W4260 Final Project

Projects are due May 5, 2015.

A substantial component of this course involves a larger-scale computational project which is formulated more independently by you, the student. This must involve some *significant* computation and should be within the general realm of physics, astrophysics or astronomy. When handing in the project, please include a brief but complete write-up. This write-up should include: (i) an introduction (including references to papers and other source materials); (ii) a brief description of the numerical method used (again including references if appropriate); (iii) a summary of the simulations/data analysis performed and the results obtained; (iv) a discussion of the implications of these results, and a conclusion.

You must choose a topic by Tuesday, April 21 and email me the topic chosen along with a one paragraph description of what you plan to do.

I provide below a list of suggestions for various projects. Note that these are often somewhat vaguely described, with the idea that the student would chose a particular direction to investigate. Projects outside of this list are very much encouraged, but please check with me to make sure the project is appropriate and feasible.

Some of the packages described here will require a C/C++ compiler. If you don't already have one, there are many open-source (and commercial) options. For example, Linux machines often come with compilers installed; the best choice for Mac's is probably the Xcode package (available from Apple's store); and windows machines can use GNU compilers (although some packages may not have support this platform).

1. **Numerical addition to a research project** The best problem would be a numerical application in an area of research that you have already carried out, or are currently carrying out. Talk with me if you would like to try this and we can come up with possible applications.

2. **Formation of an (idealized) spheroidal galaxies [N-body].** In this project, you will use an N-body code to investigate some simple models for the formation and evolution of the stellar component of a galaxy. First, find an N-body integrator (e.g. Joshua Barnes' treecode, Volker Springel's GADGET-2, or see me for a simple python integrator), then set up initial conditions for a uniform sphere of stars with radius r and mass M under three assumptions: (i) first, no motion: the particles would initially be at rest; (ii) circular orbits; (iii) isotropic pressure support (i.e. random velocities with a variance of $\sigma^2 \approx GM/r$). In each case you could follow the evolution of the density profile, total energy, and virial ratio (ratio of kinetic to potential energy).

3. **Disk galaxy evolution [N-body].** As in the previous problem, find and install a tree-based N-body solver. Create a distribution of particles in a rotationally supported disk. The number of particles should decrease exponentially radially to mimc the exponential density decrease of observed galactic disks: $\rho(r, z) = \exp(-r/r_0)\exp(-|z|/z_0)$, where r is the radius from the center along the plane of the disk, and z is the height of the disk. The velocities should be set to be purely tangential with a magnitude equal to the circular velocity at that radius (note that with an exponentially decreasing density profile, the rotational curve $v_c(r) = (GM(r)/r)^{1/2}$ will not be flat). Once you have set up the particle

distribution, use an N-body code to evolve the particles for a few rotation periods and observe how the disk evolves.

4. **Long-term planetary integrator** [**N-body**] The N-body integrator we wrote for problem set #4 is too slow and inaccurate to do planetary systems for long time periods (millions or billions of years). However, there are more sophisticated techniques – one such tool is the `SWIFT` package. Download and compile this package and use it to carry out the integration of an exoplanet system for a million years.

5. **Transit Timing Variations [Model Fitting].** In class, we fit a single transit and one of the parameters found was the central point of the eclipse (t_0). In this project, you would repeat that fitting for multiple eclipses (of the same object) and determine the central point of each eclipse: t_0, t_1, t_2, etc. Then determine, the time between eclipses ($t_1 - t_0$) and look for variations in these transit times. For a single planet/star system, the timings should be regular, but if there are variations, that could indicate an additional planet (or possibly even a moon).

6. **More sophisticated light curve fitting [Model Fitting].** We developed our own tools for fitting Kepler transit curves, but there are packages available to do more sophisticated analysis (in particular, `PyKE`). Use one of these packages to process the data from a different Kepler object, including combining the data, de-trending, and fitting.

7. **White dwarf structure [ODE].** Write down the set of ODEs that describe the structure of a white dwarf – this is similar to the problem set in class, but uses an equation of state for the degenerate pressure of electrons (i.e. a Fermi "gas"). To do the integration, choose a central pressure and integrate outwards in radius until the pressure is zero. Repeating this with different central pressures will give you a mass-radius relation for white dwarfs. Carry out the integration in the limits of non-relativistic and ultra relativistic electrons, showing that in the second case, there is a maximum mass (the Chandresekhar mass).

8. **Stellar structure [ODE].** Develop a simple stellar structure model by solving the three coupled ODEs that describes how a star on the main sequence looks given its mass. See, for example Carroll and Ostlie's introductory astrophysics textbook (or elsewhere) for the stellar structure equations, which look much like the ones for the planet, except the pressure is thermal pressure, rather than solid mechanical bonds, and we need to include an ODE for the diffusion of radiation and the production of radiation via nuclear burning. Carry out the computation for a one solar mass star and plot the resulting stellar structure (density, mass, temperature, luminosity vs. temperature).

9. **Gravitational Lensing [Integration].** Compute the magnification due to gravitational lensing as a background (luminous) star passes behind a compact (non-luminous) gravitational source, such as a black hole or white dwarf. The star will move with a velocity v so you should compute the magnification increase of the background star as a function of time. This kind of gravitational lensing is known as micro-lensing.

10. **Markov Chain Monte Carlo [Model Fitting].** Repeat the fitting of Problem Set 7 but using a Markov Chain Monte Carlo method to sample the allowed parameter space (e.g. the python package *emcee*). Find the best fitting parameters (under the same assumptions as in that problem) and also the uncertainties in the parameters. How do the results compare?

11. **Physics/Astronomy Teaching App.** Write a short demonstration of a concept from astronomy or physics (e.g. orbital dynamics, gravitational lensing, black-body radiation, magnitudes, etc.). Ideally the code would include some sort of interactive aspect (e.g. placing particles in an orbit) that would allow the student to explore the concept. The graphical component of this could be done either with a python graphical user interface library or the world wide telescope.

12. **Write your own tree code [Tree].** In this project, you would write a code that computes the gravitational acceleration using the tree method discussed in class. You can integrate Newton's equations using a leap-frog method. Use this N-body code to calculate the orbit of a thousand particles, checking to see how well the total energy is conserved.

13. **Exoplanet correlations [Model Fitting].** Download the dataset of the most recent exoplanet characteristics (mass, semi-major axis, ellipticity, etc.), or some other public datatset, and fit a relation between the exoplanet mass and semi-major axis, and other variables. Does the fit depend on how you account for the error bars? Comment on a possible physical explanation for the fits that you find.

14. **Image Noise Analysis [Statistics].** Determine the gain and read-out noise from a pair of GALAX CCD images.

15. **Cosmic Microwave Background (CMB) non-Gaussianity [Statistics].** The CMB has recently been measured with extreme precision with the Planck Satellite. Standard models (such as inflation) predict that the brightness temperatures across the sky should be Gaussian distributed – in this project you would download a CMB map from the Planck site and examine the distribution of the temperatures measured in all the map pixels, looking for any disagreement with a simple Gaussian distribution. You might, for example, compute higher order statistics such as the Kertosis.

16. **A better integrator [Integration].** Find a different way to carry out numerical integration (e.g. Guassian Quadrature, or 6th order Runge Kutta), and implement it. Apply it to the lookback integral we have done before and check to see how the error changes with N, the number of intervals. Does the error decrease with N as expected?

17. **Wave Equation [PDE].** Write a simple PDE solver (see Numerical Recipes or other text) to solve the wave equation. Investigate how a string evolves if it starts with a triangular kink in the center.

18. **Heat conduction [PDE].** Write a simple PDE solver (see Numerical Recipes or other text) to solve the heat equation in one dimension. Investigate the final temperature distribution of a bar of metal if one is held at a temperature T_0 and the other end is at temperature T_1.

19. **Fluid Dynamics with SPH [PDE].** Smoothed Particle Hydrodynamics (SPH) is an interesting and relatively simple way to solve the fluid equations. In this package, you would use the python (+C) package `pysph` to set up and solve for a simple fluid dynamics problem – the package includes a number of sample problems that you can run.

20. **Satellite galaxies [N-body].** Given a spherical galaxy realization, make several copies of the galaxy of different masses: in one set scale the positions so the density is the same;

and in a second keep all the same position (ie so density varies). In both cases scale the velocities so the galaxy you create is still in equilibrium. Check you've got it right by running the results in the treecode. All versions should be stable for at least 1 Gyr. Now alter the treecode accelerations and potential to include an external galaxy influence and run your galaxies in orbit around this analytic 'parent'. How do your results depend on density of the satellite galaxy? What happens to the stars in the satellite, and can you describe this analytically?

21. **Numerical relaxation [N-body].** Given a set of particles in a spherical system with isotropic velocities such that the system is in equilibrium, evolve the system using an N-body code for enough time that the particles get to orbit at least 20 times (assuming you are using a few thousand particles). Check the final time to make sure that the system doesn't change too much compared to the initial conditions. Then double the mass of 10% of the particles, randomly chosen (correct the mass of the remaining 90% of the systems to keep the total mass of the whole system constant). Evolve the system again and look at the distribution of the massive particles compared to the original distribution – in particular, are they more or less concentrated towards the center?

22. **Numerical Methods.** Look through *Numerical Recipes* or some other text on numerical methods, and find and implement a numerical method in Python (or another language). Possible examples include: interpolation, extrapolation, Fourier transform, linear algebra.

23. **Zombie Invasion [ODE].** Write down and solve the set of ODE's describing a Zombie invasion, described in this paper http://arxiv.org/abs/1503.01104. Confirm you can reproduce one of the solutions found in that paper, and then vary the initial conditions to explore the sensitivity of the results to parameters.

24. **Disk galaxy collision [N-body].** Given the distribution of particle positions and velocities for the stellar and dark matter components of a pair of disk galaxies (along with an associated bulge and dark matter halo, provided with the GADGET code), carry out a simulation of the collision, examining the morphology of the resulting system. [harder]

25. **Galactic plane vertical structure [ODE]** The gas in the galactic disk is quite thin: most of the HI gas in the disk is within about 300 pc of the central plane. However, even this thickness is less than predicted. If the gas were in hydrostatic equilibrium supported only by thermal pressure (assuming an ideal gas law), then it would have a scale-height of less than 100 pc. One reason for the extra thickness of the gas is that cosmic-rays produced in supernovae explosions provide an extra source of pressure. However, these cosmic rays also diffuse out of the plane and a detailed investigation of this point has never been made. In this project, the student would write down the set of (one-dimensional) differential equations including hydrostatic equilibrium combined with the diffusion equation for cosmic rays and solve them for a variety of parameters in order to solve for the time-independent vertical structure of the disk. [harder]