

# Sage: Curves of One Variable in Differential Geometry

Dylan Estes

June 2, 2014

This was a project done 'collaboratively' between myself, Dylan Estes, and Evan Sadler. I modeled Curves of One Variable and Evan did Surfaces of Two Variables as well as Surfaces of Revolution. Each of us worked on our code independently, except for the instance described below. Other than being used for the same Math 442 class, they have no other interactions or similarities. Therefore his project is not very important in the context of my project, although still very cool. As stated in the presentation on Friday, this project was motivated in order to provide an additional educational tool for the Math 442 class -Differential Geometry- given that a vast majority of the class dealt with models of curves and surfaces in three dimensions but was only represented only in two dimensions (on a chalkboard). The classes we wrote were to help provide an additional tool for students in the Math 442 class.

Inside this folder there are several different files. There is the `GeometricCurve.sage`, which is solely the code I wrote for this project. Then there is `GeometricCurveCodeWithExamples.sagews`. This is the code the I wrote with some ready made examples listed at the bottom for the user to play with as well as comments explaining why that example is being used. Additionally, there is a beamer - `GeometricCurveInformation.pdf` - included that gives a short presentation giving the Definitions and Properties used in this class that were important in Math 442. This gives an extremely brief overview in order to contextualize the code with no Differential Geometry background. For anything deeper, other sources are recommended. The other beamer -`FinalPresentation`- was the presentation beamer we used, although the important parts are already in `GeometricCurveInformation.pdf`.

Finally, I included part of Evan's code as I helped him with some suggestions on optimization and since technically we're working on a collaborative project. You'll find two files, `SurfaceRevolutionInitial.sagews` and `SurfaceRevolutionFinal.sagews`. The initial code ran rather slow and I helped Evan to try and speed it up. There were minor changes in restructuring the code, but the biggest problem was that in the initial code, when `plot3d` was run on a list of

3-tuples, (points (i,j,k)) it would try and assign a keyword to the point when an iterator was used inside the plot3d function. This makes no sense for a point to contain any sort of keyword, especially in this instance. This is a bug in Sage and the discovery as well as the solution to work around this are mainly the reasons why I included this section in my project as it took several hours to discover as well as find a solution to. Several tickets have already been filed on this issue and well as suggestions on how to fix the Sage code. The way we found to work around it inside Evan's code was instead to create each point individually and then plot them together. Note the subtly: the original used a plot of a list of points, running into a bug with the iterator in plot3d on the list, and the second uses a list of plot points where the iterator was used in the list rather than plot3d. If you want to view it, it takes a bit to compile. I also included his code for surfaces of two variables, SurfaceOfTwoVariables.sagews, although again this isn't technically part of my project as I didn't help on that project at all.

Hopefully this gave a good overview of the project: why its important and what it is. One part that bugs me about my code is that I had to make the explanation of what each vector does as print statements in the plot() function since the 3D sage modeler doesn't having a legend display. I meant to make it look a little more professional, but couldn't find any smoothly integratable displays. Also, when using the plot() function it may not intialize or load with boxes. This is a 'bug' in sage. Just re-run the code again and it should load properly. Hope you enjoy the project.