Using Polymake

Madeline Brandt

January 26

Introduction

Polymake is a piece of software run through the terminal used for studying combinatorics and geometry of convex polytopes. It can also handle computations related to simplicial complexes, matroids, polyhedral fans, graphs, permutation group actions, and tropical geometry. Online, there is an extensive collection of tutorials for using each of these types of objects. In this talk, I will outline the basics of how to use Polymake, and I will discuss how to use the tropical application for computations related to this course. The website *www.polymake.org* is very useful in learning more about Polymake.

Installation

At *www.polymake.org*, click downloads. Here, you can find a link to the online version of Polymake. The online version has the advantage that you can run the latest beta before new versions are released. The online version is slow, however. To install Polymake, follow the instructions here to download the most recent version for your machine. The website *polymake.org/doku.php/howto/start* has very specific instructions.

Getting Started

Click the Polymake icon (orange tetrahedron) to start a terminal window running Polymake. There are a few things to note about using Polymake before attempting any computations:

- Polymake speaks Perl. There is a Perl tutorial online that can be useful. I will list a couple of things here to get you started. Scalar variables start with a $, arrays start with a @, and hashes start with a %. Properties of an object are defined using $->$. All commands end in a semicolon. If you press enter before putting in a semicolon, you can type a semicolon and push enter again. Scalar properties can be used in arithmetic:

```
polytope > $p=cube(3);

polytope > $i =($p->N_FACETS*$p->N_FACETS)*15-7;

polytope > print $i;
533
```

- Polymake uses homogeneous coordinates. (Why? “in order to obtain a unified view on the polytope and the cone section of a polyhedron”) Therefore, when defining Polytopes in terms of a convex hull of points, you must set the first coordinate to 1 on all of those points.
- Polymake works over the rationals and does all computations precisely. This means that if you define a polytope as the convex hull of 200 points and compute the volume, your answer will be the quotient of two integers with around 70 digits each. If you are doing computations with large numbers of points and Polymake is taking too long, you can try Qhull which works numerically and can compute similar things as Polymake.
- Polymake has several different *applications* which can be used. The default application is *polytope*. Another example is *tropical*. Applications correspond to collections of mathematical objects and their properties from a specific subject, and have various commands for those objects. Polymake always has one current application, indicated by the first word on the command line when you type, and the others are running in the backstage. Each application has webpage listing all of the object types and properties related to that application. To change applications, type *application “name”;* as follows:

```
polytope > application "tropical";


Application tropical currently uses following third-party software packages:
cdd, jreality, libnormaliz, lrs, nauty, permlib, ppl, singular, singular, sketch
, sympol, threejs, tikz, tosimplex
For more details:  show_credits;

Warning: some rulefiles could not be configured automatically
due to lacking third-party software and/or other issues.
To see the complete list: show_unconfigured;
tropical >
```

Polytopes

Some well-known families of polytopes are built-in to Polymake, such as *cube(n)*. Other than this, there are two main ways to define polytopes in Polymake.

V-Description

You may define a polytope as the convex hull of finitely many points. Here, we will make a 3-simplex. Remember that Polymake uses homogeneous coordinates.

```
polytope > $points=new Matrix<Rational>([[1,0,0,0],[1,1,0,0],[1,0,1,0],[1,0,0,1]
]);

polytope > $p=new Polytope<Rational>(POINTS=>$points);
```

Now, Polymake can tell us various things about $p:

```
polytope > print $p->VOLUME;
1/6
polytope > print $p->N_POINTS;
4
polytope > print $p->POINTS;
1 0 0 0
1 1 0 0
1 0 1 0
1 0 0 1

polytope > print $p->H_VECTOR;
1 1 1 1
```

You can find a complete list of the properties one can ask for at the website for the application *polytope*: polymake.org/release_docs/2.12/polytope.html.

H-Description

We may also define a polytope as the intersection of half-spaces. In Polymake, one denotes the inequality $a_0 + a_1x_1 + \cdots + a_nx_n \geq 0$ as $[a_0, a_1, \ldots, a_n]$. To define our 3-simplex in this way, we type:

```
polytope > $inequalities=new Matrix<Rational>([[0,1,0,0],[0,0,1,0],[0,0,0,1],[1,
-1,-1,-1]]);

polytope > $p=new Polytope<Rational>(INEQUALITIES=>$inequalities);
```

Example

Problem 12 from the book says: What is the maximal number of facets of any four-dimensional polytope with 8 vertices? How many edges are in such a polytope?

Answer: the cyclic polytope achieves the upper bound. The cyclic polytope has any 8 points form the set $\{(i, i^2, i^3, i^4) \mid i \in \mathbb{R}\}$ as its vertices. Can you guess how many edges it will have?

```
polytope > $points = new Matrix<Rational>([[1,0.725426, 0.526243, 0.381751, 0.27
6932], [1,1.71403, 2.93789, 5.03562, 8.63119], [1,1.80351, 3.25263, 5.86615, 10.
5796], [1,1.89725, 3.59956, 6.82927, 12.9568], [1,1.74107, 3.03134, 5.27779, 9.1
8902], [1,1.06607, 1.1365, 1.21158, 1.29162], [1,0.201847, 0.0407422, 0.00822368
, 0.00165993], [1,1.25097, 1.56492, 1.95766, 2.44898]]);

polytope > $p = new Polytope<Rational>(POINTS=>$points);

polytope > print $p->F_VECTOR;
8 28 40 20
```

It has $28 = \binom{8}{2}$ edges!

Regular Subdivision

We compute the regular subdivisions for the examples on pages 64-65.

```
polytope > $M = new Matrix<Rational>([[1,0,1],[1,1,1],[1,2,1],[1,3,1]]);

polytope > $w = new Vector<Rational>([4,2,1,2]);

polytope > $F = regular_subdivision($M,$w);

polytope > print $F;
{0 1}
{1 2}
{2 3}

polytope > $w= new Vector<Rational>([3,2,1,2]);

polytope > $F = regular_subdivision($M,$w);

polytope > print $F;
{0 1 2}
{2 3}
```

```
polytope > $M = new Matrix<Rational>([[1,2,0],[1,1,1],[1,0,2],[1,1,0],[1,0,1],[1
,0,0]]);

polytope > $w = new Vector<Rational>([1,0,1,0,0,2]);

polytope > $F = regular_subdivision($M,$w);

polytope > print $F;
{3 4 5}
{1 2 4}
{1 3 4}
{0 1 3}
```

Can you construct a weight vector to get various subdivisions?

Tropical

I will now briefly discuss tropical arithmetic in Polymake. To use this, you either need the newest version of Polymake, or you should use the online version. Polymake can do tropical computations, and you can define tropical polynomials, and take tropical determinants.