Convex Geometry: MATH 114
A SHORT INTRODUCTION TO POLYMAKE

Polymake is a framework to manipulate polytopes and polyhedra. One can compute important features and visualize these objects.

1 Quick Basics

Each polyhedron is stored in a single file and then the typical polymake command is

```
polymake file_name COMMANDS
```

COMMANDS should always be a sequence of all-capitals words separated by spaces. For example

```
polymake cube.poly N_FACETS SIMPLE
```

Calculates the number of facets and decides whether cube.poly is a simple polytope. So essentially, polymake answers questions that one asks about a polytope!

In addition to the polymake COMMANDS, there are independent polymake CONSTRUCTIONS that allow you to create complicated polytopes from nothing. Here are a few examples of CONSTRUCTIONS:

```
rand_sphere randompoly 3 20

n-gon myngon n

cube cube.poly 3 0

cross crosspolytope 4 5

simplex tetrahedron 3

product s3xs3.poly s3.poly s3.poly

permutahedron permutapoly 5
```

Lines starting with # are treated as comments. They can be arbitrarily intermixed with the data lines.

2 Creating a polytope or polyhedron

WARNING: To do this you need to use a computer editor: VI or EMACS are two possibilities available in the computer lab.

It can be created either by specifying the vertices or by giving the inequalities. For example, a 3-cube can be expressed by

```
POINTS
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
```

But the same polytope can be given by the six inequalities

```
INEQUALITIES
0 1 0 0
0 -1 0 0
0 0 1 0
0 0 -1 0
0 0 0 1
0 0 0 -1
```

IMPORTANT: In POLYMAKE one writes inequalities in the form $b - Ax \geq 0$ which means the left column numbers is the vector b and the other entries are the **negatives** of the entries of A.

But one can also use other operations such as intersection, Minkowski sum, etc.

```
intersection <output_file> <input_file_1> <input_file_2> [ ... ]
(Construct a new polyhedron as the intersection of given polyhedra).

conv <output_file> <input_file_1> <input_file_2> [ ... ]
Construct a new polyhedron as the convex hull of given polyhedra.

minkowski_sum <output_file> <scalar1> <infile1> <scalar2> <infile2>
Produces the polytope lambda*P+mu*Q, where * and + are scalar
multiplication and Minkowski addition, respectively.
```

3 Visualizing a polytope

To see a three-dimensional polytope one can write

```
> rand_sphere random.poly 3 20
> polymake random.poly VISUAL
```

These commands first create random polytope with 20 vertices and then they display an interactive picture. For 4-dimensional polytopes one can print a Schlegel diagram of the polytope:

```
polymake P SCHLEGEL
```

4 Dimension, Facets and vertices

To compute the dimension of a polytope:

```
> polymake square.poly DIM
DIM
2
```

One can count vertices or facets

```
polymake square.poly N_FACETS

N_FACETS
4
```

You can also study a polytope using a vertex-facet-incidence matrix. For each facet you have a line with a list of the vertices contained in that facet. The vertices are specified by numbers. They are numbered consecutively starting from 0. In each row the vertices are listed in ascending order.

The following is a valid polymake description of a square.

```
VERTICES_IN_FACETS
{0 1}
{1 2}
{2 3}
{0 3}
```

5 Linear Optimization and volumes

One can maximize a linear functional (which helps finding a box that contains the polytope completely or the support functions). For example one can create a file

```
linear_program.poly
```

containing

```
LINEAR_OBJECTIVE
0 1 1 1
INEQUALITIES
0 1 0 0
0 0 1 0
0 0 0 1
1 -1 0 0
1 0 -1 0
1 0 0 -1
5/2 -1 -1 -1
8 -1 17 0
```

then we can find the optimal value of the functional by

```
polymake linear_program.poly MAXIMAL_VALUE
MAXIMAL_VALUE
5/2
```

Similarly, the volume of the polyhedron can be found by the command

```
polymake -vv linear_program.poly VOLUME
polymake: reading rules from ...
VOLUME
47/48
```