CAP5705 Computer Graphics
Homework 3
Curve Modeling

Manasi Sharma
UFID 90009559

Cubic C1 Bezier Curve Modelling from user input control points

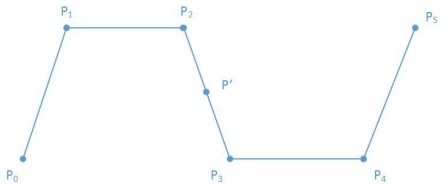
The program takes an input of minimum 4 control points from the user to design a cubic Bezier Curve through the control points through one of the following methods

1. Blending Function
2. De Casteljau's Algorithm
3. OpenGL API (glMap1f and glEvalCoord1f)
4. Sub Division

C1 continuity between Bezier Segments

An even number of control points, minimum 4, are taken as input from the user in order to maintain C1 continuity.

n Bezier segments are drawn for an input of 2n+2 control points.
For instance, Consider the case of 6 control points P_0, P_1, P_2, P_3, P_4 and P_5. We assume another point. P'. Such that it is the midpoint of P_2 and P_3. A cubic C1 continuous Bezier Spline is drawn with 2 Bezier Segments.
Segment 1 is drawn with the points P_0, P_1, P_2 and P' and segment 2 with the points P', P_3, P_4 and P_5.



Blending Function

In order to compute a Bezier curve between the given four control points P_0, P_1 and P_2 and P_3, this method uses Bernstein Polynomials (B_0, B_1, B_2, and B_3).

The point on the curve is given by,

$P = B_0 * P_0 + B_1 * P_1 + B_2 * P_2 + B_3 * P_3$

Bernstein Polynomials are defined by

$B_0 = (1 - t)^3$

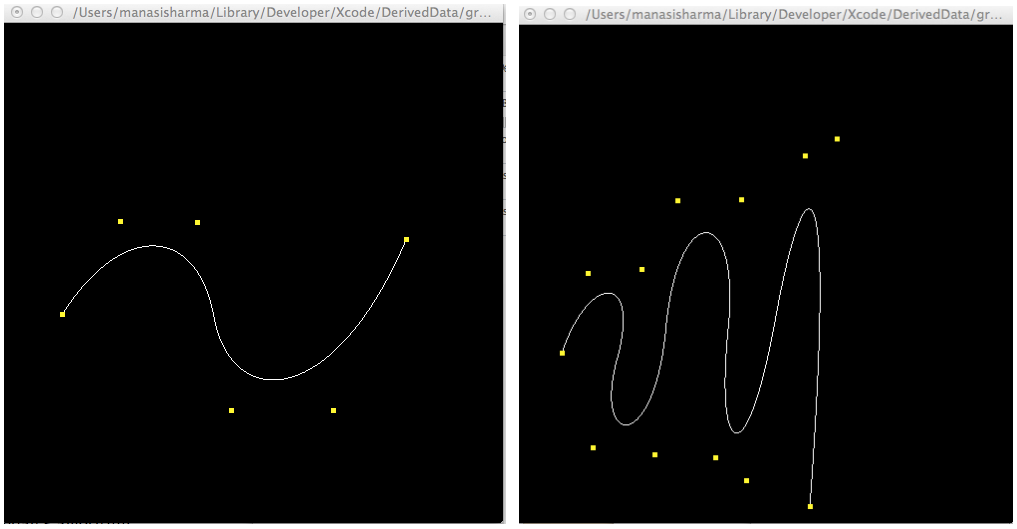
$B_1 = 3 * t * (1 - t)^2$

$B_2 = 3 * t^2 * (1 - t)$

$B_3 = t^3$

As per my program this parameter t varies from 0.0 to 1.0 with 0.01 increments, hence we get 101 points on the curve which are joined by a line using OpenGL API.

Images for the curves drawn using Blending Functions



Advantages

- Numerically more stable
- Easier to compute
- Fewer bumps and wiggles

Disadvantages

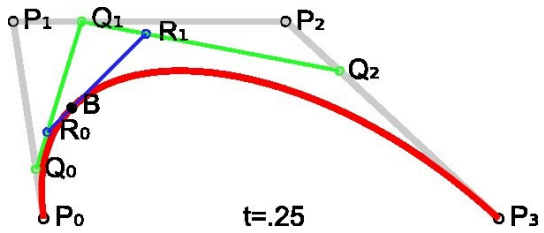
This has the practical disadvantage that any given curve can be represented by infinitely many different control point positions. It also means that, for certain

control point arrangements, the curve collapses to a single point, even though the control points are not all at that point.

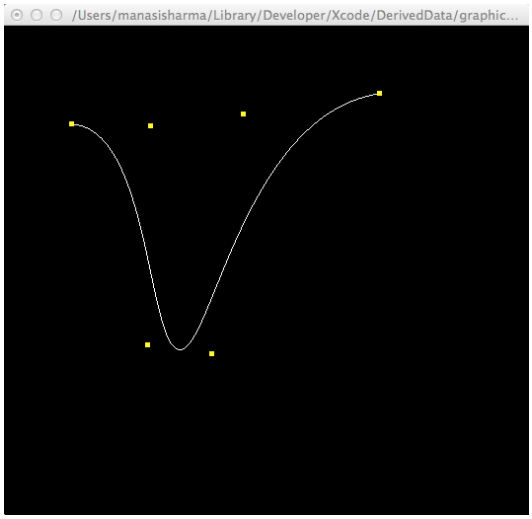
De Casteljau's Algorithm

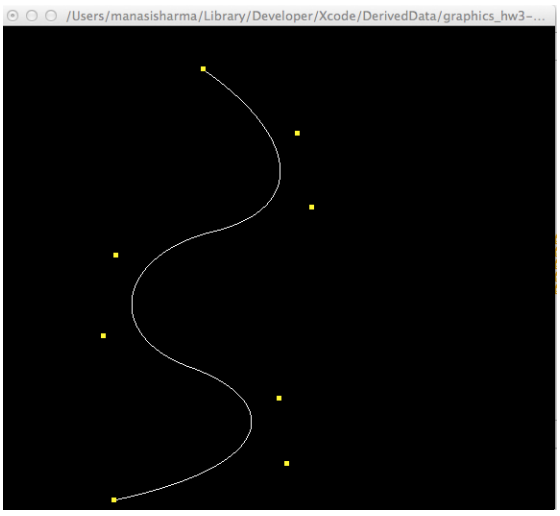
For four control points P_0, P_1, P_2 and P_3, De Casteljau's Algorithm finds points Q_0, Q_1 and Q_2 that describe a linear Bezier curve and then find points R_0 and R1 that describe a quadratic Bezier curve.

All these points are linearly interpolated using the same parameter t defined above.



Images for the curves drawn using De Casteljau's Algorithm





Advantages

Easier to implement : Easily subdivided into parts for drawing.

Disadvantages

- Single piece, moving one point affects whole curve (no local control as in B-splines later)
- Invariant to translations, rotations, scales etc. That is, translating all control points translates entire curve

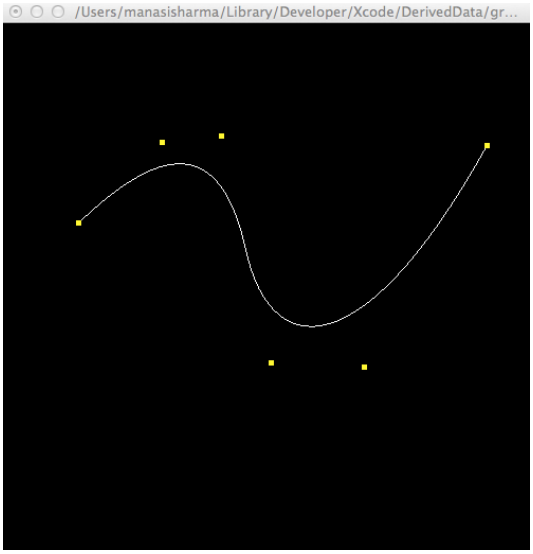
OpenGL API

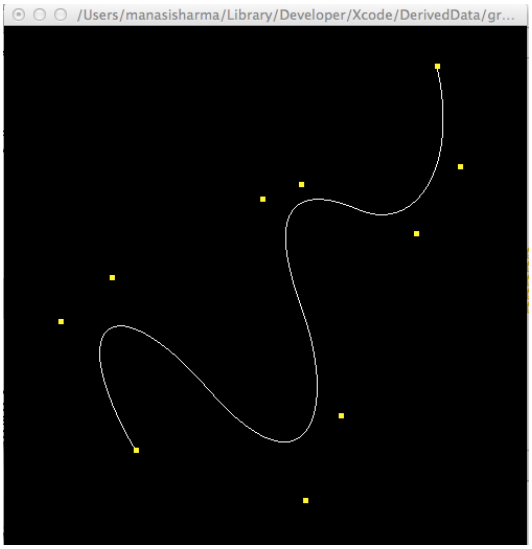
A bezier curve is rendered using OpenGL API's glMap1f() and glEvalCoord1f()

glMap1f functions define a one-dimensional evaluator. It takes as input the parameterization range, the order of the spline and the pointer to the control point array as input and returns a one dimensional evaluator, which is then passed to glEvalCoord1f().

The **glEvalCoord1f** function evaluates enabled one-dimensional maps.

Images for the curves drawn with OpenGL API's





Advantages

- OpenGL provides a procedural model of graphics
- OpenGL provides direct access to the rendering pipeline
- OpenGL is optimized in every imaginable way
- Vendors of every kind of 3D graphics-related hardware support OpenGL

Disadvantages

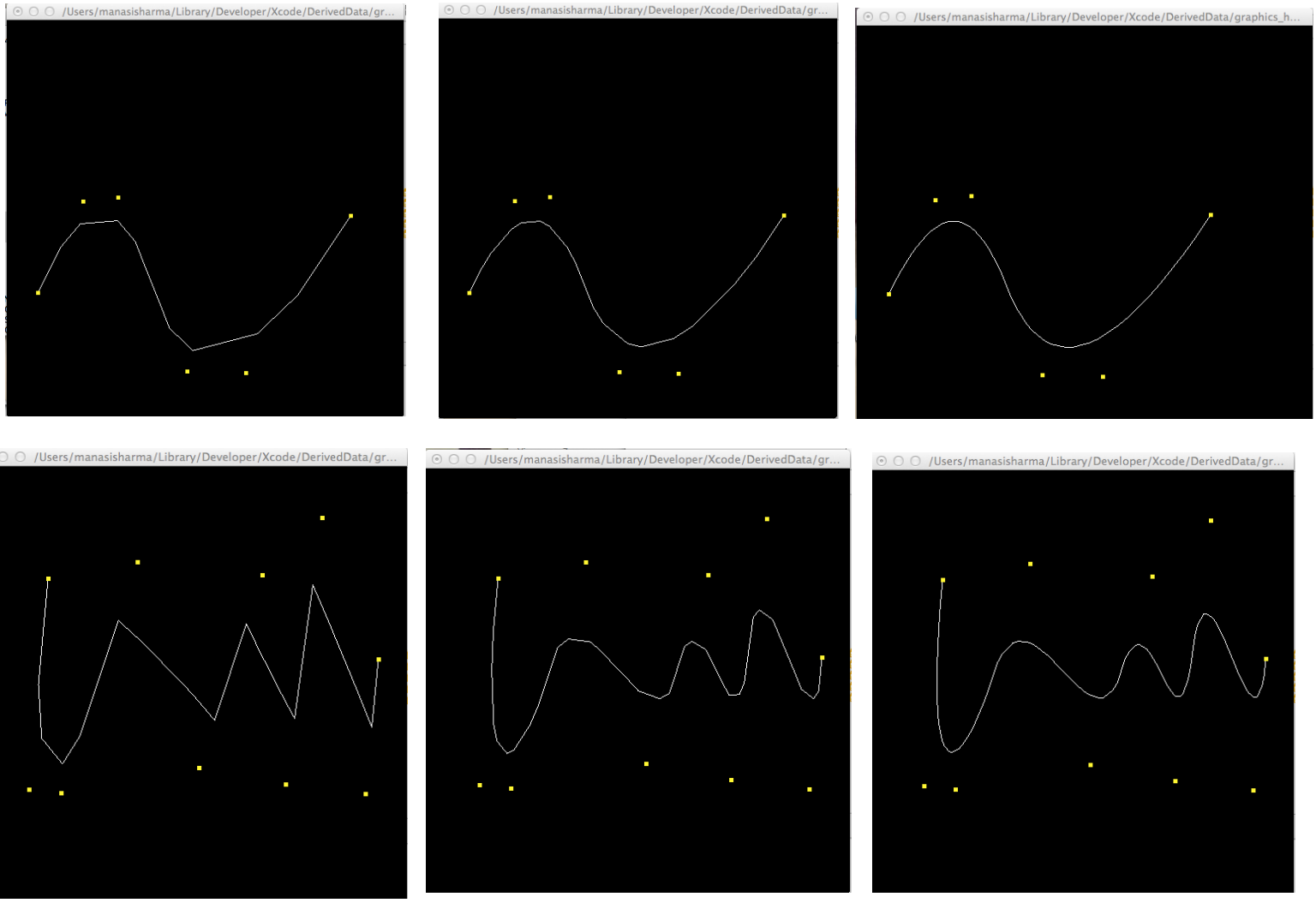
- The strengths of the procedural approach to graphics programming are simultaneously a weakness for many Java programmers
- Many vendors' OpenGL optimizations are meant to decrease hardware choice
- While C interfaces to OpenGL are ubiquitous, Java interfaces aren't yet standardized and aren't widely available
- OpenGL's exposure of the inner details of the rendering process can significantly complicate otherwise simple 3D graphics programs

Sub Division

Applying De Casteljau's Algorithm at every sub division level recursively with parameter t set to 0.5 to get 7 control points (P_0, Q_0, R_0, B, R_1, Q_2 and P_3) from the initial input of 4 control points (P_0, P_1, P_2 and P_3).

The first Sub Division level is taken as an input from the user and levels can further be increased/decreased using up and down arrow keys.

Images of the curves at different subdivision level



Increasing Sub Division Level (left to right)

Advantages

Higher level allows finer control because there are more vertices. Lower level allows control of broader regions, and it's often easier to select the desired vertices since there are fewer of them.

The advantages of this method is that we can move points and add detail to the generated object yet still drive it with its control hull. You can also create multiple high-resolution objects from the same low-resolution one.

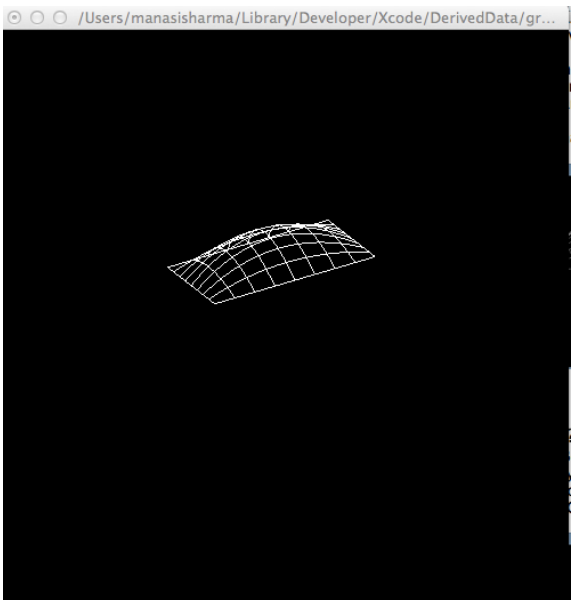
Disadvantages

A disadvantage is that the high-resolution object's geometry can result in heavy scenes and large files. Recursive structure for subdivision surfaces are much harder to maintain.

Bezier Surface Patch using glMap2f

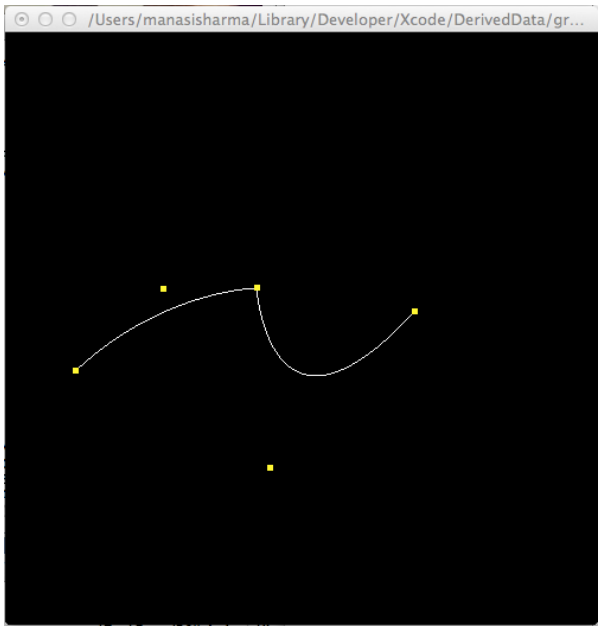
To render a bezier surface patch OpenGL API's glMap2f() and glEvalCoord2f() are used.
The glMap2f() takes parameterization range, order of the spline and pointer to the control point array as input for both u and v axis and returns a two dimensional evaluator, which is then passed to glEvalCoord2f().

Images for the Bezier Surface Patch drawn with initialized control points



What happens if a control point is repeated?

When a control point is repeated, then the resulting Bezier Curve will no longer be a cubic Bezier, it will be a quadratic Bezier curve with only 3 distinct control points. Consider the following scenario where user has input four control points P_0, P_1, P_2 ,P_3, P_4 and P_5. If points P_2 and P_3 are same. Resulting curve is a quadratic Bezier Curve between points P_0, P_1 and P_3.



References

1. Cubic Bezier : http://cubic-bezier.com/#.2,.87,.86,.36
2. Bezier Curve Wiki: http://en.wikipedia.org/wiki/B%C3%A9zier_curve
3. OpenGL Bezier Curve : http://www.glprogramming.com
4. glMap1f: http://msdn.microsoft.com/en-us/library/windows/desktop/ee872051(v=vs.85).aspx