Title

January 14, 2020

1 Getting Started

Say that we are given n polynomials with complex coefficient in n variables, such as the following.

$$f_1 = (x-1)(y-3)$$
$$f_2 = (x-2)(y-4)$$

Our goal is to describe algorithmically all points of the set

$$\mathcal{V}(f_1, f_2) = \{(z_1, z_2) \in \mathbb{C}^2 : f_1(z_1, z_2) = f_2(z_1, z_2) = 0\}$$

which we call the *zero locus* or the *vanishing* of f_1 and f_2.

Bézout's Theorem (one formulation of it) states that if $f_1, \ldots, f_n$ are complex polynomials in n variables of degrees $d_1, \ldots, d_n$ and $\mathcal{V}(f_1, \ldots, f_n)$ is finite, then the size of $\mathcal{V}(f_1, \ldots, f_n)$ is at most $d_1 d_2 \ldots d_n$. For now we will assume that our system has finitely many solutions. Therefore the degrees $d_1, \ldots, d_n$ give an upper bound on the size of the output, and in fact they are necessary for the algorithm itsself.

To solve the system above, first we declare the names of our variables in a file named "bertiniInput_variables" with the following contents.

```
variable_group x,y;
```

Next we declare the equations in a file named "bertiniInput_equations" with the following contents.

```
function f1,f2;
f1 = (x-1)*(y-3);
f2 = (x-2)*(y-4);
```

Create a file called "bertiniInput_trackingOptions" which we will leave blank for now.

Finally, create a file called "inputFile.py" with the following contents

```
degrees = [[2], [2]]
workingDirectory = "run"
```

Now we

2 Multiple variable groups

3 Projective variable groups

4 Nonsquare systems