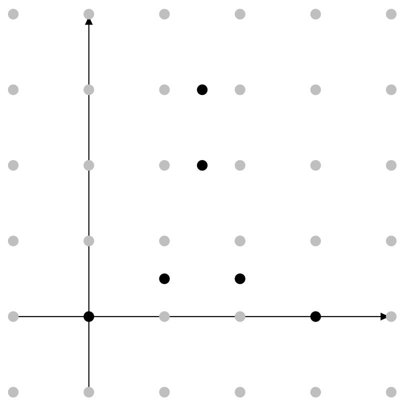
Second Discussion Session for Lectures on Tropical Geometry

Alheydis Geiger and Paul Görlach

August 25, 2020

Exercises

1. Consider the point configuration of the black points below. Draw the inner normal fan to its convex hull and determine all possible triangulations of the point configuration. Which of these are regular?



2. Show: Is $I \subset K[x_0,...,x_n]$ a homogeneous principal ideal generated by the homogeneous polynomial $f \in K[x_0,...,x_n]$, then f is a universal Gröbner basis of I.

3. Let $I = \langle 7 + 8x_1 - x_1^2 + x_2 + 3x_2^2 \rangle \subset \mathbb{C}[x_1^{\pm},x_2^{\pm}]$.

   a) Compute all initial ideals of $I_{\text{proj}} \subset \mathbb{C}[x_0,x_1,x_2]$ and draw the Gröbner complex of I_{proj}.

   b) Draw $\{\mathbf{w} \in \mathbb{R}^2 : \text{in}_{\mathbf{w}}(I) \neq \langle 1 \rangle\}$.

   c) Repeat a) and b) for $J = \langle tx_1^2 + 3x_1x_2 - tx_2^2 + 5x_0x_1 - x_0x_2 + 2tx_0^2 \rangle \subset \mathbb{C}\{\{t\}\}[x_1^{\pm},x_2^{\pm}]$.

4. Find a tropical basis for $I = \langle x_1 + x_2 + 3, x_1 + 5x_2 + 7 \rangle \subset \mathbb{C}[x_1^{\pm},x_2^{\pm}]$.

Hints

2. Read the proof of Lemma 2.6.2. (3).

3. Consider Theorem 2.5.7.

4. Follow the procedure from the proof of Theorem 2.6.6.

Software Code and more Information

Exercise 1 Compute the normal fan using polymake (if you do not have polymake installed, you can try it online: https://polymake.org/doku.php/boxdoc):

```
application "fan";
$P = new Polytope(POINTS=>[[1,0,0],[1,3,0],[1,1.5,3]]);
$nf = normal_fan($P);
$nf->VISUAL;
```

How to check whether a subdivision of the polytope P given by the maximal Cells C or D is regular:

```
$P = new Matrix<Rational>([[1,0,0],[1,3,0],[1,1.5,3],[1,1,0.5],[1,2,0.5],[1,1.5,1.5]]);
$C = new Array<Set<Int>>([[0,3,4],[0,1,4],[1,4,5],[1,5,2],[2,3,5],[0,2,3],[3,4,5]]);
$S = new fan::SubdivisionOfPoints(POINTS=>$P,MAXIMAL_CELLS=>$C);
$S->VISUAL;
print is_regular($P,$C);
$D = new Array<Set<Int>>([[0,1,3],[1,3,4],[1,4,5],[1,5,2],[2,3,5],[0,2,3],[3,4,5]]);
print is_regular($P,$D);
```

Exercise 3 How to compute the Gröbner Complex and all the initial ideals of an ideal I with Singular, using its polymake and gfan interface (if you do not have Singular installed, you can try it out online: https://www.singular.uni-kl.de/tryonline):

```
option(noredefine);  //suppresses warnings when redefining variables.
LIB "tropical.lib";

//===========
//Exercise 3a
//===========

ring r = 0,(x(0..2)),dp;
ideal I = 7*x(0)^2+8*x(0)*x(1)-x(1)^2+x(0)*x(2)+3*x(2)^2;
"Computing the Groebner fan...";
fan GF = groebnerFan(I);

int d = dimension(GF);
for(int i = 1; i <= d; i++)
{
  int m = numberOfConesOfDimension(GF,i);
  "Number of cones of dimension",i,":",m;
  if(m > 0)
  {
    "The corresponding initial ideals are:";
    for(int j = 1; j <= m; j++)
    {
      cone C = getCone(GF,i,j);
      bigintmat w = relativeInteriorPoint(C);
      ideal inI = initial(I,w);  //no need to compute a Groebner basis, as I is principal
      inI, "(weight vector:", -w, ")"; //WARNING: Singular uses MAX-convention, hence -w
    }
  }
}

//===========
//Exercise 3c
//===========

ring r = 0,(x(0..2)),dp;
```

```
number t = 7;
//(Here, we compute with respect to the 7-adic valuation.
// Note that the Groebner complex for our ideal here is the same as
// when working over the field of Puiseux series.)
ideal J = t*x(1)^2+3*x(1)*x(2)-t*x(2)^2+5*x(0)*x(1)-x(0)*x(2)+2*t*x(0)^2;

//for computations of initial forms (with valuations) we will compute in an auxiliary ring:
ring s = 0,(t,x(0..2)),dp;
ideal J = t*x(1)^2+3*x(1)*x(2)-t*x(2)^2+5*x(0)*x(1)-x(0)*x(2)+2*t*x(0)^2;

setring r;

fan GC = groebnerComplex(J,t);
//This computes the fan over the Gröbner complex (taking the cone over each polyhedron)
int d = dimension(GC);
for(int i = 1; i <= d; i++)
{
  int m = numberOfConesOfDimension(GC,i);
  "Number of cones of dimension",i,":",m, "(including cones at infinity)";
  if(m > 0)
  {
    "The corresponding initial ideals are:";
    for(int j = 1; j <= m; j++)
    {
      cone C = getCone(GC,i,j);
      bigintmat w = relativeInteriorPoint(C);
      if(w[1,1] != 0)
      {
        setring s;
        ideal inJ = subst(initial(J,w),t,1);
        matrix ww[1][3] = 1/w[1,1]*w[1,2..4];
        inJ, "(weight vector:", string(ww), ")";
        setring r;
      }
    }
  }
  "";
}
```

When I know how the Polyhedral Complex looks I can visualise it with `polymake` (this is the code for the Gröbner Complex of Exercise 3c)):

```
application "fan";
$pc1 = new PolyhedralComplex(POINTS=>[[1,0,0],[1,1,1],[1,1,4],[1,-1,0],[1,-1,4],
[1,4,1],[1,4,-1],[1,0,-1],[1,-3,-4],[1,-4,-3]],
INPUT_POLYTOPES=>[[0,1,2,3,4],[1,2,6],[0,1,6,7,8],[7,8,10],[0,3,8,10,12],[3,4,12]]);
$pc1->VISUAL;
```

There is also a way to use Singular and polymake combined so you can use the output from `Singular` to `polymake` and visualize it there. See https://polymake.org/doku.php/install/installsingular

More on computing tropical bases:
https://software.mis.mpg.de/tropicalBases/index.html