Solution Curves of Families of Polynomial Systems

Jakub Dlugosz, Kevin Mazzarella, Anja Schulz
University of Illinois at Chicago

Mathematical Computing Laboratory

UIC

Project Description

PHCpack is a software package for solving systems of polynomial equations. One method the package can use to try to solve a given system is homotopy continuation. To better understand how the particular continuation method used by PHCpack performs near "difficult" points, we visualize data generated by the software using Python and Matplotlib.

Python Package

To visualize data from PHCpack in a useful and aesthetically pleasing manner, we created a Python package called VISCON. The package consists of a collection of plotting functions along with a text-based wizard. The wizard provides an intuitive interface to the Python frontend phcpy. The code takes user-input polynomial systems and a solution to one of those systems, reformats it into PHCpack-parseable form, and then passes it to the PHCpack path tracker. Then, the data generated by the path tracker is passed into a visualization function of the user's choice. The plotting functions range from allowing the user to plot multiple polygonal curves on one axis, to generating 3d animations with a custom colormap. To view the code, visit the project's GitHub repository: *github.com/mrkrtkpf/viscon*.

Example Continuation

Homotopy continuation is a numerical method for finding solutions to systems of equations. Homotopy continuation finds solutions to a system of equations $G(x) = 0$ by deforming a system of equations with known solutions $F(x) = 0$ little-by-little, tracking the movement of the solutions as it goes.

Example

$$F(x) = x^5 + 1 = 0$$

Solutions: $e^{i\frac{\pi}{5}}$, $e^{i\frac{3\pi}{5}}$, $e^{i\pi}$, $e^{i\frac{7\pi}{5}}$, $e^{i\frac{9\pi}{5}}$

$$G(x) = x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120 = 0$$

Solutions: ???

Homotopy Function

The function H defined by

$$H(x,t) = (1-t)F(x) + tG(x)$$

for $t \in [0,1]$ defines a family of polynomials, one for each value of t. The γ value is 1. Note that H "continuously deforms" F into G:

$$H(x,0) = (1-0)\cdot F(x) + 0 \cdot G(x) = F(x)$$

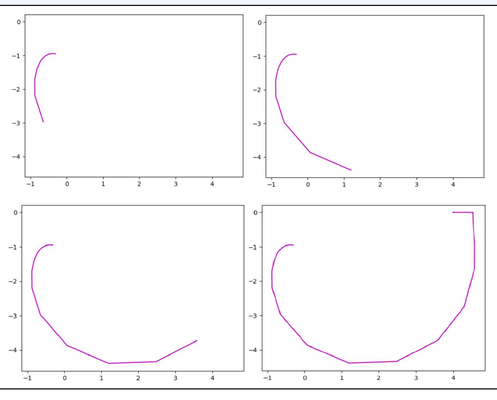
and

$$H(x,1) = (1-1)\cdot F(x) + 1 \cdot G(x) = G(x).$$

Explanation of Algorithm

A solution to $H(x,0) = 0$ will be near a solution to $H(x,t) = 0$ if t is near 0. Hence Newton's method can be used with any of the solutions to $F(x) = 0$ as initial guess to find a solution to $H(x,t) = 0$ if t is small. This solution to $H(x,t) = 0$ can then be used as an initial guess in Newton's method to find a solution to $H(x,t') = 0$ for t' near t. By incrementing t in small steps and repeatedly using Newton's method in this manner, eventually a solution to $H(x,1) = G(x) = 0$ can be found.

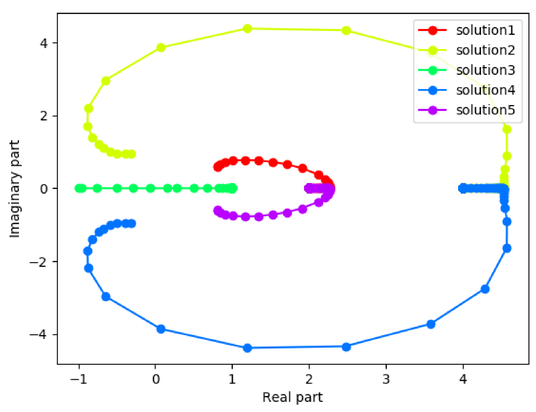
Animation of Path Tracker



Discussion

The path tracker tracks the solution $x = e^{i\frac{7\pi}{5}}$ of $F(x) = 0$ to the solution $x = 4$ of $G(x) = 0$. The concentration of points near the beginning and end of the path suggests that the path tracker takes its smallest steps near the solutions to the start and target systems and its largest steps in between the two. In the following plot of all solution curves computed by phcpy, we can see that the path tracker behaves in this manner when tracking the other solutions as well.

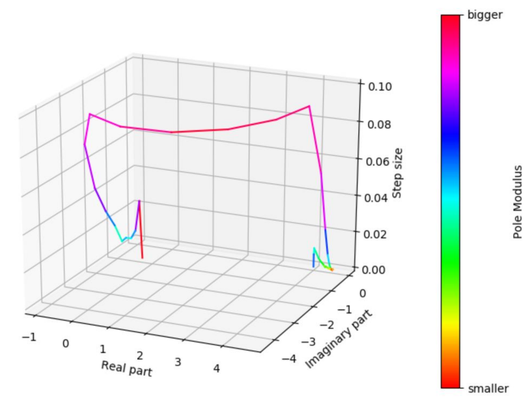
All Solution Curves



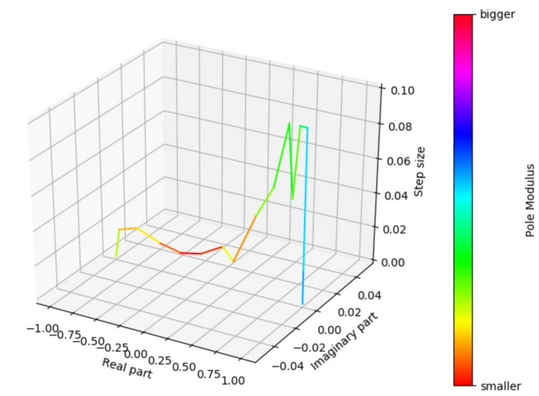
Discussion

Plotting the solution curve in three dimensions with the third dimension corresponding to the step size reveals that there must be other factors affecting how well the path tracker can accurately predict successive solutions. In the plots below, we see that the step size jumps around near the beginnings and ends of the paths.

Step Size vs Pole Modulus, Solution 4



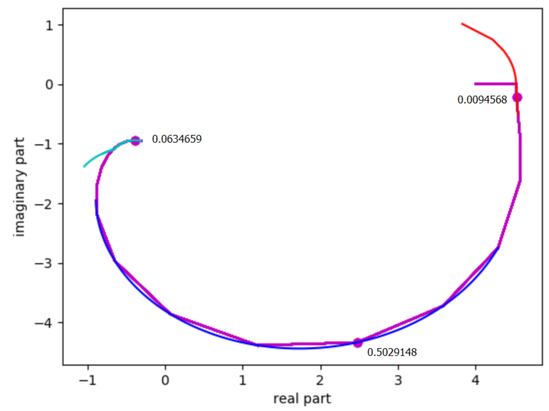
Step Size vs Pole Modulus, Solution 3



Discussion

The plots above use color as a fourth dimension. To make Newton's method converge more rapidly, the path tracker attempts to predict the location of the next point on the solution curve using a Padé approximant (a rational function in t) which approximates the shape of the curve. The colors in the above plots correspond to the moduli of the Padé approximant poles closest to the origin at each path tracker step. When the moduli are small the path tracker is forced to take smaller steps, as the accuracy of the prediction degrades when t is near a singularity.

Padé Approximants



Discussion

The plot above shows Padé approximants calculated by phcpy superimposed on the solution curve they are intended to fit. The number next to each point is the modulus of the pole of the Padé approximant calculated at that point which is closest to the origin. For these points, the accuracy of the approximation is correlated with the moduli of the closest poles. But it is apparent that the accuracy of the predictor depends on other (unknown) factors as well: the step size fluctuates at the beginning of the third solution curve even though the moduli of the closest poles remains relatively constant.

Challenges

A number of challenges were overcome in order to produce the end product. Scaling the data in three dimensions and finding a way to represent a fourth dimension required implementing a custom colormap for the data. Adapting Matplotlib's animation features for different presentations of data and types of plots also proved to be challenging. Due to portability issues, several mock user interfaces to the wizard written using PyQt, Python curses, and ANSI terminal codes were created before settling on the final text-based UI.

References

Morgan, Alexander. *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*. Prentice Hall, 1987.

Baker, George A., Jr. and Graves-Morris, Peter. *Padé Approximants*. Cambridge UP, 1996.

Bliss, Nathan and Verschelde, Jan. "The Method of Gauss-Newton to Compute Power Series Solutions of Polynomial Homotopies." *Linear Algebra and its Applications*, vol. 542, 2016, pp. 569-588.