Astronomy 575: Project #3 (Fitting via a Grid Search)

Week 1:
Create a new managed C/C++ project within the ECLIPSE IDE. Within the project, write a C program (fgs01.c) which reads in a set of t (time) and v (velocity) data points from input files (fgs_a.dat through fgs_d.dat) available from the ASTR575 class web pages, performs a gridded search (described below) to fit the data and writes out both the final fit parameters and the total search results (rms values versus period and phase) to an output file (fgs_a.fit through fgs_d.fit). You may assume that there are no more than 2048 lines of data points within each input file.

The data files will contain times of observation (t) and radial velocities (v), derived from spectra of stars which may have companion planets, and thus manifest a sinusoidal variation in velocity as a function of time. Because we cannot obtain regular sampling of the velocity curve without an estimate of the period having been deduced before the observations, we will use a grid technique to solve for the period p and the phase ϕ of each signal. You may assume that the velocity data are distributed symmetrically about a value of zero, and that the signal amplitudes range from -1 to 1. Your grid search should begin as linear in phase, and logarithmic in period. We will restrict each period to the interval bounded by twice the minimum point spacing in time and twice the complete sampled time interval.

After calculating the rms quality of the fit throughout the $n \times m$ points (n in phase and m in period), use the SURFACE or CONTOUR macros within SUPERMONGO to map its distribution across the grid, to get a sense of the extent and frequency of local minimum values. Investigate the typical width of the minima, in order to possibly refine your grid spacing. Plot the quality of fit versus period, using the best individual phase value for each period value, and plot the quality of fit versus phase, using the best individual period value for each phase value. Overplot your best-fit sinusoidal waves on top of the t and v data points (adding error bars in v) for each case.

When your program is complete, create a compressed tarfile containing both your C source file and output files (fgs01.c and fgs_a.fit through fgs_d.fit), your MAKEFILE, your SUPERMONGO macro file (fgs01.mac), attach it to an e-mail (or add a report01.pdf file) describing your work (SUBJECT: PROJECT03-WEEK 1 MATERIALS FROM YOUR NAME), and submit it.

Week 2:
Begin with the best-fit position(s) within your two-dimensional $n \times m$ grid of phase and period values, and then perform an iterative search on the surrounding regions to fine-tune your results, alternating small steps in period and then in phase values. Your step size should decrease over time, via a functional form which matches the complexity of the phase-period plane data within each data set. Include a mechanism that checks for "stalled" test particles which no longer move, and either jumps them out of local cusps (where the rms increases in all four cardinal directions) or deactivates them. If you regrid the local region finely when caught in a cusp, to characterize the situation, should you devote resources equally to sampling in period and in phase, or prioritize one axis? Consider how much of a random element you wish to include in particle movement, as opposed to shifts in direction and step size based upon observable quantities.

Consider how best to evaluate the success of your algorithm efficiently, as you develop it. You might draw test particle positions in time on top of a regional rms contour map, for example, or plot trends in step size and rms as a function of time. You should determine clearly how and where most of your computational resources are being utilized. Do your test particles spend more of their time in regions with low, or high, rms? Do many particles end up at or near to the final best-fit position, or are they dispersed widely in period and phase? What fraction of test particles end up stalled out at the bottom of local cusps, and how quickly does this occur? Does your algorithm perform differently when presented with different data sets?

When your program is complete, create a compressed tarfile containing both your C source file and output files (fgs02.c and fgs_a2.fit through fgs_d2.fit), your MAKEFILE, your SUPERMONGO macro file (fgs02.mac), attach it to an e-mail (or add a report02.pdf file) describing your work (SUBJECT: PROJECT03-WEEK 2 MATERIALS FROM YOUR NAME), and submit it.

Week 3:
Explore additional aspects of potential iterative algorithms. For example, what are the effects of varying

the number of starting positions for iteration, or varying the coordinates of the starting positions? Should test particle algorithms vary explicitly with starting positions (dependent upon initial period for example), randomly, or not at all? Should one distribute a starting set of test particles across period-phase space (perhaps at a set of best-fit points drawn from a coarse grid analysis), or start multiple particles at the same position but with different algorithms controlling their movements?

Should step size be responsive to local rms, to terrain (the first or second derivative of local rms), to the number of elapsed steps, to the number of test particles that have gravitated to the local region, and/or to the cumulative mobility of the test particle? Should step size correlate with the expected width of sharp, deep features (low rms sinkholes), and if so is this a function of period or of phase? Are there systematic differences in general in the variation of terrain with period, versus with phase?

Should one begin with a large set of test particles and selectively cull them periodically, based on their success? Should one begin with a smaller set of test particles and selectively clone them periodically, based on their success? Should such clones be distributed around their parents in all cardinal directions, dropped randomly throughout the local region, or placed exactly with their parents with competing (variable) algorithms?

What risks are run (and how much time is gained) by evaluating rms from only a subset of the input data, for large input data sets?

What defines a successful iterative algorithm? Consider, for example, the accuracy of the final period and phase values, the speed of convergence to a final solution, the required memory allocation, the fraction of test particles who converge on the final solution, and the final distribution of all test particles.

Adapt your finalized source file so that it will analyze an additional two input data sets (fgs_g.dat and fgs_h.dat) if they are present. If your iterative technique depends on the results of an initial coarse grid, your code should conduct both phases of analysis as necessary. I will run all of the submitted algorithms on these two additional data sets, and we will compare the results (the range of period and phase solutions, speed of execution, and memory requirements) in class. Be sure to include a rough estimate ($\pm 10\%$ is fine) of the total required memory allocation in your project report for this week.

Analyze the results from the three weeks of work on fitting sine waves to the data sets. Discuss your results in a LaTex project report which incorporates key Supermongo plots to aid in your discussion. Include a table with final solutions for period and phase for each data set, run time (either a total value, or separated into grid, iterative, and total components), and rough memory allocation. Please be explicit about which Additional Tasks you have explored, and your conclusions. When your project report is complete, attach a tarfile containing your C source file (fgs03.c) and output files (fgs_a3.fit through fgs_d3.fit), your Makefile, your Supermongo macro file (fgs03.mac), and your project report (report03.pdf) to an e-mail (Subject: Project03-Week 3 Materials from Your Name), and submit it.

Week 4:

Model the effect of non-zero eccentricity for a planetary orbit on the radial velocity of its primary. Evaluate both the changes in the orbital velocities about the barycenter, and those viewed from an arbitrary viewing angle (though within the orbital plane). You may choose to perform your study entirely within Supermongo, as it will be plot-intensive.

You should perform a literature search to find observational data to compare to your model radial velocity curves; be sure to include the HD 80606b system. Discuss your results in a LaTex project report which incorporates key Supermongo plots to aid in your discussion. Please be explicit about any more Additional Tasks you have explored, and your conclusions. When your project report is complete, attach a tarfile containing your Supermongo macro file (fgs04.mac), and your project report (report04.pdf) to an e-mail (Subject: Project03-Week 4 Materials from Your Name), and submit it.

Additional Project Tasks (for Week 1 through Week 4):

1. For the data within fgs_d.dat, explore the effect of using a fraction of the data points rather than the whole set. What is the minimum complete sampled time interval (total time window) necessary to constrain the planetary period well? Could this be done in a single observing run? Compare also the results of fitting the first 10% of the observations versus fitting a randomly chosen subset containing 10% of the points.

2. One could also analyze these data using least-squares analysis. For the data within fgs_a.dat, re-express the relationship between t and v as a linear function, and apply your least-squares fitting function from Project01.

$$t = \frac{P}{2\pi}\left[\sin^{-1}(v) - \phi\right], \text{ where } y = t,\ x = \sin^{-1}(v),\ m = \frac{P}{2\pi}, \text{ and } b = -\frac{P}{2\pi}\,\phi.$$

3. The four data sets fgs_a.dat through fgs_d.dat were each stripped of any linear velocity offsets and then normalized in amplitude before being presented to you. Download the file fgs_e.dat, and make such corrections yourself. Construct a histogram of all velocity values, and then fit the two peaks in the distribution. By averaging these minimum and maximum amplitudes, you can estimate an average value for velocity. Conduct your sinusoidal analysis on the adjusted data, after first removing the average velocity and renormalizing the amplitude.

4. Begin with the best-fit positions within your initial $n \times m$ grid of phase and period values, and then focus a second $n \times m$ grid on the immediate regions surrounding these points (a simple adaptive refinement grid technique). Compare the output of this set of grid fits with that of your iterative fit.

5. Explore genetic algorithms, as an alternative to grid and iterative fits, using "Genetic Algorithms in Astronomy and Astrophysics" (Charbonneau 1995, ApJS, 101, 309) as a starting point.

6. Determine the expected rms value for a sine wave (with amplitude of unity) fit with randomly determined parameters, where

$$rms = \sqrt{\frac{\sum_{i=1}^{n}[v_{\rm mod}(i) - v_{\rm obs}(i)]^2}{n}}.$$

7. As one varies the eccentricity of a planetary orbit from 0 toward unity, does the probability of detection via a radial velocity search change? If so, how?

8. Explore the purpose of the minimum and maximum limits in period, and calculate the quality of fit for harmonics of the best-fit period which fall outside of the fitting range.

9. Perform a literature search to find observational data for radial velocity curves with multiple components. Explore the possibility (by adapting your C program to fit two independent sine waves with different periods, phases, and amplitudes) that they represent stars with two detectable orbiting planets.

10. Your C code should contain a sorting routine used to order the input data monotonically in time. Consider the following sorting possibilities: BUBBLE, HEAP, INSERTION, QUICK, SELECTION, SHELL; discuss their relative strengths and weaknesses, and justify your choice of sorting algorithm for this problem.

11. For each of the four initial data sets (fgs_a.dat through fgs_d.dat), use your period estimate to plot each data set over a single period (superimposing successive periods of velocity data with SUPERMONGO by changing the point sizes, shapes, and/or colors.

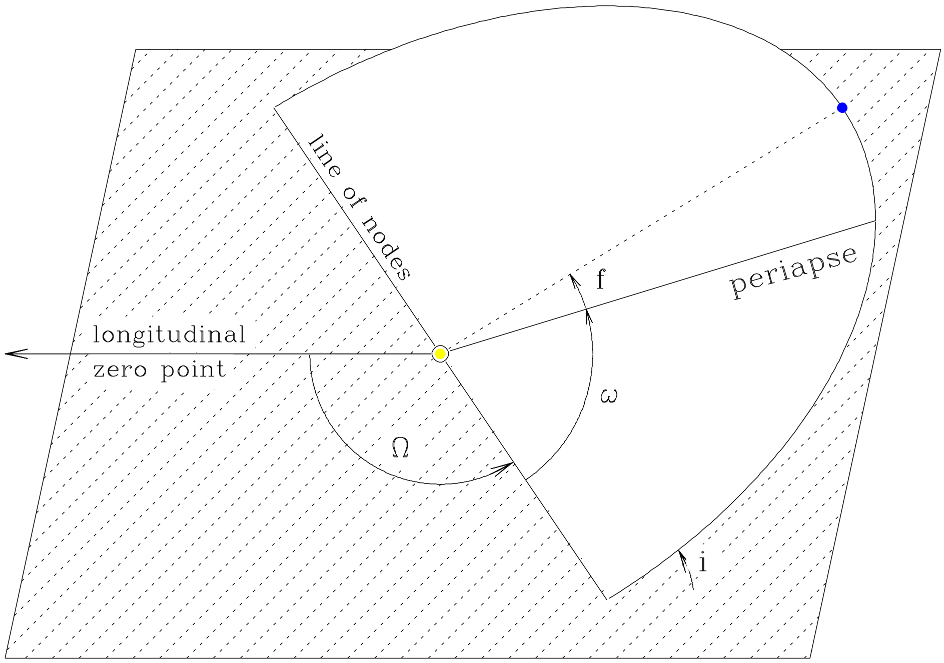
12. Recreate the following orbital mechanics diagram (see next page), via SUPERMONGO.

Additional Project Tasks (reprise):

The following Additional Tasks from our first projects are of good general utility. If you did not complete them previously, you may investigate them as a part of this project.

1. Add a time stamp to your output files indicating when the fit was performed.

2. Replace the individual variables which contain the fit parameters with a single STRUCT, so that they can all be passed to and from the MAIN program as a single argument.



3. Use MALLOC within your C program to dynamically allocate memory for the input data, to handle and arbitrarily large set of input data. You might either allocate memory in a series of blocks, or allocate the exact amount needed after first reading the input file once.

4. The basic MAKEFILE provided in this assignment is rather simplistic, because it recompiles the C source file whether or not it has been updated since the last compilation. Adapt the file so that it only runs `gcc` and `latex` when actually necessary.

5. Create a SUPERMONGO plot of the data within fgs_a.dat, in which points with an even integer x-value are solid and points with an odd integer x-value are hollow. Be sure that the point centers for the odd valued points are not obscured by the associated error bars.