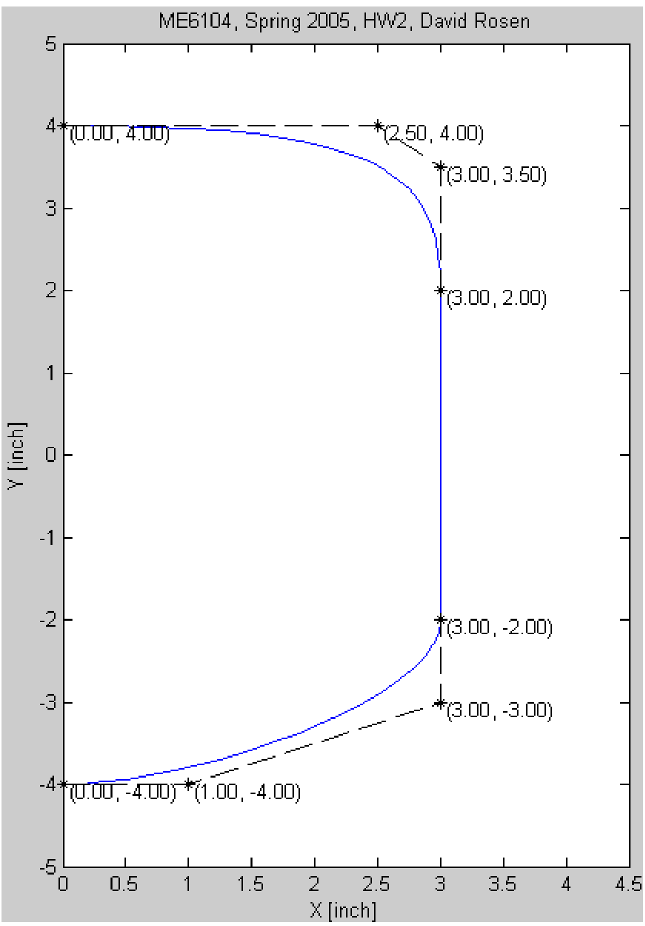
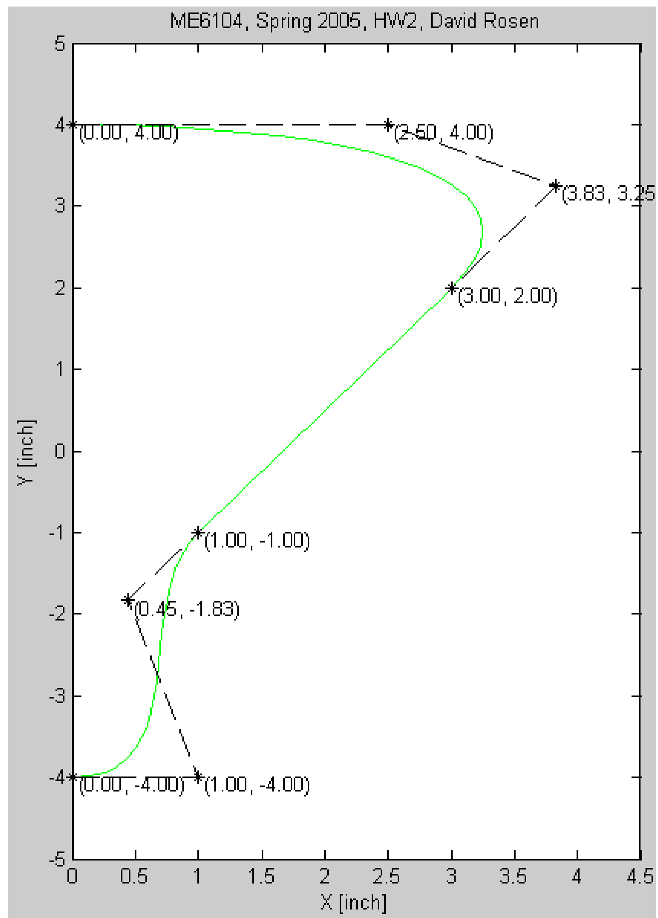
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Homework #2 Transformations & Parametric Modeling Due: Feb. 15
ME 6104 Fundamentals of Computer-Aided Design Spring 2005

LEARNING OBJECTIVES FOR PARAMETRIC CURVES MODULE OF ME 6104:

- Learning parametric curves: Hermite curves, Bezier curves, and B-Spline curves
- Learning composite curves and continuity
- Understanding the applications of parametric curves in geometric modeling

In this assignment, we will study the application of parametric curves in the context of processing CAD models for rapid prototyping. Consider the simple part cross section shown at left consisting of 2 cubic Bezier curves and 2 straight lines. On the right, the cross section is shown where the first CV on the second curve is translated to (1,1).

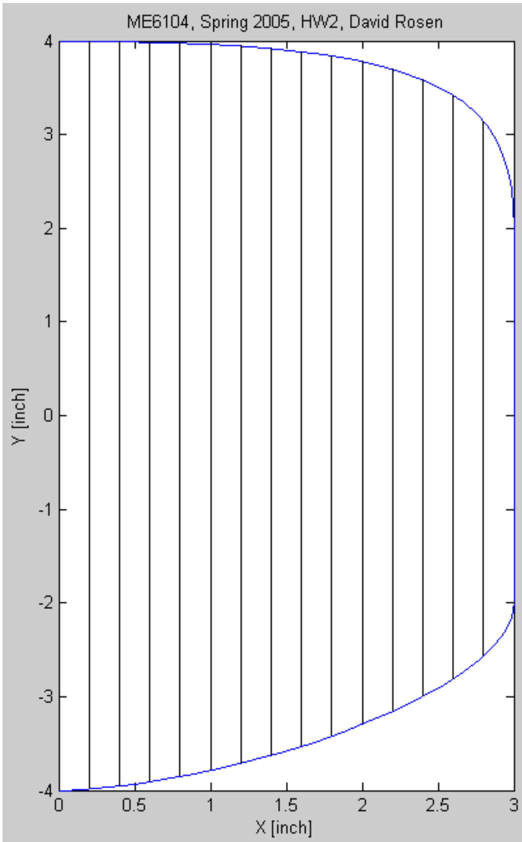


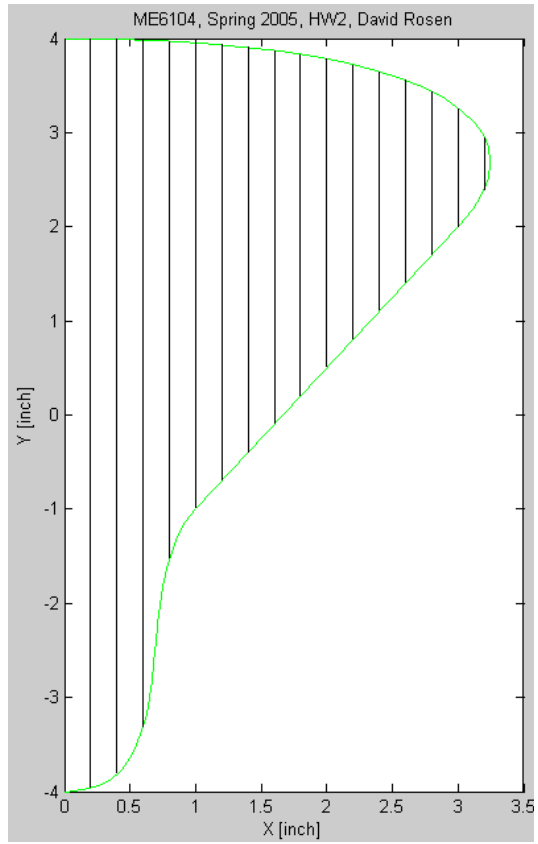


1. **(5 points)** Derive by hand the equations for both Bezier curves in their original positions (left fig).
2. **(10 points)** Develop a program in Matlab that enables you to 1) compute and plot points along a Bezier curve, 2) displays the cross section with the 2 Bezier curves and 2 lines, and 3) enforces G1 continuity among the curves and the line between them.

   Print a table containing the positions of moved control vertices. That is, print out the positions of the third CV (first curve) and second CV (second curve) when these CV's are moved to ensure G1 continuity.

3. **(35 points)** For fabricating parts in many rapid prototyping systems, it is necessary to slice each part cross section into individual vector strokes that a laser scans. The resulting strokes are called scan vectors, or simply scans. You are to develop a program that computes scan vectors for the two part cross sections shown above. The results should look like this with a scan vector increment of 0.2.





For a given X coordinate, a scan vector starts from the top Bezier curve and ends at the bottom curve, the line, or itself, for these cross sections. The intersection of a scan vector with a curve can be formulated as a root-finding problem. One way to solve this root finding problem is to first convert a Bezier curve equation into standard polynomial form, $x(u) = a\,u^3 + b\,u^2 + c\,u + d$, then to find the roots of this equation using the Matlab function 'roots()'. Code for calling roots() and interpreting the results for this problem is appended to this homework assignment.

Other methods of solving this problem can be used. You can write your own root finding function, use a calculator and manually solve the equations, etc. But I do want a Matlab program that generates plots similar to that shown above. If you are doing some of this work manually, you need not use a scan increment of 0.2. But do compute scan vectors at the following X positions, at a minimum: X = {0, 0.2, 1.2, 2.0, 3.2}.

Turn in a listing of your Matlab scripts and functions, output graphs, and a brief description of how you solved this problem.

Code to call the Matlab 'roots()' function and interpret results for this application:

```
%% Determines roots of polynomial for the scan vector application for
%% ME6104, Spring 2005, HW 2.  Valid roots are those that are real (not
%% complex) and that are between 0 and 1.
%% INPUT:
%%     poly - coefficients of polynomial to find the roots of.
%%            Coefficients must be ordered in descending powers.
```

```
%%
%% OUTPUT:  rts - vector of roots found

function rts = scan_roots (poly)

uvec = roots (poly);
uind = find (imag(uvec') == 0  &  real(uvec') <= 1.0001  &  real(uvec') >= 0);

rts = uvec(uind);
```

4. **(20 points)** Using your favorite CAD package, develop CAD models of the cross sections. Construct the cross section on the left (first figure) first using two splines and two straight lines. Ensure that you impose tangency constraints between the splines and the line between them. If you use IDEAS, you can enter the exact equation of the splines (Bezier curves). If you use ProEngineer, you create them as splines, but can then edit the control vertices (6 CV's are created. Delete 2 and change the coordinates of the others). If you use SolidWorks, you have little control over the curves. You need to enter points through which the spline passes (compute these using your Matlab code) and edit their coordinates as needed. Other packages will likely be similar to one of these mentioned.

   After imposing tangency constraints, extrude the section into a solid (extrusion distance = 2 units).

   Return to the cross section and move the CV from (3,-2) to (1, -1). What are the new CV locations? How did your curves change? Explain what happened.

   Extrude this modified cross section.

   Turn in screen-captures of your cross sections and your extruded solids. Also, write up a description of your modeling approach, difficulties you encountered, if any, and answers to the questions I asked. See the Homework Explanations link on WebCT for more guidance.