COMP 258 Homework 2 Sep. 20, 1995, **Due:** Oct. 4, 1995

Problems

1. Use of I-COLLIDE: Ftp the code using the WWW pointer. The package comes with README files and a User's manual. It includes a sample program in the "nbody" directory, which demonstrates the use of the package. Use the I-COLLIDE collision package to measure the performance of the collision detection algorithms.

   - In the directory /afs/cs.unc.edu/project/geom/COLLISION/COURSE/Q2 different approximations of a sphere with triangles are given ($8, 32, 128, 512, 2048$ and 8192 triangles). Your goal is to measure the performance of the collision detection algorithms as a function of number of triangles. You will choose a start position for each sphere in a bounding box and assign a random velocity. You update the position after some discrete time instance and compute the closest features. You can also change the velocity vector at each instance using the random number generator. If the two spheres collide, their velocities are reversed. Choose a sequence with more than 2000 positions for each sphere and measure the total amount of time spent in collision detection and distance computation routines. Initially use the sphere approximated with 8 triangles, followed by $32, 128, 512$ and 8192 triangles. How does the performance of distance computation algorithm varies as a function of number of triangles? Plot a graph. Make sure that you use the same motion sequence for each approximation (controlled by the *seed* value to the random number generator).
   - Repeat the simulation with an approximation consisting of 128 triangles. Initially use four spheres in your environment and increase the number of spheres by eight till you have 100 spheres. How does the performance of collision detection algorithm varies as a function of number of spheres? Show it using a graph.

2. Implement an algorithm for the following: Given a polyhedron in 3-D (with no self-intersections), whether it is convex. There are 10 polyhedra, test*.dat, in the directory /afs/cs.unc.edu/project/geom/COLLISION/COURSE/Q1. Run your algorithm on each of these polyhedra and classify them into convex and non-convex polyhedra.