MACAULAY2 - RTG SEMINAR

OCTOBER 25TH, 2023

Macaulay2 can compute the singular locus. Let's learn about that. We'll also learn about the `FastMinors` package and a bit about how to play around with testing how long things take to run.

Exercise 1. Create a singular ring `R` (ie, `R = ZZ/101[x,y,z]/ideal(x*y-z^2)`). Run

```
singularLocus R
```

Figure out what it is doing. Discuss with people around you why this might be slow.

Exercise 2. Suppose `R = ZZ/101[x,y,z]/ideal(f)` and `S = ZZ/101[u,v,w]/ideal(g)` are two homogeneous coordinate rings of (smooth) curves X and Y. Use curves with simpler equations, like `f = x^3 + y^3 + z^3` or `g = x^3 + y^3 + z^3 + x*y*z`.

Recall that $X \times Y$ can be embedded in $\mathbb{P}^8$ via the Segre embedding. On a section ring, if `T = ZZ/101[a_0..a_8]` this is given by a surjection:

$$\texttt{T} \longrightarrow \texttt{R \# S}$$

which sends `a_0` $\mapsto$ `x*u`, `a_0` $\mapsto$ `x*v`, etc.

Macaulay2 won't create the Segre product for you (well, maybe some packages might), let's do it manually. Note `R # S` in our case is a subring of

```
ZZ/101[x,y,z,u,v,w]/ideal(f,g).
```

Use this and the `ker` function to give a presentation of the homogeneous coordinate ring of $X \times Y$ inside $\mathbb{P}^8 \cong \operatorname{Proj} \texttt{T}$. You can check your work for example by making sure the `dim (T/J) = 3 =` $1 + \dim X \times Y$.

Exercise 3. Now, take the homogeneous coordinate ring of $X \times Y$ (it will look like `T / J`). Try to compute `elapsedTime singularLocus (R/J)`. You can give up after a little while. Take a look at `jacobian J` to understand why this is taking so long. Figure out how many determinants Macaulay2 is trying to compute (note `dim(T/J)` $= 3 = \dim(X \times Y) + 1$). (Note the singular locus should be the irrelevant ideal, why?)

In general we don't know a good way to compute all those determinants, or to compute the non-singular locus a different way. One approach to this problem is to try to compute some of the determinants and see if that's enough.

Try loading the package:

```
needsPackage "FastMinors"
```

Then run the commands

```
elapsedTime regularInCodimension(1, T/J)
elapsedTime regularInCodimension(2, T/J) --this one might take a little while (luck?)
```

What Macaulay2 is doing is choosing different submatrices, computing their determinants, and periodically checking to see if they prove that that `T/J` is regular in codimension 1 or 2. (In this case if you chose nonsingular curves, `T/J` is regular except at the origin, and it's dimension 3, so it should be regular in codimension 1 and codimension 2, but not 3). You can see what Macaulay2 is doing by running something like:

```
elapsedTime regularInCodimension(1, T/J, Verbose=>true)
elapsedTime regularInCodimension(2, T/J, Verbose=>true)
```

If you just want to compute the minors[1] directly, try to use the `chooseGoodMinors` function. Since in this case we want 6×6 submatrices, try:

```
partialJ = elapsedTime chooseGoodMinors(20, 6, jacobian J);
```

This will choose an ideal of 20 "interesting" 6×6 minors.

Exercise 4. Probably something more useful might be to prove that the singularLocus of `T/J` really is the irrelevant ideal (up to radical). You can do that manually by adding one minor at a time. First run:

```
J1 := J;
```

Now run

```
J1 = chooseGoodMinors(1, 6, jacobian J, J1, Strategy=>StrategyPoints) + J1; dim J1
```

Rerun that line over an over until you get 0. Or make a loop to do it manually. `Strategy=>StrategyPoints` is a very accurate strategy for finding good minors, but finding each minor can be slow (you can also run it on the `regularInCodimension` functions above with Verbose on, to see it in action). When running this we are adding one minor at a time, and checking the dimension.

Once you get zero, you've found an ideal of minors that defines a zero dimensional locus. Compute `radical J1` to verify it really is the irrelevant ideal.

[1]determinants of certain submatrices