Dynamical software exploration

Varun Wadia, Class of 2015

This summer I spent my time experimenting with mathematical software packages to determine which are best suited to different biological modeling operations. The final goal was to develop lab assignments for Prof. Zeeman’s Biomathematics class. When I took the class in my freshman fall we used a software called Berkeley Madonna to model the differential equation systems in the lab after having discussed them in class. This was meant to give students a better understanding of the equations themselves and how the system changes if the variables in the equations are modified.

Madonna worked well except that it was slow and would quit unexpectedly. The big plus point about Madonna as a software was that it was very clearly laid out and easy to learn. Hence a student could pick up how to use it within an hour and implement it effectively for the rest of the semester, because of this the class stuck with Madonna despite its shortcomings. Though this summer the lab computers were upgraded to Mac OSX Lion and hence Madonna wouldn’t run on them anymore. Therefore it was my job to find various potential replacement software packages, explore their capabilities completely and determine which were best suited to the various modeling operations the student is expected to complete in the lab. Once that was done I had to then try to solve all the assignments given in the previously used lab manual (written for Berkeley Madonna) and come up with my own ‘manual’ for the student to follow.

I first tried putting together my own software in the programming language python using the code snippets other programmers had already written and left online for public use. Python is a language with fantastic capabilities and a very English-like syntax, which makes it easy to use. That attempt failed because Python’s intrinsic GUI (graphical user interface) did not have the capabilities required to do the assignments and writing a new user interface would have taken too long and was beyond my programming ability.

I then looked into Mathematica and XPP. Both are good software packages with the required capabilities but Mathematica has a very difficult learning curve associated with it. It would take the student a very long time (people have spent full semesters) to master Mathematica well enough for them to implement it for the rest of the semester.

Hence, the software I ending up choosing was XPP, written by the mathematics department of the University of Pittsburgh for the sole purpose of differential equation analysis. It is very clearly laid out and has most of the capabilities needed for dynamical systems modeling of all kinds (disease modeling, predator-prey interactions etc.), thus making it the perfect tool to develop lab assignments that will cover and test the students’ knowledge of all the relevant material they must know as part of the class. I have completed a ‘manual’ with details on how to install, run and solve the lab assignments in XPP.

Faculty Mentor: Mary Lou Zeeman

Funded by the Howard Hughes Medical Institute Fellowship.