Nikoleta E. Glynatsi

Resultants in SymPy

Jun 5, 2018

I have recently been doing some work related to systems of polynomial equations. This work has lead me to study several academics papers on **resultant theory**. In this blog post I aim to give an introduction to resultant theory and a demonstration of how we can use the Python package **SymPy** to calculate the resultant of systems. I will end the blog post by describing the particular contribution I've recently made to **SymPy**.

To start of let us familiarise ourselves with some mathematical definitions and I am going to use SymPy to demonstrate several examples.

```
>>> import sympy as sym
```

Initially what is polynomial? A **polynomial** is an expression consisting of variables and coefficients. Using the following code we can define the polynomials f and g which are depended on a variable x.

```
>>> x = sym.symbols('x')
>>> f = x ** 2 - 5 * x + 6
>>> g = x ** 2 - 3 * x + 2
```

Now a **system of polynomial equations** is a set of simultaneous equations. For example, the following two expressions compose a system of two polynomials where both must equal 0.

$$f(x) = 0$$

$$g(x) = 0$$

We are interested in the values of x for which both equations f and g simultaneously fall to zero. More superficially we are interested if values for which all polynomials are nullified do exist! Though there are other methods for

addressing such problem here we consider the **resultant**.

The resultant of two polynomials is a polynomial expression of their coefficients, which is equal to zero if and only if the polynomials have a common root.

Several resultant formulations exist within the literature. One of the most common ones is Sylvester's resultant which is defined as the determinant of **Sylvester's matrix**. Sylvester's formulation is implemented within SymPy (**docs**) and it can easily be calculated using a few lines of code.

```
>>> from sympy.polys import subresultants_qq_zz
>>> matrix = subresultants_qq_zz.sylvester(f, g, x)
>>> matrix
Matrix([
[1, -5,  6, 0],
[0,  1, -5, 6],
[1, -3,  2, 0],
[0,  1, -3, 2]])
```

By calculating the determinant of Sylvester's resultant we know that the system has a common root. That is because the determinant is equal to 0. The common root is for $x = 2$ which is trivial if we were to factorise f and g.

```
>>> matrix.det()
0
>>> f.factor()
(x - 3)*(x - 2)
>>> g.factor()
(x - 2)*(x - 1)
```

The resultant can do more than just assure us that systems do have roots. For example when we have a system of 2 polynomial equations in two variables we can solve for one variable where the second one is kept as a coefficient. This actually allow us to find the roots of the system. That is also why the resultant is often refereed to as the **eliminator**.

Let's consider another example where now f and g are also depended on y.

```
>>> y = sym.symbols('y')
>>> f = x ** 2 + x * y + 2 * x + y -1
>>> g = x ** 2 + 3 * x - y ** 2 + 2 * y - 1
```

```
>> matrix = subresultants_qq_zz.sylvester(f, g, y)
>> matrix
Matrix([
[x + 1, x**2 + 2*x - 1,               0],
[    0,          x + 1, x**2 + 2*x - 1],
[   -1,               2, x**2 + 3*x - 1]])
>>> matrix.det().factor()
-x*(x - 1)*(x + 3)
```

What we see is that in order for the system to have a common root, x must be $\in \{-3, 0, 1\}$. Now by substituting these values of x, each at a time, in f and g and repeat the process and find roots for y.

But what if we had a more generic system? Lets say a system of m polynomials in n variables. Sylvester's formulation would not be able to be applied to such systems. For such systems we use **multivariate resultants**.

A number of multivariate resultants can be found in the literature. An example of a multivariate resultant is **Dixon's resultant**. Let us consider our final example with a 3 polynomial system dependent on x and y.

```
>>> from sympy.polys.multivariate_resultants import DixonResultant
>>> p = x + y
>>> q = x ** 2 + y ** 3
>>> h = x ** 2 + y
>>> matrix = DixonResultant(variables=[x, y], polynomials=[p, q, h])
>>> matrix.det()
0
```

Dixon's resultant, the determinant of the matrix, is equal to zero. This indicates that our system has indeed a common root.

Multivariate resultants have many advantages. It allow us to know if a large system has roots but moreover there are several ways the roots can be extract, but we will not cover those, however this great paper does **Comparison of various multivariate resultant formulations**.

Multivariate resultants were not implemented within SymPy until recently. I am very happy to say that Dixon's and **Macaulay's resultant** have been my first contribution to the library. Sympy is a great project that continuously helps with me with my

research and I am thrilled to have contributed to the project!

Contact me

Nikoleta-v3

NikoletaGlyn

GlynatsiNE@cardiff.ac.uk