CS/CNS 257c: Simulation
Due Saturday, June 5, 5:00pm *electronic submission*

Homework 7 - Final

In this homework/final you will be asked to implement two solutions to the gravitational N-Body problem in two dimensions. Under gravity, each planets position satisfies the following differential equation:

$$x_i'' = \sum_{j \neq i=1}^{n} G\, m_j \frac{x_j - x_i}{|x_j - x_i|^3}$$

1. $O(N^2)$ **N-Body Problem.**
   Implement the usual $O(N^2)$ N-Body algorithm, found in `gravAccelNSq.m`. Verify your implementation by simulating the motions of our solar system, found in `ss.dat`. All calculations should use real units, using the gravitational constant. For time integration, use the explicit Newmark scheme ($\beta = 0, \gamma = .5$).

   Each row of the data contains the mass, initial position and velocity of one of the planets (5 numbers in all) in the following units: distance=10^6 km, mass=10^{24} kg, time=10^7 sec.

2. $O(N)$ **N-Body Problem.**
   Implement an $O(N)$ N-Body algorithm, found in `gravAccelN.m`. Verify the correctness of your implementation by simulating the motion of our solar system. Verify the time complexity of the algorithm by calculating the time to find the acceleration for 1000, 2000, and 4000 planets using both algorithms. Show a short animation of the evolution of a large galaxy.

   You can use the function `makeGalaxy.m` to generate galaxies with arbitrary number of planets, or you can design your own galaxies if you wish.

   Hint:
   To implement the $O(n)$ algorithm, you will need to partition the bounding box of the planets into regions, calculating the total mass and center of mass for each region. Then you will need to *transport* acceleration between regions (similar to the radiosity transport algorithm discussed in class).

   One way to do this is to create a hierarchically m levels deep, subdividing regions in half at each level. You can evaluate the acceleration from one region to another by starting at the highest level and recursively dropping to finer regions when the oracle criteria dictates (i.e. $\frac{diameter}{dist} < c \leq \frac{1}{2}$). If the oracle criteria fails at the finest region, evaluate the effect of one region on another using the $O(n^2)$ algorithm from part 1.

You may implement this assignment in a language other than Matlab. If you choose to do so, please follow conventions similar to those above.