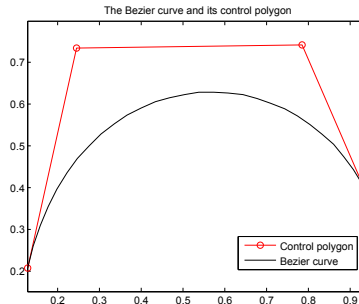


G22.3033-008: Geometric Modeling — Homework 1

Handout: Wednesday, 01/27/2010 — Due date: Wednesday, 02/10/2010

1 Practical implementation (70%)

Write a program that lets the user specify an arbitrary control polygon by clicking on the screen, computes the corresponding Bezier curve using the De Casteljau algorithm and displays the curve along with the control polygon. The output of your program can look something like this:



You may implement this assignment in MATLAB, which requires very little work. Below is a MATLAB script that lets the user click on points and displays the control polygon; you just need to write the De Casteljau evaluation of the Bezier curve at an arbitrary parameter value and display the curve.

```
% Script to input the control polygon and draw its curve.

% the degree of the curve - the user can enter an arbitrary number here
d = 3;

s = sprintf('Bezier curve of degree %d. Please input %d control points', d, d+1);
figure(1);
title(s);

[x,y] = ginput(d+1);

plot(x, y, 'ro-');
axis equal;

% Here you should compute some samples of the Bezier curve and plot it
% A useful function to get evenly-spaced parameter values is linspace
```

If you have never used MATLAB, there are plenty of tutorials on the web to get started. For example, this one: <http://www.math.ufl.edu/help/matlab-tutorial/index.html>.

You may also use any programming language of your choice; you will need to be able to open a graphical window and draw on it for that purpose.

1.1 Bonus assignments (30%)

Implement the following features for extra credit (10 points each):

1. Add the capability to interactively move the control points and update the Bezier curve on the fly.
2. Implement degree elevation.
3. Implement subdivision of the Bezier curve at a given parameter value t_{cut} .

2 Theoretical assignment (30%)

1. Prove that Bezier curves are affine-invariant.
2. Compute the minima and maxima of a Bernstein polynomial $B_i^n(t)$ (parameter values and function values).
3. One wishes to change the tangent direction of one of the endpoints of a Bezier curve. What modifications need to be made to the control polygon so that the prescribed tangent is attained? How about changing the tangent of an arbitrary point on the curve?
4. One wishes to move a point on the Bezier curve (say, with parameter value t_{move}) to a prescribed location while keeping the endpoints of the curve fixed. Is it possible to find a corresponding control polygon that will achieve this? If yes, how?