13.003
Computational Geometry and Visualization

Problem Set 2

Massachusetts Institute of Technology
Department of Ocean Engineering
Cambridge, MA 02139-4307

Out: February 14, 2000
Due: February 22, 2000

In this problem set, we will work with C and the graphics library OpenGL to visualize our yacht tender. A skeleton program graph2D, which opens a graphics window, can be copied from the 13.003 Course Locker. Between comments placed in the code and the introductory lecture on OpenGL, the source code should be self-explanatory.

1. Attach the 13.003 course locker and the glut locker. On an Athena SGI, type:

   athena% add 13.003
   athena% add glut
   athena% cd /mit/13.003/

With the course locker and the glut locker attached you will have access to the library files needed to compile graph2D. The code needed for this problem set is in the ProblemSets/PS2/ directory in the course locker.

2. Download the files README, Makefile, graph2D.c, util2D.c, and util2D.h to your directory. Examine the source code, compile it, and observe what it does. Notice the source code is in several files and a Makefile is provided to simplify compilation of the code (the README file gives additional information on using the “make” command).

Graph2D opens a viewpoint in which we will draw our yacht tender. The extents of the viewport are $-2.0 \leq x \leq 2.0, -2.0 \leq y \leq 2.0$ which is set in the routine reshapeWindow(int w, int h). Currently, the viewport is empty.

3. Add code to the routine display() to draw $x$ and $y$ axes in the viewport. How you draw the axes is left up to you but please provide tick marks at regular intervals and arrow heads on the axes.
Recall from Problem Set 1 that the sheer line (edge of deck) of our yacht tender is represented by two cubic planar parametric curves \( r_1(u) = \{x_1(u), y_1(u), z_1(u)\} \) and \( r_2(v) = \{x_2(v), y_2(v), z_2(v)\} \), where \( 0 \leq u, v \leq 1 \):

\[
\begin{align*}
\mathbf{r}_1(u) &= \\
&= \begin{bmatrix} x_1(u) \\ y_1(u) \\ z_1(u) \end{bmatrix} = \begin{bmatrix} -105.3372u^3 + 257.0999u^2 + 232.2372u \\ 57.0275u^3 - 279.1296u^2 + 392.1021u \\ 38.7212u^3 - 19.1985u^2 - 92.5227u + 170 \end{bmatrix} \\
\mathbf{r}_2(v) &= \\
&= \begin{bmatrix} x_2(v) \\ y_2(v) \\ z_2(v) \end{bmatrix} = \begin{bmatrix} -2.9684v^3 - 3.4818v^2 + 390.4533v + 384 \\ 10.7240v^3 - 68.9430v^2 + 5.2191v + 170 \\ -14.3627v^3 + 47.3284v^2 - 25.9656v + 97 \end{bmatrix}
\end{align*}
\]

For questions 4 through 6, ignore the \( z \) coordinate of these two curves and think of them as planar curves in two dimensions \( x, y \) (we will deal with the 3D curve in subsequent problem sets).

4. Modify the \textit{display()} routine to plot the sheer line (in two dimensions) in the viewport. Experiment with different OpenGL primitives for representing the sheer line.

5. Create a \textit{porcupine} plot of the curvature of the 2D sheer line. This consists of evaluating the curvature at regular intervals along the profile. At each evaluation point, place a line segment at the point normal to the curve with a length proportional to the evaluated curvature.

6. Compute the discontinuity of curvature \( |\kappa_2 - \kappa_1| \), at the interface of the two curves.

To make hard copies of your graphics output, use the program \texttt{xv}. Attach the graphics locker (type: \texttt{add graphics}) and then type: \texttt{xv} in a shell. \texttt{xv} will open a separate window. Press the right mouse button in that window to get a menu of options. With your program running, press the \texttt{grab} button (lower right corner) and then capture your window as an image. With the image captured, you can either print directly or save the image as a postscript file. Use \texttt{xv} to make hard copies of the output of your program in different development stages (axes only; axes and sheer line; axes, sheer line, and porcupine; any other experimenting you feel inclined to do).