_1\_object0\_INP\_flag\_orbits\_draw.bmp
14177 \ > \ open\_RM\_3\_1\_object0\_TDA\_flag\_orbits\_draw.bmp
14178
14179
14180
14181 \ RM\_4\_1\_group:
14182 \ > \ $(\texttt{ORBITER\_PATH})\texttt{oriber.out\_v.2}\$
14183 \ > \ > \ -define\_F\_finite\_field\_q\_2\_end\$
14184 \ > \ > \ -define\_gemma\_vector\_field\_F\_format\_5\$
14185 \ > \ > \ > \ -compact $(\texttt{CODE\_RM\_4\_1\_GENMA})$
14186 \ > \ > \ > \ -end$
14187 \ > \ > \ > \ -define\_P\_projective\_space\_4\_F\_end$
14188 \ > \ > \ > \ -with\_P\_do$
14189 \ > \ > \ > \ -projective\_space\_activity$
14190 \ > \ > \ > \ > \ -canonical\_form\_of\_code$
14191 \ > \ > \ > \ > \ > \ "RM\_4\_1\".gemma\_save\_ago\_label\_"RM\_4\_1\"$
14192 \ > \ > \ > \ > \ > \ > \ -classification\_prefix\_"RM\_4\_1\"$
14193 \ > \ > \ > \ > \ > \ > \ > \ -end$
14194 \ > \ > \ > \ > \ -end$
14195 \ > \ > \ > \ pdf\_latex\_RM\_4\_1\_classification.tex
14196 \ > \ > \ open\_RM\_4\_1\_classification.pdf
14197 \ > \ $(\texttt{ORBITER\_PATH})\texttt{oriber.out\_v.2\_draw_matrix}\$
14198 \ > \ > \ -input\_csv\_file\_RM\_4\_1\_object0\_INP\_flag\_orbits.csv\$
14199 \ > \ > \ -secondary\_input\_csv\_file\_RM\_4\_1\_object0\_INP.csv\$
14200 \ > \ > \ -box\_width\_16\_bit\_depth\_24$
14201 \ > \ > \ -end
14202 \ > \ $(\texttt{ORBITER\_PATH})\texttt{oriber.out\_v.2\_draw_matrix}\$
14203 \ > \ > \ -input\_csv\_file\_RM\_4\_1\_object0\_TDA\_flag\_orbits.csv\$
14204 \ > \ > \ -secondary\_input\_csv\_file\_RM\_4\_1\_object0\_TDA.csv\$
14205 \ > \ > \ -box\_width\_16\_bit\_depth\_24$
14206 \ > \ > \ -end
14207 \ > \ open\_RM\_4\_1\_object0\_INP\_flag\_orbits\_draw.bmp
14208 \ > \ open\_RM\_4\_1\_object0\_TDA\_flag\_orbits\_draw.bmp
14209
14210
14211 \ > \ \#\_group\_order\_322560\_\textasciitilde\_24\_30\_28\_16
14212
14213
14214 \ RS\_6\_4\_7\_group:
14215 \ > \ $(\texttt{ORBITER\_PATH})\texttt{oriber.out\_v.20}\$
14216 \ > \ > \ -define\_F\_finite\_field\_q\_7\_end$

689
\texttt{define-genma-vector-field-F-format-4} \backslash  \\
\texttt{compact-$\text{CODE_RS_6}_4_7$} \backslash  \\
\texttt{define-P-projective_space-3-F-end} \backslash  \\
\texttt{projective_space_activity} \backslash  \\
\texttt{canonical_form_of_code} \backslash  \\
\texttt{"RS_6\text{-genma-save}_ago\text{-label}"RS_6\text{-"} \backslash  \\
\texttt{classification_prefix} \text{"RS_6"} \backslash  \\
\texttt{-end} \backslash  \\
\texttt{-end} \backslash  \\
\texttt{pdflatex-RS_6\text{-classification}.tex} \backslash  \\
\texttt{open-RS_6\text{-classification}.pdf} \backslash  \\
\texttt{\$(\text{ORBITER PATH})\text{orbiter.out}\text{-v.2}\text{-draw_matrix} \backslash  \\
\texttt{-input_csv_file\text{-RS_6\_object0\text{-TDA.csv}} \backslash  \\
\texttt{-box\_width.2\text{-bit\_depth.24} \backslash  \\
\texttt{-end} \backslash  \\
\texttt{\#group\_of\_order.12} \backslash  \\
\texttt{\#set\_of\_points.\_0,9,51,344,253,3} \backslash  \\
\texttt{\#1,0,0,0,2,0,0,1,5,0,6,0,5,1,0,0,0,0,0,1,6,0,0,2,0,6,0,5,0,0,6,5,} \backslash  \\
\texttt{GV_n15_k6_d5\_group:} \backslash  \\
\texttt{\$(\text{ORBITER PATH})\text{orbiter.out}\text{-v.20} \backslash  \\
\texttt{-define-F-finite_field-q2\text{-end} \backslash  \\
\texttt{-define-genma-vector-field-F\_format-6} \backslash  \\
\texttt{-compact-$\text{CODE_GV_N15}_k6$\_} \backslash  \\
\texttt{\-end} \backslash  \\
\texttt{-end} \backslash  \\
\texttt{-projective_space_activity} \backslash  \\
\texttt{canonical_form_of_code} \backslash  \\
\texttt{"GV_n15_k6_d5\_genma-save}_ago\text{-label}"GV_n15_k6_d5\_\text{-"} \backslash  \\
\texttt{classification_prefix} \text{"GV_n15_k6_d5"} \backslash  \\
\texttt{-end} \backslash  \\
\texttt{-end} \backslash  \\
\texttt{pdflatex-GV_n15_k6_d5\_classification.tex} \backslash  \\
\texttt{open-GV_n15_k6_d5\_classification.pdf} \backslash  \\
\texttt{\#ago=12} \backslash  \\
\texttt{code_n15_k6_d6\_a\_group:} \backslash  \\
\texttt{\$(\text{ORBITER PATH})\text{orbiter.out}\text{-v.20} \backslash  \\
\texttt{-define-F-finite_field-q2\text{-end} \backslash  \\
\texttt{-define-genma-vector-field-F\_format-6} \backslash  \\
\texttt{690}
code_n15_k6_d6_b_group:

$(\text{ORBITER\_PATH})\text{orbiter}\_\text{out\_v\_20}\$

-define\_F\_finite\_field\_q\_2\_end

-define\_genma\_vector\_field\_F\_format\_6

-compact\_$\text{CODE}_{15\_6\_6\_B}\$

-define\_projective\_space\_5\_F\_end

-with\_P\_do

-projective\_space\_activity

-canonical\_form\_of\_code

"n15\_k6\_d6\_b\_a"\_\text{genma\_save\_ago\_label}\_"n15\_k6\_d6\_a\_a"

-classification\_prefix\_"n15\_k6\_d6\_a\_b"

-end

-section\_12\_6: General\_Codes

SECTION CANONICAL FORMS OF GENERAL CODES:

-section\_12\_7: Graphs
SECTION_CANonical_FORMs_OF_GRAphs:

Cycle_13_aut:

chain_graph_report.tex
open_chain_graph_report.pdf

JK_graph_pp16_1:

#nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling::nb_backtrack1=6

JK_graph_pp16_2:
JK_graph.pp16-9:
$($ORBITER_PATH$)orbiter.out -v.2\ 
$define-Gamma=graph-load_dimacs\ 
../JUNTTILA/KASKI/benchmarks/pp/pp16-9\ 
-end\ 
-with-Gamma-do\ 
-graph_theoretic_activity-save-end\ 
-with-Gamma-do\ 
-graph_theoretic_activity-save-end\ 
-with-Gamma-do\ 
-graph_theoretic_activity-automorphism_group-end\ 

JK_graph.grid.3.3:
$($ORBITER_PATH$)orbiter.out -v.2\ 
$define-Gamma=graph-load_dimacs\ 
../JUNTTILA/KASKI/benchmarks/grid/grid-w-3-3\ 
-end\ 
-with-Gamma-do\ 
-graph_theoretic_activity-save-end\ 
-with-Gamma-do\ 
-graph_theoretic_activity-save-end\ 
-with-Gamma-do\ 
-graph_theoretic_activity-automorphism_group-end\ 

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\ 

#does not finish

nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack1=4
nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack2=9
Written file grid-w-3-3_group.makefile.of.size 579
User.time: 0 of a second, dt=0 tps=100
nb_calls_to_densenauty=1

JK_graph.sts.13:
#nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack1 = 3

#nauty_interface_with_group::create_automorphism_group_of_graph_with_partition_and_labeling: nb_backtrack2 = 24

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

HJ group and orbits:

$(ORBITER_PATH)orbiter.out -v 2 \  
 DEFINE gens = vector = file \  
 LOAD\csv\ halljanko315.gens.csv \  
 END \  
 DEFINE G = permutation_group \  
 BSGS halljanko315 "File\ Halljanko315" \  
 315 1209600 0,1,2 gens \  
 END \  
 WITH G = do \  
 GROUP\theoretic\activity \  
 POSET\classification\control \  
 W \  
 PROBLEM\label HJ orbits \  
 DEPTH 2 \  
 END \  
 ORBITS\on\subsets 2 \  
 REPORT \  
 END

ROWS,REP,AGO,OL

0,0,1,96,12600

694
HJ orbital graph 3:

\[(\text{ORBITER PATH})\text{orbiter.out -v.2:\}
\]
\[
-\text{define-gens -vector -file:\}
\]
\[
-\text{define-G -permutation group:\}
\]
\[
-\text{bsgs-halljanko315 -File\halljanko315:\}
\]
\[
315 \cdot 1209600 \cdot 0,1,2 -\text{gens:}\]
\[
-\text{end:\}
\]
\[
-\text{define -Gamma -graph:\}
\]
\[
3 \cdot \text{orbital graph G:}\]
\[
-\text{end:\}
\]
\[
-\text{with -Gamma -do:\}
\]
\[
-\text{graph-theoretic activity:\}
\]
\[
-\text{properties:\}
\]
\[
-\text{end:\}
\]
\[
-\text{with -Gamma -do:\}
\]
\[
-\text{graph-theoretic activity:\}
\]
\[
-\text{save:\}
\]
\[
-\text{end}\]

Hamming graph 7 with Hamming code:

\[(\text{ORBITER PATH})\text{orbiter.out -v.2:\}
\]
\[
-\text{define-G -graph -Hamming -7.2:\}
\]
\[
-\text{subset -Hamming code:\}
\]
\[
-\text{end:\}
\]
\[
-\text{Hamming\ code:\}
\]
\[
-\text{end:\}
\]
\[
-\text{with -G -do:\}
\]
\[
-\text{graph-theoretic activity -export csv -end:\}
\]
\[
-\text{end:\}
\]
\[
-\text{with -G -do:\}
\]
\[
-\text{graph-theoretic activity -export graphviz -end:}\]
\[
-\text{end:\}
\]
\[
-\text{graph-theoretic activity -save -end:\}
\]
\[
-\text{end:}\]
\[
-\text{graph-theoretic activity -automorphism group -end}\]
\[
-\text{pdf\latex\Hamming_7_2_Hamming_code_report.tex}\]
\[
-\text{open \Hamming_7_2_Hamming_code_report.pdf}\]
\[
-\text{group of order -2688=16*16}\]
PGO_5_2_graph_group: 0.5_2.incidence_matrix.csv

$\texttt{(ORBITER\_PATH)orbiter.out\,-v\,-3}\$

-define Inc -vector -file 0.5_2.incidence_matrix.csv -end \n-define Gamma -graph -collinearity_graph Inc -end 
-with Gamma -do \n-graph_theoretic_activity \n
-automorphism_group 
-end 
-with Gamma -do 
-graph_theoretic_activity \n-eigenvalues 
-end 
-pdflatex collinearity_graph_eigenvalues.tex 
-open collinearity_graph_eigenvalues.pdf 

-define F -finite field -q 17 -end 
-with F -do -finite_field_activity 
- cheat_sheet_GF -end 
-pdflatex GF_17.tex 
-open GF_17.pdf 

# Section 12.7: Quartic Curves

SECTION_CANONICAL_FORMS_OF_QUARTIC_CURVES:

F_17_edge: 

-pdflatex GF_17.tex 
-open GF_17.pdf
Edge_curve_17:

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

-define F -finite_field -q 17 - end -

-define P -projective_space -2 F - end -

-with P - do -

-projective_space_activity -

-define object Edge - projective_v_r_i_o_r_i_e -

-define object Edge - projective_v_r_i_o_r_i_e -

-projective_space_activity -

-report -

-pdflatex by coefficients q17.tex

-open by coefficients q17.pdf

Edge_curve_17_line_type:

echo $(FILE Q17) > edge_q17.csv

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

-define F -finite_field -q 17 - end -

-define P -projective_space -2 F - end -

-with P - do -

-projective_space_activity -

-define object E -

-set $(EDGE CURVE Q17 AS POINTS) -

-end -
Edge_curve_17_nauty:

14600  $(ORBITER_PATH)orbiter.out-v3
14601  -define=C-combinatorial_objects:
14602  -set_of_points=$(EDGE_CURVE_Q17_AS_POINTS)
14603  -define=F-finite_field-q17:
14604  -define=P-projective_space:2F:
14605  -with=C--do:
14606  -define=C-combinatorial_object_activity:
14607  -define=C-canoncial_form_PG:P:
14608  -classification_prefix:Edge_curve_q17:
14609  -label:Edge_curve_q17:
14610  -save:ago
14611  -save:transversal:
14612  -max_TDO:depth:10:
14613  -report:
14614  -export:flag:orbits:
14615  -show_TDO:
14616  -show:TDA:
14617  -dont_show:incidence_matrices:
14618  -export:group:
14619  -end

Edge_curve_q17_classification.tex
open-Edge_curve_q17_classification.pdf

pdflatex-Edge_curve_q17_classification.tex
open-Edge_curve_q17_classification.pdf

$ORBITER_PATH)orbiter.out-v2-draw_matrix:

-input_csv_file:Edge_curve_q17_object0_TDA_flag_orbits.csv:

-secondary_input_csv_file:Edge_curve_q17_object0_TDA.csv:

-box_width:4-bit_depth:24:
14630 ▽ ▽ -end
14631 ▽ open-Edge_curve_q17_object0_TDA_flag_orbits_draw.bmp
14632
14633 #9-backtrack-nodes-total
14634
14635
14636 #aut.=.24
14637 #User.time:.04.of.a.second,.dt=4.tps=.100
14638
14639
14640 #generators-for-a-group-of-order.24:
14641 #1,0,0,0,13,0,0,0,4,.
14642 #1,0,0,0,0,16,0,16,0,.
14643 #0,1,16,2,4,4,15,4,4,
14644
14645
14646 Edge_curve_17_group:
14647 ▽ $(ORBITER_PATH)orbiter.out--v.3\n14648 ▽ ▽ -define-G=-linear_group--PGL3:17:\n14649 ▽ ▽ -subgroup_by_generators."Stab_Edge"."24".3\n14650 ▽ ▽ ▽ "1,0,0,0,13,0,0,0,4".\n14651 ▽ ▽ ▽ "1,0,0,0,0,16,0,16,0".\n14652 ▽ ▽ ▽ "0,1,16,2,4,4,15,4,4".\n14653 ▽ ▽ -end.\n14654 ▽ ▽ -with-G=.do.\n14655 ▽ ▽ -group_theoretic_activities.\n14656 ▽ ▽ -print_elements_tex.\n14657 ▽ ▽ -group_table.\n14658 ▽ ▽ -report.\n14659 ▽ ▽ -end
14660 ▽ pdflatex-PGL_3_17_Subgroup_Stab_Edge_24_report.tex
14661 ▽ open-PGL_3_17_Subgroup_Stab_Edge_24_report.pdf
14662
14663
14664 #the.last.element.is:
14665 #.0..1.1
14666 #15..1.16
14667 #15..16..1
14668
14669
14670
14671
14672 #################################################################################
14673
14674
14675
699
Edge_curve_23:

\$\text{ORBITER\_PATH}\text{orbiter.out}\text{-v.2}$

-define F -finite_field q 23 -end

-define P -projective_space 2 F -end

-with P -do -projective_space_activity -

-define object Edge -projective Variety -

-Edge 4 $(\text{EDGE\_CURVE}\_Q\text{23\_EQUATION})$

-define P -projective Space 2 F -end

-with P -do projective Space activity -

-define object Edge -projective Variety -

-Edge 4 $(\text{EDGE\_CURVE}\_Q\text{23\_EQUATION})$

-combinatorial object create::init created a set of size 48

(#-4,-25,-26,47,48,71,92,95,114,119,136,143,158,167,180,191,202,21
5,224,239,246,263,268,287,290,311,312,334,335,356,359,378,383,400
407,422,431,444,455,466,479,488,503,510,527,530,532,551)

-combinatorial object create::init created a set of size 48

-combinatorial object create::init created a set of size 48

-define C -combinatorial objects -

-set of points $(\text{EDGE\_CURVE}\_Q\text{23\_AS\_POINTS})$

-end -

-define F -finite_field q 23 -end

-define P -projective_space 2 F -end

-with C -do -

-combinatorial object activity -

-canonanical form PG -P-

-classification prefix Edge_curve_q23 -

-label Edge_curve_q23 -

-save ago -

-save transversal -

-max TDO depth 4 -

-end -

-report -

-prefix Edge_curve_q23 -

-export flag orbits -

-show TDO -

-show TDA -

-dont show incidence matrices -

-export group -

-end -

-pdflatex Edge_curve_q23_classification.tex

-open Edge_curve_q23_classification.pdf

700
#-76-backtrack-nodes-total
#User-time: 0.41 of a second, dt=41 tps=.100
#group-order: 1012=4*11*23

### \(\text{ORBITER PATH}\ orbiter.out \-v 3\)
### \(\text{ORBITER PATH}\ orbiter.out \-v 2\)
### \(\text{ORBITER PATH}\ orbiter.out \-v 1\)

### \(\text{ORBITER PATH}\ orbiter.out \-v 0\)

### \(\text{ORBITER PATH}\ orbiter.out \-v -1\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -2\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -3\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -4\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -5\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -6\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -7\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -8\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -9\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -10\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -11\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -12\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -13\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -14\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -15\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -16\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -17\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -18\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -19\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -20\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -21\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -22\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -23\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -24\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -25\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -26\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -27\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -28\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -29\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -30\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -31\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -32\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -33\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -34\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -35\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -36\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -37\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -38\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -39\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -40\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -41\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -42\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -43\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -44\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -45\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -46\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -47\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -48\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -49\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -50\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -51\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -52\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -53\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -54\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -55\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -56\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -57\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -58\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -59\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -60\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -61\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -62\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -63\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -64\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -65\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -66\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -67\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -68\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -69\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -70\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -71\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -72\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -73\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -74\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -75\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -76\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -77\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -78\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -79\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -80\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -81\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -82\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -83\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -84\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -85\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -86\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -87\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -88\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -89\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -90\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -91\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -92\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -93\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -94\)
### \(\text{ORBITER PATH}\ orbiter.out \-v -95\)
14766 \> echo $(FILE_Q25)\>edge_q25.csv
14767 \> $(ORBITER_PATH)oriter.out--v.2\>
14768 \> \> -define F\>finite_field\>-q.25\>-end\>
14769 \> \> -define P\>projective_space\>-2F\>-end\>
14770 \> \> -with P\>-do\>
14771 \> \> -projective_space_activity\>
14772 \> \> \> -define_object E\>
14773 \> \> \> \> -set $(EDGE_CURVE_Q25\_AS_POINTS)\>
14774 \> \> \> \> -end\>
14775 \> \> -end\>
14776 \> \> -with E\>-do\>
14777 \> \> -combinatorial_object_activity\>
14778 \> \> \> -save\>
14779 \> \> -end\>
14780 \> \> -with E\>-do\>
14781 \> \> -combinatorial_object_activity\>
14782 \> \> \> -line_type\>
14783 \> \> -end\>
14784 \> \> -print_symbols
14785
14786 #(4\^14, 3\^8, 2\^168, 1\^208, 0\^253)
14787
14788 Edge_curve_25_nauty:\>
14789 \> $(ORBITER_PATH)oriter.out--v.3\>
14790 \> \> -define C\>-combinatorial_objects\>
14791 \> \> \> -set_of_points $(EDGE_CURVE_Q25\_AS_POINTS)\>
14792 \> \> -end\>
14793 \> \> -define F\>finite_field\>-q.25\>-end\>
14794 \> \> -define P\>projective_space\>-2F\>-end\>
14795 \> \> -with C\>-do\>
14796 \> \> -combinatorial_object_activity\>
14797 \> \> \> -canonical_form PG-P\>
14798 \> \> \> \> -classification_prefix Edge_curve_q25\>
14799 \> \> \> \> -label Edge_curve_q25\>
14800 \> \> \> \> -save ago\>
14801 \> \> \> \> -save transversal\>
14802 \> \> \> \> -max TDO depth 4\>
14803 \> \> \> \> -end\>
14804 \> \> \> -report\>
14805 \> \> \> \> -prefix Edge_curve_q25\>
14806 \> \> \> \> -export flag orbits\>
14807 \> \> \> \> -show TDO\>
14808 \> \> \> \> -show TDA\>
14809 \> \> \> \> -dont show incidence matrices\>
14810 \> \> \> \> -export group\>
14811 \> \> \> \> -end\>
14812 \> \> -end
pdflatex Edge_curve_q25_classification.tex
open Edge_curve_q25_classification.pdf

# 7-backtrack-nodes-total
# group-order 8

# Classification of curves:
# transversal:
# 0: x(0, 0, 0, 3, 4, 3, 5, 15, 22, 6, 1)

# User time: 0:4

# OCN=12 and OCN=41 both have ago=8 and nb_pts=24

# OCN12: 0, 19, 19, 13, 21, 8, 3, 20, 12, 7, 1
# OCN41: 0, 0, 6, 10, 12, 11, 23, 15, 0, 7, 1

Edge_curve_25_nauty_again:

$\text{(ORBITER PATH)orbiter.out}$

-define C -combinatorial_objects:
-define F -finite_field -q 25 -end
-define P -projective_space 2 F -end
-with C -do:
-combinatorial_object_activity:
-canoncal_form PG P:
-classification_prefix Edge_curve_q25:
-label Edge_curve_q25:
-save ago:
-save transversal:
-max TDO depth 4:
-report:
-prefix Edge_curve_q25:
-export_flag_orbits:
-show TDO:
-show TDA:
-dont show incidence_matrices:
-export group:
-end:
#all-three-are-non-isomorphic

```plaintext
Edge_curve_29:

$(\text{ORBITER\ PATH})$orbiter.out-v.2\ 
\define F-\text{finite field}-q.29-end\ 
\define P-\text{projective space}\ 2 F-end\ 
\with P-do-\text{projective space activity}-\define object-Edge-\text{projective variety-Equation}\-end-end

#combinatorial\ object\ create::init-\text{created}\-\text{a set of size}\-114

#(4,15,20,31,32,43,48,59,60,64,83,89,95,110,116,119,127,136,14
4,148,149,172,173,179,189,190,200,207,209,228,230,239,242,253,256
,269,270,283,284,295,299,312,316,329,333,336,340,350,359,368,377,
389,396,413,419,424,430,442,449,452,459,476,479,480,483,495,508,
509,522,536,539,556,564,569,577,589,592,599,602,619,620,629,630,6
42,648,659,665,676,682,683,689,699,704,719,724,732,736,745,749,76
0,762,777,779,788,789,808,809,816,825,830,839,842,844,869)```

```plaintext
Edge_curve_29\ nauty:

$(\text{ORBITER\ PATH})$orbiter.out-v.3\ 
\define C-\text{combinatorial objects}\-\set\ of\ points-\text{Equation}$(\text{EDGE\ CURVE\ Q29 \ AS\ POINTS})\-end\-end```

704
14899 \> \> -define-F-finite_field-q.29.-end\:
14900 \> \> -define-P-projective_space-2:F.-end\:
14901 \> \> -with-C-do\:
14902 \> \> -combinatorial_object_activity\:
14903 \> \> \> -canonical_form_PG-P\:
14904 \> \> \> \> -classification_prefix-q29\:
14905 \> \> \> \> -label.q29\:
14906 \> \> \> \> -save Ago\:
14907 \> \> \> \> -save_transversal\:
14908 \> \> \> \> -max_TDO_depth_4\:
14909 \> \> \> \> -end\:
14910 \> \> \> -report\:
14911 \> \> \> \> -prefix.q29\:
14912 \> \> \> \> -export.flag.orbits\:
14913 \> \> \> \> -show_TDO\:
14914 \> \> \> \> -show_TDA\:
14915 \> \> \> \> -dont_show_incidence_matrices\:
14916 \> \> \> \> -export_group\:
14917 \> \> \> \> -end\:
14918 \> \> \> -end
14919 \> \> pdfflatex.Edge_curve.q29.classification.tex
14920 \> \> open.Edge_curve.q29.classification.pdf
14921
14922 #9{-backtrack-nodes-total
14923 #User-time:0.93-of-a-second,dt=93tps=100
14924
14925 ###########################################################################
14926
14927
14928
14929
14930
14931 Edge_curve.41:
14932 \> 
\$(ORBITER_PATH)orbi\>er.out-v.2:\
14933 \> \> -define-F-finite_field-q41.-end\:
14934 \> \> -define-P-projective_space-2:F.-end\:
14935 \> \> -with-P-do-projective_space_activity.-define_object-Edge.-projective_variet
y.Edge-4 \$(EDGE_CURVE.Q41_EQUATION) -end -end
14936
14937
14938 #combinatorial_object_create: :init-created-a-set-of-size 162
14939 
\#(-4,12,35,43,44,52,75,83,84,101,106,125,138,151,164,167,170,201
,204,209,210,243,244,251,260,275,284,293,307,310,324,335,339,360,
364,371,377,404,410,419,444,461,469,476,484,503,508,519,524,540,
545,564,569,578,587,604,613,628,629,644,645,671,677,678,684,709,7
13,724,728,741,755,764,778,787,797,804,814,837,839,844,846,878,88
705
1, 884, 887, 910, 923, 924, 937, 946, 964, 965, 983, 996, 1004, 1007, 1015, 1044, 1046, 1047, 1049, 1079, 1084, 1091, 1096, 1111, 1124, 1133, 1146, 1155, 1164, 1175, 1184, 1204, 1205, 1216, 1244, 1255, 1259, 1280, 1284, 1301, 1305, 1314, 1324, 1343, 1353, 1364, 1385, 1404, 1414, 1417, 1427, 1444, 1449, 1464, 1469, 1481, 1484, 1511, 1514, 1523, 1524, 1553, 1555, 1564, 1573, 1586, 1595, 1604, 1618, 1623, 1637, 1644, 1650, 1673, 1679, 1682, 1684, 1721

14940
14941
14942
14943
14944
14945
14946
14947  Edge_curve_41.nauty:
14948  $(ORBITER_PATH)orbiter.out--v.3\  
14949  \define-C=\combinatorial_objects\  
14950  \define-F=\finite_field=q41\  
14951  \define-P=\projective_space=2\F\  
14952  \with-C=do\  
14953  \combinatorial_object_activity\  
14954  \canonical_form_PG.P\  
14955  \classification_prefix=Edge_curve_q41\  
14956  \label=Edge_curve_q41\  
14957  \save=\  
14958  \save_transversal\  
14959  \max_TDO_depth=4\  
14960  \report\  
14961  \prefix=Edge_curve_q41\  
14962  \export_flag_orbits\  
14963  \show_TDO\  
14964  \show_TDA\  
14965  \dont_show_incedence_matrices\  
14966  \export_group\  
14967  \end\  
14968  pdflatex-Edge_curve_q41_classification.tex
14969  open-Edge_curve_q41_classification.pdf
14970  end
14971
14972  #9-backtrack-nodes-total
14973  #User-time:0:4
14974
14975  #===============================================
14976  #===============================================
14977
14978
14979
14980
14981
14982
14983
14984  Edge_curve_61:
14985  > $(ORBITER_PATH)orbiter.out -v.2 \n14986  > -define-F-finite_field-q.61-end \n14987  > -define-P-projective_space-2-F-end \n14988  > -with-P-do-projective_space_activity-defne_object-Edge-projective_variet y-Edge-4-$(EDGE_CURVE_Q61_EQUATION)-end-end
14989
14990
14991
14992  #combinatorial_object.create::init-created-a-set-of-size-242
14994
14995
14996
14997
14998
14999
15000
15001
15002
15003  Edge_curve_61_nauty:
15004  > $(ORBITER_PATH)orbiter.out -v.3 \n15005  > -define-C-combinatorial_objects \n15006  > -set_of_points-$(EDGE_CURVE_Q61_AS_POINTS) \n15007  > -end \n15008  > -define-F-finite_field-q.61-end \n
707
-define P-projective_space:2::F:-end\ 
-with C:-do\ 
-combinatorial_object_activity\ 
-canonical_form PG P\ 
-classification_prefix Edge_curve_q61\ 
-label Edge_curve_q61\ 
-save ago\ 
-save_transversal\ 
-max TDO:depth:4\ 
-end\ 
-report\ 
-prefix Edge_curve_q61\ 
-export_flag_orbits\ 
-show TDO\ 
-show TDA\ 
-dont_show_incidence_matrices\ 
-export_group\ 
-end\ 
-pdflatex Edge_curve_q61_classification.tex
-open Edge_curve_q61_classification.pdf

#-9-backtrack-nodes:total
#ago=24
#User time:0:7

Edge_curve_81:
$ (ORBITER_PATH) orbiter.out --v:2\ 
-define F-finite_field:q:81:-end\ 
-define P-projective_space:2::F:-end\ 
-with P-do-projective_space_activity--define_object Edge--projective_variety Edge-4 $(EDGE_CURVE_Q81_EQUATION)--end-end
15054
15055
15056  \#combinatorial_object_create::init-created-a-set-of-size-322
15057  \#(-4,5,50,67,-84,-85,130,-147,164,210,227,244,245,246,330,332,364,4
03,409,412,423,428,491,495,521,546,572,576,585,590,652,655,688,72
7,735,737,764,789,822,823,830,835,901,910,924,953,983,993,1017,10
49,1068,-1079,-1091,-1128,-1147,-1159,-1162,-1189,-1226,-1229,-1236,-1242,-1314,
-1319,-1346,-1368,-1383,-1385,-1393,-1399,-1475,-1482,1510,1525,1550,1560,15
72,1601,-1630,-1633,-1639,-1645,-1713,-1721,-1746,-1778,-1788,-1790,-1799,-1806,
31,2144,-2181,-2198,-2202,-2208,-2213,-2298,2300,2324,2331,2377,2392,-2404,
-2425,-2445,2456,2459,2487,2540,2544,2568,2573,2604,2612,2623,2653,27
05,2710,2736,2746,2758,2761,2790,2813,2869,2889,2893,2903,2951,2967,
-2976,2987,3011,3024,3036,3071,-3115,-3122,-3151,-3156,-3190,-3197,3220,32
34,3282,3289,3310,3320,3361,3374,3390,3400,3407,3411,3443,3483,3522,
-3528,3557,3560,3574,3583,3607,3640,3675,3689,3701,3723,3772,3774,-37
90,3791,3809,3810,3853,3871,3935,3954,3986,3993,4020,4040,-4048,-4099,
-4108,4120,-4172,4181,4203,4212,4254,-4266,-4280,4292,4329,4345,4352,43
60,4390,4396,-4427,-4443,-4484,-4498,-4512,-4531,-4564,4568,-4591,-4599,-4632,
-4643,4647,-4673,-4733,-4752,-4758,-4766,-4783,-4786,-4813,4837,4896,4911,49
19,4931,-4953,-4961,-4973,5004,5051,5053,5079,-5083,-5136,-5140,-5165,-5176,
-5186,5187,5231,5250,5311,5329,-5393,-5394,5411,5412,5463,5480,5486,54
96,5550,5560,5575,-5588,-5605,-5611,-5643,5657,-5704,-5720,-5728,-5742,-5768,
-5775,-5800,-5821,-5873,-5883,-5903,-5913,-5951,-5957,-5988,-5991,6008,6021,60
31,-6067,-6106,-6114,-6136,-6149,-6166,-6175,6200,-6234,-6276,-6280,-6311,-6313,
-6348,-6363,6374,6395,-6440,-6452,-6468,-6480,-6520,-6529,-6550,6559,6566,-65
70,-6603,-6641.)

15058
15059
15060
15061
15062
15063
15064  \#142-backtrack-nodes.total
15065  \#User-time::1:34
15066  \#go=.96
15067
15068
15069
15070
15071
15072
15073
15074
15075
15076
15077  ####################################################################################################################


### Section 13.1: Graphical Output

**SECTION_GRAPHICAL_OUTPUT:**

```
> F_7_tables:
> $(ORBITER_PATH)orbiter.out -v 3:
> -define F -finite_field -q 7 -end:
> -with F -do F -finite_field_activity:
> -cheat_sheet_GF:
> -end
> $(ORBITER_PATH)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:
> $(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:
> 1:0,1,0,0,0,1,2,1,1,0" PG_2_4_singer":
> -end
> $(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
```
\begin{verbatim}
15122
15123
15124
15125
15126 PGL_4.2_Wedge_4.0_graphical_output:
15127 ▶ $(ORBITER_PATH)oritzer.out--v.4\n
15128 ▶ ▶ -define-G:linear_group=PGL.4.2\n15129 ▶ ▶ ▶ -wedge_detached\n
15130 ▶ ▶ -end\n15131 ▶ ▶ -with-G:do\n15132 ▶ ▶ -group_theoretic_activity\n15133 ▶ ▶ ▶ -report\n
15134 ▶ ▶ -end
15135 ▶ pdflatex PGL_4.2_Wedge_4.2_detached_report.tex
15136 ▶ open-PGL_4.2_Wedge_4.2_detached_report.pdf

15137
15138 #▶ ▶ ▶ -draw_options:-radius.200:-end\n
15139
15140 schreier_tree_graphical_output:
15141 ▶ $(ORBITER_PATH)oritzer.out--v.4\n15142 ▶ ▶ -draw_options:\n15143 ▶ ▶ ▶ -yout.500000\n15144 ▶ ▶ ▶ -radius.15:-nodes_empty\n15145 ▶ ▶ ▶ ▶ -line_width.0.5:-y_stretch.0.25\n
15146 ▶ ▶ -end\n15147 ▶ ▶ -define-G:linear_group=PGL.4.2:-end\n15148 ▶ ▶ -with-G:do\n
15149 ▶ ▶ -group_theoretic_activity\n15150 ▶ ▶ ▶ -orbits_on_polynomials.3\n
15151 ▶ ▶ ▶ ▶ -orbits_on_polynomials_draw_tree.6\n15152 ▶ ▶ ▶ -end\n
15153 ▶ pdflatex poly_orbits_d3.n3.q2.tex
15154 ▶ open-poly_orbits_d3.n3.q2.pdf

15155
15156
15157 Queens_graph:
15158 ▶ $(ORBITER_PATH)oritzer.out--v.2\n
15159 ▶ ▶ -define-G:graph:non_attacking_queens_graph.8:-end\n
15160 ▶ ▶ -with-G:do\n15161 ▶ ▶ -group_theoretic_activity:-export_csv:-end\n
15162 ▶ ▶ -with-G:do\n
15163 ▶ ▶ -group_theoretic_activity:-export_graphviz:-end\n
15164 ▶ ▶ -with-G:do\n15165 ▶ ▶ -group_theoretic_activity:-save:-end\n
15166 ▶ ▶ -with-G:do\n
15167 ▶ ▶ -group_theoretic_activity:-automorphism_group:-end\n
15168 ▶ ▶ -with-G:do\n
15170

711
\end{verbatim}
-graph_theoretic_activity -find_cliques

-target_size 8 -output_file non_attacking_queens.graph 8.csv

#$\{\text{ORBITER\_PATH}\}\texttt{orbiter.out} -v 2 -draw_matrix -input_csv_file non_attacking_queens.graph 8.csv

#$ -box_width 8 -bit_depth 24 -partition 4 64 64

twopi -Tpng non_attacking_queens.graph 8.gv > non_attacking_queens.graph 8.png

#pdflatex non_attacking_queens.graph 8.report.tex

#open non_attacking_queens.graph 8.report.pdf

#$(\text{ORBITER\_PATH})\texttt{orbiter.out} -v 2 -povray

-round 0 -nb_fraction_default 30

-output_mask_cube %d %03d.pov

-video_options -W 1024 -H 768

-global_picture_scale 0.5

-default_angle 75

-clipping_radius 2.7

-end

-scene_objects

-obj_file cube_centered.obj

-edge "0,1"

-edge "0,2"

-edge "0,4"

-edge "1,3"

-edge "1,5"

-edge "2,3"

-edge "2,6"

-edge "3,7"

-edge "4,5"

-edge "4,6"

-edge "5,7"

-edge "6,7"

-group_of_things_as_interval 0.8

-spheres 0.0.3

...
15214 "texture{Polished_Chrome-pigment{quick_color-White}}"
15215 d d d -group_of_things_as_interval-0.6-
15216 d d d -prisms-1.0.05-
15217 "texture{pigment{color-Yellow-transmit-0.7}}"
15218 finish{diffuse-0.9-phong-0.6}".
15219 d d d -group_of_things_as_interval-0.12-
15220 d d d -cylinders-2.0.15-
15221 "texture{pigment{color-Yellow-transmit-0.7}}"
15222 d d -scene_objects_end-
15223 d -povray_end
15224 -rm.-rf-POV
15225 mkdir-POV
15226 mv-cube_0.*.pov-POV
15227 mv-makefile_animation-POV
15228
15229
15230
15231
15232 monkey:
15233 d $(ORBITER_PATH)orbiter.out-v.2-povray-
15234 d d -round-0--nb_frames_default-30-
15235 d d -output_mask-monkey_%d_03d.pov-
15236 d d -video_options-W-1024-H-768-
15237 d d -global_picture_scale-0.8-
15238 d d -default_angle-75-
15239 d d -clipping_radius-0.8-
15240 d d -camera-0."0,0,1"."1,1,0.5"."0,0,0"..
15241 d d -rotate_about_z_axis-
15242 d d -end-
15243 d d -scene_objects-
15244 d d d -cubic_lex-$(MONKEY_SADDLE_CUBIC)-
15245 d d d -plane_by_dual_coordinates."0,0,1,0"-
15246 d d d -group_of_things."0"-
15247 d d d -group_of_things."0"-
15248 d d d -cubics-0:"texture{pigment{Gold}}-finish-
15249 {ambient-0.4-diffuse-0.5-roughness-0.001-
15250 reflection-0.1-specular-.8}"
15251 d d d -planes-1:"texture{pigment{color-Blue}-
15252 transmit-0.5}-finish{diffuse-0.9-phong-0.2}"
15253 d d -scene_objects_end-
15254 d -povray_end
15255 -rm.-rf-POV
15256 mkdir-POV
15257 mv-monkey_0.*.pov-POV
15258 mv-makefile_animation-POV
15259
15260
713
15261
15262
15263
15264 Eckardt:
15265 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15266 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15267 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15268 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15269 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15270 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15271 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15272 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15273 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15274 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15275 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15276 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15277 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15278 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15279 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15280 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15281 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15282 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15283 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15284 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15285 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15286 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15287 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15288 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15289 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15290 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15291 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15292 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15293 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15294 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15295 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15296 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15297 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15298 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15299 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15300 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15301 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15302 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15303 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15304 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15305 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15306 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)
15307 \( \text{\$\{ORBITER\_PATH\}orbiter.out-v.2.-povray\} \)

714
-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
-group_of_things_as_interval.6-6
-texts-4:0.2:0.15."texture{pigment{Black}}.no_shadow"
-texts-5:0.2:0.15."texture{pigment{Black}}.no_shadow"
-texts-6:0.2:0.15."texture{pigment{Black}}.no_shadow"
-scene_objects_end
-povray_end
-rem-rf-POV
mkdir-POV
mv-Eckardt_0.*.pov-POV
mv-makefile_animation-POV
Eckardt
dehorm:
$(ORBITER_PATH)orbiter.out-v.2-povray
-round-0-nb_frames_default.93
-output_mask-Eckardt_deform_%d_03d.pov
-video_options.-W.1024.-H.768
-global_picture_scale.0.9
-default_angle.75
-clipping_radius.2.4
-camera.0."1,1,1"."-3,1,3"."0.12,0.12,0.12".
-end
-scene_objects
-Hilbert_Cohn_Vossen_surface
-group_of_things.0
-deformation_of_cubic_lex.93.1.107148718.1.570796327.0

Eckardt_deform_2:

$({ORBITER_PATH}/orbiter.out-v.2.-povray-

-round-0-nb_frames_default-30-

-output_mask-Eckardt_deform_%d_%03d.pov-

-video_options-W:1024-H:768-

-global_picture_scale:0.9-

-default_angle:75-

-clipping_radius:2.4-

camera-0."1,1,1"."-3,1,3"."0.12,0.12,0.12".-

-end-

-scene_objects-

-Hilbert_Cohn_Vossen_surface-

-group_of_things."0"

deformation_of_cubic_lex:93.1.107148718-1.570796327-0-

-cubics-1"texture{pigment{White*0.5-transmit:0.5}}-

finish{ambient-0.4-diffuse-0.5-roughness:0.001-reflection:0.1-specular:.8}"

-cubics-2"texture{pigment{Red*0.5-transmit:0.5}}-

finish{ambient-0.4-diffuse-0.5-roughness:0.001-reflection:0.1-specular:.8}"

-cubics-3"texture{pigment{Blue*0.5-transmit:0.5}}-

finish{ambient-0.4-diffuse-0.5-roughness:0.001-reflection:0.1-specular:.8}"

-scene_objects_end-

-povray_end

-rm.-rf-POV

mkdir-POV

mv-Eckardt_deform_0.*.pov-POV

mv-makefile_animation-POV

Eckardt_deform_2:
15401  mkdir-POV
15402  mv-Eckardt_deform_0_*.pov-POV
15403  mv-makefile_animation-POV
15404
15405
15406
15407
15408  
15409
15410
15411  Clebsch:
15412  $(ORBITER_PATH)orbiter.out-v.2-povray-
15413  -round-0--mbFrames_default30-
15414  -output_mask:Clebsch_%d_03d.pov-
15415  -video_options:-W-1024--H-768-
15416  -global_picture_scale-0.9-
15417  -default_angle-80-
15418  -clipping_radius-2.4-
15419  -camera-0."1,1,1"."-4.5,3.5,6"."0,0,0"-
15420  -end-
15421  -scene_objects-
15422  - Clebsch_surface-
15423  -group_of_things-"0"-
15424  -cubics-0."texture{pigment{White*0.5}.finish-
15425  {ambient-0.4-diffuse-0.5-roughness-0.001-reflection-0.1-specular-0.8}}-
15426  -group_of_things_as_interval-0-6-
15427  -group_of_things_as_interval-6-6-
15428  -group_of_things_as_interval-12-15-
15429  -lines-1-0.02."texture{pigment{color:Red}.finish-
15430  {diffuse-0.9-phong-1}}-
15431  -lines-2-0.02."texture{pigment{color:Blue}.finish-
15432  {diffuse-0.9-phong-1}}-
15433  -lines-3-0.02."texture{pigment{color:Yellow}.finish-
15434  {diffuse-0.9-phong-1}}-
15435  -group_of_things_as_interval-0-12-
15436  -spheres-4-0.08."texture{pigment{Cyan*1.3}}-
15437  -finish{ambient-0.4-diffuse-0.6-roughness-0.001-reflection-0-specular-.8}-
15438  -scene_objects_end-
15439  -povray_end
15440  -rm-rf-POV
15441  -mkdir-POV
15442  mv-Clebsch_0_*.pov-POV
15443  mv-makefile_animation-POV
15444
15445
15446
15447
endrass8:
$(ORBITER_PATH)orbiter.out--v.2--povray--
-round0--nb_frames_default.30--
-output_mask-endrass_octic.%d%03d.pov--
-video_options--W1024--H768--
-global_picture_scale.75--
default_angle.75--
-clipping_radius.3.7--
-no_bottom_plane--
camera.0"1,1,1".6,6,3"0,0,0"--
-rotate_about.111--
-end--
-scene_objects--
-line_through_two_points.recentered_from.csv_file--
-coordinate_grid.csv--
-group_of_things."0"--
-group_of_things."1"--
-group_of_things."2"--
-group_of_things_as_interval.3.39--
-lines.0.0.15"texture{pigment{color.Orange}}"--
-diffuse.0.9-phonl."--
-lines.1.0.15"texture{pigment{color.Green}}"--
-diffuse.0.9-phonl."--
-lines.2.0.15"texture{pigment{color.Blue}}"--
-diffuse.0.9-phonl."--
-lines.3.0.05"texture{pigment{color.Black}}"--
-diffuse.0.9-phonl."--
octic_lex.165$(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--"--
-planets.5"texture{pigment{color.Blue-transmit.0.5.--}}--
-diffuse.0.9-phonl."--
-scene_objects_end--
povray_end--
-rm-rf-POV--
mdir-POV--
mv-endrass.octic.0.*.pov-POV--
mv-makefile_animation-POV--


Section 13.3: Creating Animations

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:
- $(ORBITER PATH)orbiter.out -v:2 -povray
  -round 0 -nb_frames_default 30
  -output_mask Eckardt_%d%03d.pov
  -video_options -W 1024 -H 768
  -global_picture_scale 0.9
  -default_angle 75
  -clipping_radius 2.4
  -camera "1,1,1", "-3,1,3", "0.12,0.12,0.12"
  -end
  -scene_objects
  -cubic Goursat "6,3,-15"
  -group_of_things "0"
15541 ❮ ❮ ❮ -cubics-0:"texture{pigment{\n15542 White*0.5-transmit 0.5};finish{ambient 0.4\n15543 diffuse 0.5-roughness 0.001-reflection 0.1\n15544 specular .8}};\n15545 ❮ ❮ ❮ -scene_objects_end\n15546 ❮ ❮ ❮ -povray_end
15547 ❮ ❮ ❮ -rm -rf POV
15548 ❮ ❮ ❮ mkdir POV
15549 ❮ ❮ ❮ mv Eckardt_0_* .pov POV
15550 ❮ ❮ ❮ mv -makefile_animation-POV
15551
15552
15553
15554 Kummer_surface:
15555 ❮ ❮ -$(ORBITER_PATH) orbiter.out -v 2 -povray\n15556 ❮ ❮ -round 0 -nb_frames_default 30\n15557 ❮ ❮ -output_mask Kummer_%d_%03d .pov\n15558 ❮ ❮ -video_options -W 1024 -H 768\n15559 ❮ ❮ -global_picture_scale 0.9\n15560 ❮ ❮ -default_angle 75\n15561 ❮ ❮ -clipping_radius 2.4\n15562 ❮ ❮ -camera 0 "1,1,1" "-3,1,3" "0.12,0.12,0.12"..\n15563 ❮ ❮ -end\n15564 ❮ ❮ -scene_objects\n15565 ❮ ❮ ❮ -quartic_lex 35 $(KUMMER_QUARTIC_LEX 35)\n15566 ❮ ❮ ❮ -group_of_things "0"\n15567 ❮ ❮ ❮ -quartics 0:"texture{pigment{White*0.5-transmit\n15568 0.5};finish{ambient 0.4-diffuse 0.5-roughness 0.001\n15569 reflection 0.1-specular .8}};\n15570 ❮ ❮ ❮ -scene_objects_end\n15571 ❮ ❮ ❮ -povray_end
15572 ❮ ❮ -rm -rf POV
15573 ❮ ❮ mkdir POV
15574 ❮ ❮ mv Kummer_0_* .pov POV
15575 ❮ ❮ mv -makefile_animation-POV
15576
15577
15578 # Maple:
15579 # Kummer := expand((x0^2 + x1^2 + x2^2 + x3^2)^2 - 3*(x0^4 + x1^4 + x2^4 + x3^4))
15580
15581
15582 Kummer_video:
15583 ❮ ❮ -rm -r FRAMES
15584 ❮ ❮ -mkdir FRAMES
15585 ❮ ❮ -rm Kummer.mp4
15586 ❮ ❮ $(ORBITER_PATH) orbiter.out -v 2 -povray\n15587 ❮ ❮ -prepare_frames\n
720
Beauville_surface:

$(\text{BEAUVILLE_QUINTIC LEX}_56)\cdot(3+\sqrt{5})/2$

# 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.

A = 2.618033988
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, 0.4472135957999158, 0.4472135957999158, 0.4472136021531272], [-0.4472135957999158, -4.2360680044124255, -4.2360680044124255, -4.2360680044124255], [4.2360680044124255, -4.2360680044124255, -4.2360680044124255, -4.2360680044124255]])

#-0.44721360215312733, 1.1708204000530853, 1.1708204000530853, -0.4472135957999158
#-1.1708204000530853, 0.4472135957999158, 0.4472135957999158, 0.4472136021531272
#4.2360680044124255, -4.2360680044124255, -4.2360680044124255, -4.2360680044124255
#1.6180340022062127, -2.6180340022062127, -1.6180340022062127, -1.6180340022062127, 0]

T00=-0.44721360215312733
T01=1.1708204000530853
T02=1.1708204000530853
T03=-0.4472135957999158
T10=-1.1708204000530853
T11=0.4472136021531272
T12=1.4472136021531272
T13=0.4472135957999158
T20=4.2360680044124255
T21=-4.2360680044124255
T22=-4.2360680044124255
T23=0.
T30=1.6180340022062127
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-b c push-d push-d push-m mult add mult
push-a push-d push-d push-m add mult add add
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
push-d push-a push-1 push-m mult add mult
push-m push-a mult add push-c add
push-c push-m push-a mult add
mult mult
15678 ▶▶▶▶ store.c002:\n
15679 ▶▶▶▶ push.b:\n
15680 ▶▶▶▶ push.d-push.c-push.a-push.m-mult.add.mult:\n
15681 ▶▶▶▶ push.c-push.a-push.m-push.1-mult.add.mult.add.mult:\n
15682 ▶▶▶▶ push.a-push.d-mult.push.c-push.1-push.m-mult.add.mult:\n
15683 ▶▶▶▶ push.m-mult.add\n
15684 ▶▶▶▶ push.a-push.c-push.m-mult.add.mult:\n
15685 ▶▶▶▶ store.c011:\n
15686 ▶▶▶▶ push.b-push.b-push.c-mult.mult:\n
15687 ▶▶▶▶ push.1-push.d-push.m-mult.add.mult:\n
15688 ▶▶▶▶ push.a-push.b-mult.push.c-push.d-push.d-push.m-mult.mult.add.mult:\n
15689 ▶▶▶▶ push.m-mult.add\n
15690 ▶▶▶▶ push.a-push.d-mult.push.c-push.d-push.m-mult.add.mult.add.mult:\n
15691 ▶▶▶▶ push.a-push.c-push.m-mult.add.mult:\n
15692 ▶▶▶▶ store.c012:\n
15693 ▶▶▶▶ push.m\n
15694 ▶▶▶▶ push.b-push.d-push.m-mult.add.push.c.mult:\n
15695 ▶▶▶▶ push.d-push.b-push.1-push.m-mult.add.mult.push.m-mult.add.push.a.mult:\n
15696 ▶▶▶▶ push.b-push.c-mult.push.d-push.1-push.m-mult.add.mult.add.mult:\n
15697 ▶▶▶▶ push.b-push.d-push.m-mult.add.mult:\n
15698 ▶▶▶▶ store.d001:\n
15699 ▶▶▶▶ push.m\n
15700 ▶▶▶▶ push.d-push.c-push.m-mult.add.push.a-push.a.multmult:\n
15701 ▶▶▶▶ push.c-push.c-mult.push.d-push.m-mult.add.push.a-mult.add:\n
15702 ▶▶▶▶ push.m-push.b-push.c-mult.mult.push.c-push.1-push.m-mult.add.mult.add.mult.add.mult:\n
15703 ▶▶▶▶ push.b-push.d-push.m-mult.add.mult:\n
15704 ▶▶▶▶ store.d011:\n
15705 ▶▶▶▶ push.m\n
15706 ▶▶▶▶ push.c-push.d-mult.push.d-push.m-mult.add.push.a-push.a.mult.mult:\n
15707 ▶▶▶▶ push.c-push.c-mult.push.d-push.m-mult.add.push.a-push.b-push.m-mult.mult.mult.add.mult:\n
15708 ▶▶▶▶ push.b-push.d-push.c-push.m-mult.add.mult.push.c-push.m-mult.mult.add\n
15709 ▶▶▶▶ push.b-push.d-push.m-mult.add.mult.mult:\n
15710 ▶▶▶▶ store.d012:\n
15711 ▶▶▶▶ push.d-push.1-push.m-mult.add.push.a-mult.push.m-push.b-mult.push.1.add.push.c-mult.add:\n
15712 ▶▶▶▶ push.b-add.push.m-push.d-mult.add\n
15713 ▶▶▶▶ push.a-push.c-mult.mult\n
15714 ▶▶▶▶ push.b-push.d-push.m-mult.add.mult:\n
15715 ▶▶▶▶ store.d112:\n
15716 ▶▶▶▶ push.m\n
15717 ▶▶▶▶ push.b-push.d-push.m-mult.add.push.c-mult.push.d-push.b-push.1-push.m-mult.mult.add.mult.add.mult:\n
15718 ▶▶▶▶ push.m-mult.add.push.a-mult.push.b-push.c-mult.push.d-push.1.push.m-mult.add.mult.addmult:\n
15719 ▶▶▶▶ push.b-push.d-push.m-mult.add.mult.mult:\n
723
15720 ▶ ▶ ▶ ▶ store m002
15721 ▶ ▶ ▶ ▶ push m
15722 ▶ ▶ ▶ ▶ push d push c push m mult add push a push a mult mult
15723 ▶ ▶ ▶ ▶ push c push c mult push d push m mult add push a mult add
15724 ▶ ▶ ▶ ▶ push b push c push m mult push c push 1 push m mult add mult add
15725 ▶ ▶ ▶ ▶ push b push d push m mult add mult mult
15726 ▶ ▶ ▶ ▶ store m012
15727 ▶ ▶ ▶ ▶ push m
15728 ▶ ▶ ▶ ▶ push c push d mult push d push m mult add push a push a mult mult
15729 ▶ ▶ ▶ ▶ push m push c push c mult push d push m mult add push a push b mult mult mult add
15730 ▶ ▶ ▶ ▶ push m push b push d push c push m mult add push c mult mult mult add
15731 ▶ ▶ ▶ ▶ push b push d push m mult add mult mult
15732 ▶ ▶ ▶ ▶ store m022
15733 ▶ ▶ ▶ ▶ push d push 1 push m mult add push a mult
15734 ▶ ▶ ▶ ▶ push m push b mult push 1 add push c mult add
15735 ▶ ▶ ▶ ▶ push b add push m push d mult add
15736 ▶ ▶ ▶ ▶ push a push c mult mult
15737 ▶ ▶ ▶ ▶ push b push d push m mult add mult mult
15738 ▶ ▶ ▶ ▶ store m122
15739 ▶ ▶ ▶ ▶ push m push a mult push c add push d mult push c push a push 1 push m mult add mult add
15740 ▶ ▶ ▶ ▶ push b mult
15741 ▶ ▶ ▶ ▶ push m push a push d mult mult push c push 1 push m mult add mult add
15742 ▶ ▶ ▶ ▶ push b push d push m mult add mult mult
15743 ▶ ▶ ▶ ▶ store m002
15744 ▶ ▶ ▶ ▶ push m
15745 ▶ ▶ ▶ ▶ push c push d push m mult add push b mult push m push d push c push 1 push m mult mult add mult mult add
15746 ▶ ▶ ▶ ▶ push b push c mult push d push 1 push m mult add mult add mult mult
15747 ▶ ▶ ▶ ▶ push a push b push c push m mult push d push m mult add mult add add mult
15748 ▶ ▶ ▶ ▶ push a push b push c push m mult push d push m mult add mult add
15749 ▶ ▶ ▶ ▶ store m012
15750 ▶ ▶ ▶ ▶ push c push d push m mult add push b mult
15751 ▶ ▶ ▶ ▶ push m push d push c push 1 push m mult add mult mult add
15752 ▶ ▶ ▶ ▶ push a mult
15753 ▶ ▶ ▶ ▶ push b push c push m mult push 1 push m mult add mult add
15754 ▶ ▶ ▶ ▶ push a push d mult push m push b push c mult mult add mult mult
15755 ▶ ▶ ▶ ▶ store m022
15756 ▶ ▶ ▶ ▶ push m
15757 ▶ ▶ ▶ ▶ push c push d push m mult add push b mult
15758 ▶ ▶ ▶ ▶ push m push d push c push 1 push m mult add mult mult add
15759 ▶ ▶ ▶ ▶ push a mult
15760 ▶ ▶ ▶ ▶ push b push c push d push 1 push m mult add mult add
15761 ▶ ▶ ▶ ▶ push m push a mult push c add mult mult
15762 ▶ ▶ ▶ ▶ store m112
15763 ▶ ▶ ▶ ▶ push m
Clebsch_up_create_points:
> $(ORBITER_PATH)orbiter.out -v 2 \n> \n> -smooth_curve "Clebsch_map_of_circle_to_defining_eqn_r2" \n> \n> 0.07:1000:5.0:$\text{TW}_0:PI: \n> \n> -const: a$\text{CLEBSCH}_A$-b$\text{CLEBSCH}_B$-c$\text{CLEBSCH}_C$-d$\text{CLEBSCH}_D$: \n> \n> t00:$\text{t00}$-t01:$\text{t01}$-t02:$\text{t02}$-t03:$\text{t03}$: \n> \n> t10:$\text{t10}$-t11:$\text{t11}$-t12:$\text{t12}$-t13:$\text{t13}$: \n> \n> t20:$\text{t20}$-t21:$\text{t21}$-t22:$\text{t22}$-t23:$\text{t23}$: \n> \n> t30:$\text{t30}$-t31:$\text{t31}$-t32:$\text{t32}$-t33:$\text{t33}$: \n> \n> r:2.0:one:1-m:-1: \n> \n> -const_end: \n> \n> -var.t: \n> \n> c001-c002-c011-c012: \n> \n> d001-d011-d012-d112: \n> \n> m002-m012-m022-m122: \n> \n> n002-n012-n112-n022-n122: \n> \n> y0-y1-y2: \n> \n> y001-y002-y011-y012-y022-y112-y122: \n> \n> x0-x1-x2-x3: \n> \n> -code: \n> \n> push\_t:cos\_t:push\_r:mult\_store:y0: \n> \n> push\_t:sin\_t:push\_r:mult\_store:y1: \n> \n> push\_one\_store:y2: \n> \n> push\_y0:push\_y0:push\_y1:mult\_store:y001: \n> \n> push\_y0:push\_y0:push\_y2:mult\_store:y002: \n> \n> push\_y0:push\_y1:push\_y1:mult\_store:y011: \n> \n> push\_y0:push\_y1:push\_y2:mult\_store:y012: \n> \n> push\_y0:push\_y2:push\_y2:mult\_store:y022: \n> \n> push\_y1:push\_y1:push\_y2:mult\_store:y112: \n> \n> push\_y1:push\_y2:push\_y2:mult\_store:y122: \n> \n> \$(\text{CLEBSCH}\_CUBICS): \n> \n> push\_c001:push\_y001:mult: \n> \n> push\_c002:push\_y002:mult\_add: \n> \n> push\_c011:push\_y011:mult\_add: \n> \n> push\_c012:push\_y012:mult\_add: \n> \n> store:x0: \n> \n> push\_d001:push\_y001:mult: \n> \n> push\_d011:push\_y011:mult\_add: \n> \n> push\_d012:push\_y012:mult\_add:
15811 ▶ ▶ ▶ ▶ push-d112-push-y112-mult-add\n15812 ▶ ▶ ▶ ▶ store-x1.\n15813 ▶ ▶ ▶ ▶ ▶ push-m002-push-y002-mult-\n15814 ▶ ▶ ▶ ▶ ▶ push-m012-push-y012-mult-add\n15815 ▶ ▶ ▶ ▶ ▶ push-m022-push-y022-mult-add\n15816 ▶ ▶ ▶ ▶ ▶ push-m122-push-y122-mult-add\n15817 ▶ ▶ ▶ ▶ ▶ store-x2.\n15818 ▶ ▶ ▶ ▶ ▶ push-n002-push-y002-mult-\n15819 ▶ ▶ ▶ ▶ ▶ push-n012-push-y012-mult-add-\n15820 ▶ ▶ ▶ ▶ ▶ push-n022-push-y022-mult-add-\n15821 ▶ ▶ ▶ ▶ ▶ push-n112-push-y112-mult-add-\n15822 ▶ ▶ ▶ ▶ ▶ push-n122-push-y122-mult-add-\n15823 ▶ ▶ ▶ ▶ ▶ store-x3.\n15824 ▶ ▶ ▶ ▶ ▶ push-x0-push-t00-mult-\n15825 ▶ ▶ ▶ ▶ ▶ push-x1-push-t10-mult-add-\n15826 ▶ ▶ ▶ ▶ ▶ push-x2-push-t20-mult-add-\n15827 ▶ ▶ ▶ ▶ ▶ push-x3-push-t30-mult-add-\n15828 ▶ ▶ ▶ ▶ ▶ return.\n15829 ▶ ▶ ▶ ▶ ▶ push-x0-push-t01-mult-\n15830 ▶ ▶ ▶ ▶ ▶ push-x1-push-t11-mult-add-\n15831 ▶ ▶ ▶ ▶ ▶ push-x2-push-t21-mult-add-\n15832 ▶ ▶ ▶ ▶ ▶ push-x3-push-t31-mult-add-\n15833 ▶ ▶ ▶ ▶ ▶ return.\n15834 ▶ ▶ ▶ ▶ ▶ push-x0-push-t02-mult-\n15835 ▶ ▶ ▶ ▶ ▶ push-x1-push-t12-mult-add-\n15836 ▶ ▶ ▶ ▶ ▶ push-x2-push-t22-mult-add-\n15837 ▶ ▶ ▶ ▶ ▶ push-x3-push-t32-mult-add-\n15838 ▶ ▶ ▶ ▶ ▶ return.\n15839 ▶ ▶ ▶ ▶ ▶ push-x0-push-t03-mult-\n15840 ▶ ▶ ▶ ▶ ▶ push-x1-push-t13-mult-add-\n15841 ▶ ▶ ▶ ▶ ▶ push-x2-push-t23-mult-add-\n15842 ▶ ▶ ▶ ▶ ▶ push-x3-push-t33-mult-add-\n15843 ▶ ▶ ▶ ▶ ▶ return.\n15844 ▶ ▶ -code_end
15845
15846
15847 Clebsch_surface:
15848 ▶ $(ORBITER_PATH)orbiter.out.-v.2.-povray-\n15849 ▶ ▶ -round=0.-nb_frames_default=30-\n15850 ▶ ▶ -output_mask=Clebsch_\%d_\%03d.pov-\n15851 ▶ ▶ -video_options=-W=1024--H=768-\n15852 ▶ ▶ -global_picture_scale=0.9-\n15853 ▶ ▶ -default_angle=75-\n15854 ▶ ▶ -clipping_radius=2.4-\n15855 ▶ ▶ -camera=0.1,1,1"."-3,1,3"."0.12,0.12,0.12"..-\n15856 ▶ ▶ -end-\n15857 ▶ ▶ -scene_objects-\n
726
-cubic_orbiter:"0,0,0,0,-4.236067972,\n0,0,4.236067972,4.236067972,17.94427188,\n-17.94427188,0,0,-9.472135941,0,0,5.236067971,\n8.472135938,-27.41640782"\n-group_of_things "0"
-cubics:0:"texture{pigment{White*0.5-transmit:0.5.-}"\nfinish-{ambient:0.4-diffuse:0.5.roughness:0.001-}
reflection:0.1-specular:.8}"
-point_list_from_csv_file\n-function_Clebsch_map_of_circle_N1000_points.csv\n-group_of_things_as_interval:0:954
-spheres:1.0.07:"texture{pigment{color:Red.-}"
finish-{diffuse:0.9-phong:1}"
-scene_objects_end\npovray_end
-rm:rf-POV
mkdir-POV
mv:Clebsch_0.*.pov-POV
mv:makefile_animation-POV

Clebsch_surface_defining_equation:
$\text{\textcolor{black}{(ORBITER\ PATH)\ orbiter.out}}\ -v\ -2\ -povray\$
-round:0:-nb_frames_default:30-
-output_mask:Clebsch_%d_%03d.pov-
-video_options:\-W1024\-H768-
-global_picture_scale:0.6-
default_angle:75-
-clipping_radius:1.6-
camera:0.1,1,1\-2,0,2.0,0,0-
-scene_objects-
-end-
scene_objects-
cubic_orbiter:0,0,0,1,1,1,1,1,1,1,1,1,2,2,2-
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:.8}"
-scene_objects_end-
povray_end
-rm:rf-POV
mkdir-POV
mv:Clebsch_0.*.pov-POV
mv:makefile_animation-POV

Clebsch_surface_defining_equation_and_curves:
$\text{\textcolor{black}{(ORBITER\ PATH)\ orbiter.out}}\ -v\ -2\ -povray\$
15905  ▷ ▷ -round-0.-nb_frames_default-30\ 
15906  ▷ ▷ -output_mask-Clebsch_2curves_%d_%03d.pov\ 
15907  ▷ ▷ -video_options-3.1024--H-768\ 
15908  ▷ ▷ -global_picture_scale-0.6\ 
15909  ▷ ▷ -default_angle-75\ 
15910  ▷ ▷ -clipping_radius-1.6\ 
15911  ▷ ▷ -camera-0."1,1,1"."-2,0,2"."0,0,0".\ 
15912  ▷ ▷ -end\ 
15913  ▷ ▷ -scene_objects\ 
15914  ▷ ▷ ▷ -cubic_orbiter."0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2".\ 
15915  ▷ ▷ ▷ -group_of_things."0".\ 
15916  ▷ ▷ ▷ -cubics-0."texture{pigment{White*0.5.transmit0.5}}.\ 
15917  finish{ambient-0.4.diffuse-0.5.roughness-0.001.reflection-0.1.specular-.8}.\ 
15918  ▷ ▷ ▷ -point_list_from_csv_file\ 
15919  ▷ ▷ ▷ function_Clebsch_map_of_circle_to_defining_eqn_N1000.points.csv\ 
15920  ▷ ▷ ▷ -group_of_things.as_interval-0.6-656\ 
15921  ▷ ▷ ▷ -spheres-1.0.07."texture{pigment{color{Red}}}.\ 
15922  finish{diffuse-0.9.phong1}.\ 
15923  ▷ ▷ ▷ -point_list_from_csv_file\ 
15924  ▷ ▷ ▷ function_Clebsch_map_of_circle_to_defining_eqn_r2_N1000.points.csv\ 
15925  ▷ ▷ ▷ -group_of_things.as_interval-656-1042\ 
15926  ▷ ▷ ▷ -spheres-2.0.07."texture{pigment{color{Blue}}}.\ 
15927  finish{diffuse-0.9.phong1}.\ 
15928  ▷ ▷ -scene_objects_end\ 
15929  ▷ ▷ -povray_end\ 
15930  ▷ ▷ -rm.r POV\ 
15931  ▷ ▷ mkdir.POV\ 
15932  ▷ ▷ mv.Clebsch_2curves_0.*.pov.POV\ 
15933  ▷ ▷ mv.makefile.animation.POV\ 
15934  ▷ ▷ \ # ▷ ▷ -point_list_from_csv_file.function_Clebsch_map_of_circle_N1000.points.csv.\ 
15935  ▷ ▷ \ # ▷ ▷ group_of_things.as_interval-0.954\ 
15936  ▷ ▷ \ # ▷ ▷ spheres-1.0.07."texture{pigment{color{Red}}}.\ 
15937  finish{diffuse-0.9.phong1}.\ 
15938  
15939  
15940  
15941  
15942  F7.povray:\ 
15943  ▷ ▷ $(ORBITER_PATH)orbiter.out-v.2.-povray\ 
15944  ▷ ▷ -round-0.-nb_frames_default-30\ 
15945  ▷ ▷ -output_mask-F7_15_lines.%d.%03d.pov\ 
15946  ▷ ▷ -video_options-3.1024--H-768\ 
15947  ▷ ▷ -global_picture_scale-1.5\ 
15948  ▷ ▷ -default_angle-80\ 
15949  ▷ ▷ -clipping_radius-4.4\ 

728
- omit_bottom_plane\n- camera-0:"1,1,1".-4.5,3.5,6."0,0,0".-\n- end\n- scene_objects\n- cubic_lex:0,0,6,0,0,-13.39014946,-3.341901346,-6.9792931640,5.827182718,0,0,7.390149464,-7.390149464,-6.9792931640,-1.512349728,-8.485281372,0 ,0,0,0-.\n- group_of_things:"0"-.\n- cubics-0:"texture{pigment{White*0.5}.finish{ambient-0.4.diffuse-0.5.roughness-0.001.reflect-0.1.specular-0.8}}-.\n- line_through_point_with_direction:"0,0,0,1,0,0"-.\n- line_through_point_with_direction:"0,0,0,-1,0,1,0"-.\n- line_through_point_with_direction:"0,0,0,0,-1"-.\n- line_through_point_with_direction:"1,0,0,1,1,1"-.\n- line_through_point_with_direction:"-1.414213562,0,0,4.146264370,1.73205808,1.73205808-.\n- line_through_point_with_direction:"0,1.73205808,1.73205808,1.73205808,1.73205808-.\n- line_through_point_with_direction:"-2.133352390,-1.674708020,1.0"-.\n- line_through_point_with_direction:"-2.539058015,-2.133352390,0,0,0,0,0,0,0,0,-1"-.\n- line_through_point_with_direction:"1,0.148188060,0,-0.9435440612,1"-.\n- line_through_point_with_direction:"-0.971197117,0,0,1.162155272,0,1"-.\n- line_through_point_with_direction:"2.096037870,2.096037870,0,-1.0658519 05,-1.065851905,1"-.\n- line_through_point_with_direction:"3.921555783,2.921555781,0,-1.7224565 85,-1.722456585,1"-.\n- group_of_things_as_interval:0-0.12-.\n- lines:1-0.04:"texture{pigment{color-Yellow}.finish{diffuse-0.9.phong -1}}-.\n- scene_objects_end-.\n- povray_end-.\n- rm-POV-.\nmkdir-POV-.\nmv-F7_15_lines_0_*.pov-POV-.\nmv-makefile_animation-POV-.\nmv-F7_video-.\n- rm-r FRAMES.
- mkdir FRAMES
- rm fifteen_with_lines.mp4

$(ORBITER_PATH)orbiter.out

prepare frames

- i:0-30-F7b/F7_15_lines_0_%03d.png
- output_starts_at:0
- o:FRAMES/frame%04d.png
- end

ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png -f mp4 -q:v 0 -vcodec mpeg4 fifteen_with_lines.mp4

McKean_povray:

$(ORBITER_PATH)orbiter.out -v 2 -povray
- round:0 -nb_frames_default:30
- output_mask:McKean_%d_%03d.pov
- video_options:-W 1024 -H 768
- global_picture_scale:1.5
- default_angle:80
- clipping_radius:4.4
- omit_bottom_plane
- camera:0."1,1,1".."-4.5,3.5,6"."0,0,0"..
- end

- scene_objects
- cubic_lex:"0,0,1,0,0,-1,-2,1,"
- group_of_things:"0"
- cubics:0:"texture{pigment{White*0.5}}"
- reflection:0.4-diffuse:0.5-roughness:0.001
- reflection:0.1-specular:.8}

- scene_objects_end

- povray_end
- rm -rf POV
- mkdir POV
- mv McKean_0_* .pov POV
- mv makefile_animation-POV

McKean_video:

- rm -r FRAMES
- mkdir FRAMES
- rm McKean.mp4

$(ORBITER_PATH)orbiter.out

prepare frames

- i:0-30-MCKEAN/McKean_0_%03d.png
- output_starts_at:0
- o:FRAMES/frame%04d.png
- end
ffmpeg -r 5 -f image2 -i FRAMES/frame%04d.png \
-f mp4 -q:v 0 -vcodec mpeg4 McKean.mp4

# Section 13.4: Continuous Function Plotter

SECTION_CONTINUOUS_FUNCTION_PLOTTER:

lissajous:
$(ORBITER_PATH)orbiter.out -v 2 \n-smooth_curve "lissajous" 0.07 2000 0.15 0.18 0.85 \n-const a 3 b 2 c 1.57 r 7 \n-const a 3 b 2 c 1.57 r 7 \n-var t \n-code \n-push t push a multi push c add \n-push r multi return \n-push t push b multi sin \n-push r multi return \n-code_end \n
function lissajous_N2000_points.csv

lissajous.plot:
$(ORBITER_PATH)orbiter.out -v 2 -povray \n-round 0 -nb frames default 1 \n-output_mask lissajous %d %03d.pov \n-video_options -W 1024 -H 768 \n-global_picture_scale 0.4 \n-default_angle 45 \n-clipping_radius 5 \n-omit_bottom_plane \n-camera 0 0 0 10 \n-rotate about z_axis \n-end \n-scene_objects \n-line through two points recentered from csv_file \n-coordinate_grid.csv \n-group_of_things "0" \n-group_of_things "1" \n-group_of_things "2" \n-lines 0.09 "texture{pigment{color Yellow}}" \n-lines 1 0.09 "texture{pigment{color Yellow}}"
-lines 2.0.09:"texture{pigment{color Yellow}}".\n-group of things as interval 3.39:\n-lines 3.0.02:"texture{pigment{color Black}}".\n-point list from csv file:\n-function lissajous N2000 points.csv\n-group of things as interval 0.6524:\n-spheres 4.0.1:"texture{pigment{color Red}}".\nfinish { diffuse 0.9 phong 1}".\n-plane by dual coordinates "0,0,1,0".\n-group of things "0".\n-planes 5:"texture{pigment{color Blue*0.5\ntransmit 0.5}}".\n-scene objects end\n-povray end\n-rm -rf POV\nimkdir POV\nmv lissajous_0_*.pov POV\nmv makefile animation POV\nlissajous_3d:\n$(ORBITER_PATH) orbiter.out -v.2:\n-smooth curve "lissajous_3d" 0.07:2000:50:0:18.85:\n-const a 3 2 2 0.157 r 7 const end\n-var t -var end\n-code\n-push t push a mult push c add sin push r mult return\n-push t push b mult sin push r mult return\n-push t return\n-code end\nlissajous_3d plot:\n$(ORBITER_PATH) orbiter.out -v.2 -povray\n-round 0 -nb frames default 30:\n-output mask lissajous_3d %d %d %03d pov\n-video options -W 1024 -H 768\n-global picture scale 0.40\n-default angle 45\n-clipping radius 5\n-omit bottom plane\ncamera 0 0 1 7 7 5 0 0 1\n-rotate about z axis\n-end\n-scene objects\n-line through two points recentered from csv file\ncordinate grid csv\n-group of things "0\ngroup of things "1"
misc_select:
$(ORBITER_PATH)orbiter.out.-v.3\n-define-F-finite_field-q7-end
-with-F-do-finite_field_activity--cheat_sheet_GF--end
$(ORBITER_PATH)orbiter.out-v.4-csv_file_select_rows_and_cols\nGF_q7_multiplication_table_reordered.csv
"0,2,4","0,2,4".

misc.join:

$(ORBITER_PATH)orbiter.out -v 4 \n-csv_file_join=poly_orbits_d3_n3_q2_select_F2.csv Orbit_idx \n-csv_file_join=poly_orbits_d3_n3_q2_select_F4.csv Orbit_idx \n-csv_file_join=poly_orbits_d3_n3_q2_select_F8.csv Orbit_idx \n-csv_file_join=poly_orbits_d3_n3_q2_select_F16.csv Orbit_idx \n-csv_file_join=poly_orbits_d3_n3_q2_select_F32.csv Orbit_idx.

Section 14.2: Limitations

SECTION_LIMITATIONS:
Bibliography


[34] Fatma Karaoglu and Anton Betten. The number of cubic surfaces with 27 lines over a finite field. To appear in Journal of Algebraic Combinatorics.


[54] L. Schläfli. An attempt to determine the twenty-seven lines upon a surface of the third order and to divide such surfaces into species in reference to the reality of the lines upon the surface, Quart. J. Math. 2 (1858), 55–110.


