FMN011 Exam — Computational Part II
Working with Bézier Curves

On the course web page you will find more detailed instructions on how to report your results.

Deadline: May 12, 2011

1 Introduction

The purpose of this project is to study some of the properties of Bézier curves. You will design several curves and display them. Apart from writing the necessary codes, you need to give a short summary of the theory behind each algorithm, an explanation of how you solve each problem and the reasons for your particular choice. Make sure all your plots are correctly labeled and your codes are well documented. Every code must be tested and the results shown.

2 Bézier curves

Although a point on the Bézier curve with control points $P_0, P_1, \ldots, P_n$, is defined as

$$P(t) = \sum_{k=0}^{n} P_k B_k^n(t)$$

where B_k^n is the k-th Bernstein polynomial of degree n,

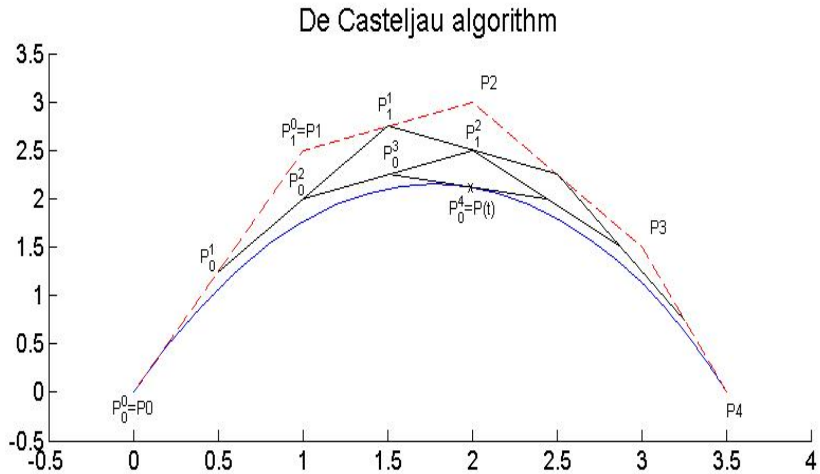
$$B_k^n(t) = \frac{n!}{k!(n-k)!} t^k (1-t)^{n-k}, \qquad t \in [0,1]$$

the most efficient way to compute the points on the Bézier curve is to use de Casteljau's algorithm:

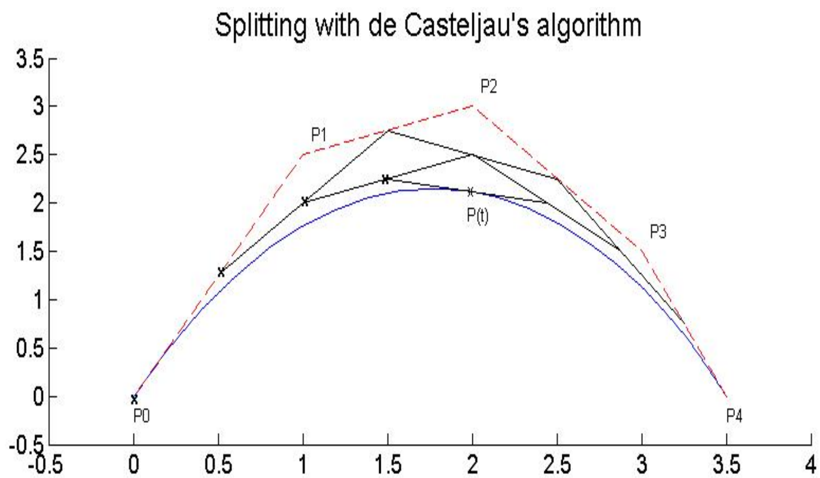
Given: Points $P_0, P_1, \ldots, P_n$ and $t \in [0,1]$,

set $P_i^k(t) = (1-t)P_i^{k-1} + tP_{i+1}^{k-1}, \quad k = 1, \ldots, n$ and $i = 0, \ldots, n-k$; starting with $P_i^0 = P_i$.

Then $P(t) = P_0^n(t)$.



If you want to split a Bézier curve at point $P(t)$ you can use de Casteljau's algorithm to find the control points for the new curves, as illustrated here.



The left curve has the control points $P_0^0, P_0^1, \ldots, P_0^n$, and the right curve has control points $P_0^n, P_1^{n-1}, \ldots, P_n^0$. Thus, splitting a curve comes "for free" with de Casteljau's algorithm.

The intersection of two given Bézier curves can be tested using the convex hull property of Bézier curves: *for $t \in [0,1]$, points $P(t)$ on a Bézier curve lie in the convex hull of its control polygon.* Thus, if the convex hulls of two Bézier curves do not intersect, the Bézier curves cannot intersect each other; if the convex hulls intersect, then it is possible that the curves also intersect each other.

If the two convex hulls of two Bézier points intersect, we can split each curve into two parts at $t = 0.5$ and check the intersections of the new (smaller) convex hulls. We can get rid of all pieces of one curve whose convex hulls do not intersect the convex hulls of the other curve's pieces, and continue splitting the remaining parts. The convex hulls will get smaller and smaller. An easy way to check these intersections is to consider rectangles that contain the convex hulls and that have sides parallel to the coordinate axes. Note that a stopping criterion is needed.

3 Tasks

1. Calculate the computational cost to generate a point on a Bézier curve of degree m
   (a) by means of its definition formula and
   (b) by de Casteljau's algorithm.
2. Implement de Casteljau's algorithm as a MATLAB function that generates a point of a Bézier curve with an arbitrary number of control points for any value $t \in [0, 1]$.
3. Compute and display Bézier curves (and their convex hulls) with the following properties:
   (a) A cubic Bézier curve.
   (b) An open curve of degree 4.
   (c) A closed curve.
   (d) A curve with a loop but no multiple control points.
   (e) A curve with a loop constructed with multiple control points.
4. Compute and display two cubic Bézier curves linked at their end and start points respectively, such that:
   (a) They do not connect in a smooth way.
   (b) They connect in a smooth way.
5. Plot the Bézier curves with control points [(0,0) , (1,1) , (2,0)] and [(0, 2.5) , (1 -2) , (2, 0.5)] and their convex hulls.
6. Write two Matlab functions that split a curve into left and right parts at an arbitrary value of t. Try it out by splitting the curve defined by the control points [(0,2.5) , (1,2.5) , (2,1.5) , (1,0) , (0,-0.5)] for t=0.5.
7. Write a function that constructs the smallest rectangle parallel to the x and y axes that contains the convex hull of a given Bézier curve. Try it out for the curve with control points [(0,2.5) , (1,2.5) , (2,1.5) , (1,0) , (0,-0.5)].

8. Write a Matlab function that calculates the intersection of two rectangles parallel to the x and y axes. Try it out for rectangles with vertices [(0,0) , (2,0) , (2,5) , (0,5)] and [(1,4) , (5,4) , (5,6) , (1,6)].

9. Write a Matlab function that calculates the intersection of two given Bézier curves. Use your program to study the intersection of the given pairs of Bézier curves:

   (a) P=[(-1,0) , (0,1) , (1.5,1.5) , (2.5,-1) , (3,0) , (2,2.5)] ;
   Q=[(0,2.5) , (1,2.25) , (2,1.5) , (1,0.25) , (0,-0.5)] ;

   (b) P=[(-1,0) , (1,1.5) , (2,2) , (3,-1) , (-1,0)] ;
   Q=[(0,2.5) , (1,2) , (2,1.5) , (2.5,0.5) , (3.5,1)] ;

10. Design a monogram with your initials containing two cursive letters. Give a list of the control points and show the monogram both with and without the control polygons.

April 1, 2011
Carmen Arévalo
Matematikcentrum, Numerisk Analys, Lunds Universitet