Math 161, Assignment 4, due Monday, June 11

1. Make a 3D graphic illustrating two intersecting pipes. Both pipes have radius 1. The main pipe (length 4) has the x axis for its axis, and the second pipe (length 2) has the y axis for its axis, with the second pipe fitting into the main pipe as the leg of a “T”. Make sure that the main pipe is cut out properly (so that water could flow unrestricted from the second pipe) and that the second pipe does not project inside the main pipe. You should suppress the drawing of polygon edges in the graphic. Make an animation that amounts to flying a complete circle around the pipes so that one may look into both during the fly-around. You may wish to use the package Graphics`Animation`.

2. (a) Make a function diagonalize[m_] which, when given a symmetric real matrix m of any finite dimension, produces \{u, diag\}, where u is an orthogonal matrix and diag is the diagonal of a diagonal matrix d, such that \(m = u^{-1}.d.u\). (You may wish to make use of EigenSystem or perhaps SchurDecomposition.)

(b) Make use of the function in (a) to make a function qdiag[exn_, vblist_] which, when fed a quadratic expression exn in the variables vblist produces output \(\{c_1u[1]^2 + c_2u[2]^2 + \ldots, \{u[1] \rightarrow exn_1, u[2] \rightarrow exn_2, \ldots\}\}\), where the \(c_j\) are constants and the \(exn_j\) are the orthonormal changes of variables making
exn a sum of squares. For example, qdiag[x^2 + 4x y + y^2, {x, y}] should result in {3u[1]^2 - u[2]^2, {u[1] → (x + y)/√2, u[2] → (x - y)/√2}} or something numerically equivalent to that, and perhaps with some permutation of the variables.

3. (a) Design a function which, given a pair {p, q} of points in the plane, produces the path {p, r, s, t, q}, where r and t lie on the line segment pq and are respectively one third and two thirds of the way along pq, and rst is an equilateral triangle which projects to the left of the segment pq looking from p to q.

    (b) Design a function based on that in (a) which, given a list of points, produces another list achieving the following operation:

        (*) for every pair of adjacent points in the list, replace the pair by the five points as in (a), and remove duplicate endpoints.

    (c) Make a picture of the result of iterating this map 5 times, starting from an equilateral triangle traversed clockwise.

4. (a) Make a function ParametricPlot2 which accepts the same inputs as ParametricPlot, but which shows in red (superimposed on the graph) the sample points selected by ParametricPlot. Apply this to show a plot of the semicircle x = cos t, y = sin t, 0 ≤ t ≤ π.

    (b) Make a function myParametricPlot which accepts the same inputs as ParametricPlot, but fixes the choice of sample points by (i) sampling at PlotPoints evenly spaced parameter values; (ii) further subdividing up to PlotDivision times so that the successive sampled points don’t have tangent lines that turn by more than MaxBend. (You may find it useful to compute the approximate curvature at each sample point, as well as the distance between successive points.)