Due Monday, January 19 2009

The following are suggestions for final term projects. If you have an idea for another project, come talk to me about it, but I have to authorize all projects before beginning. You *must* choose a topic and let me know what it is (and you *should* start working on it) by the last day of the semester, Friday December 12. I will discuss the *mechanics* of the project (using CVS to manage source code, what we expect in your write-up, what to submit) in-class.

1. **Particle-Mesh Code:** Write a three-dimensional particle mesh code. Use your leap-frog integrator implemented in HW#4 to push the particles. To compute the forces on the grid, try using the NGP method to assign the particles to the mesh, use multigrid to solve the Poisson equation, and start by computing the boundary conditions for the potential using a multipole expansion with only the first order (monopole) term. I would expect writing the Poisson solver is the greatest effort in this project. Then try re-running the collapse of a cold sphere (HW#4 again) using as many grid points as you can afford (start with 16^3). A good reference for the gravitational collapse problem is van Albada, T.S., MNRAS, 201, 939 (1982). Try running some other problems, for example merging galaxies.

2. **Tree Code:** The same as project #1, but use a tree algorithm instead of PM to compute the gravitational forces. You can use either the Barnes-Hut (Nature, 324, 446, 1986) algorithm, or the KD-tree.

3. **Gas Dynamics:** Write a code to solve the equations of ideal gas dynamics in two spatial dimensions using a second order (piecewise linear) Godunov method. You will need to write a second-order spatial reconstruction function, a Riemann solver to compute the numerical fluxes, and a two-dimensional integrator. After testing the method (ask me about tests), try computing the interaction of a supersonic flow past a dense sphere in two dimensions.

4. **Solving Elliptic systems:** Solve Poisson's equation on a two-dimensional grid using both multigrid and Fourier analysis methods. Compare the time to solution for both methods, which is more efficient? Try parallelizing your multigrid code, and comparing execution times on a parallel system (you will need a parallelized FFT library).

5. **Porting code to GPUs:** If you have access to an NVIDIA graphics card, trying writing a new code, or porting an existing code, to run on the GPU, using the CUDA interface. Your application code must be highly parallel to exploit the architecture, for example, a good application is a direct N-body code. Many codes report a 100× speed-up compared to general purpose CPUs. How much of a speed-up does your application achieve?

6. **Your own idea!:** Do you have a significant computational challenge in your thesis research? (For example, optimization or parallelization of an existing simulation code.) If the tasks involved overlap significantly with the topics in this course, it could serve as a suitable term project. Come and talk to me about the project, especially goals and milestones.