Assignment #6

Due: 6/5/2012

Assignment #7

Due: 6/5/2012

Modify your program in assignment #2 (with bug fixes that you have found) to:

1. Get N and D from command line arguments,
2. Randomly generate spheres within [-D:D, -D:D, -D:D]
3. Compute and store distances between all spheres, as in assignment #2.
4. Using the calculated distances, compute
   - 4a) the maximum distance between all spheres,
   - 4b) the minimum distance between all distances,
   - 4c) the average distance between all spheres, and
   - 4d) the number of pairs of spheres that collide.

Please parallel your code in 3 and 4 using OpenMP.

Discuss

1. Discuss the efficiency of storing N sphere data using various data layout strategies when the code is parallelized using OpenMP:
   1. Option 1: double xyz[N][3], r[N];
   2. Option 2: double xyzr[N][4];
   3. Option 3: double x[N], y[N], z[N], r[N];
   4. You may also explorer other possibilities (e.g. using struct & class)
2. Discuss the efficiency of storing distances data using various data layout strategies using OpenMP. Is the result different from your serial code (i.e. the best storage format is the same or different between OpenMP vs. serial code):
   1. Simulated 2D dynamic array + row-major orientation
   2. Look-alike 2D dynamic array + row-major orientation
   3. Simulated 2D dynamic array + column-major orientation
   4. Look-alike 2D dynamic array + column-major orientation

Discuss

3. Discuss your parallel efficiency in OpenMP under various problem sizes and data layouts. Is OpenMP always gives faste results?