Bezier Curves

- Up to this point we have been dealing with just points, straight lines and simple curves (arcs) since they are easily specified.
- We now want to look at how more complex, "free-form" surfaces can be described.
- These type of surfaces occur commonly in things such as car bodies, ship hulls, blades, etc.

One approach would be to describe the curve as a single high order polynomial.

- in practice, this approach doesn't work well since it usually extremely difficult to find a single polynomial which accurately models the entire surface.
- The polynomial may provide an accurate model in some areas of the surface but it highly inaccurate in other areas.

A much better approach is to use series of low order polynomials to model small sections of the surface and blend these polynomials together to form a description of the entire surface.

To simplify the mathematics, these low order polynomials are typically given in their parametric representations.

Parametric Representation

- A parametric representation is a way of representing a function in terms parameter, for example u, which spans a given interval, typically $[0, 1]$

For example:

A circle in the xy plane with center at (0,0) is defined parametrically by:

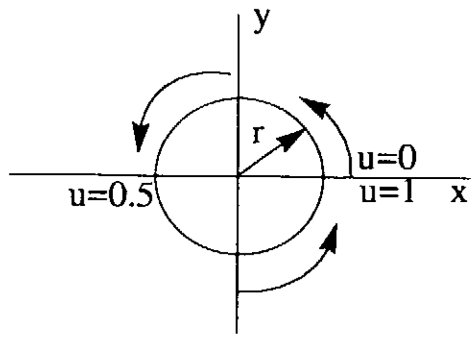
$$x(u) = r\cos(2\pi u)$$

$$y(u) = r\sin(2\pi u)$$

if $u = 0$ then $x(0) = r$ and $y(0) = 0$

if $u = 1$ then $x(1) = r$ and $y(1) = 0$

if $u = 0.5$ then $x(0.5) = -r$ and $y(0.5) = 0$



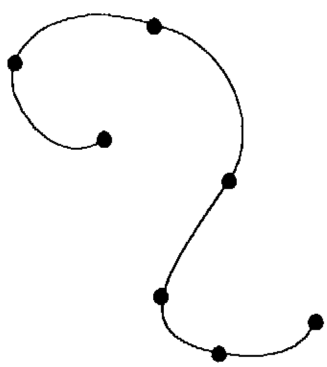
- the circle is produced as u is evaluated from 0 to 1.

Control Points

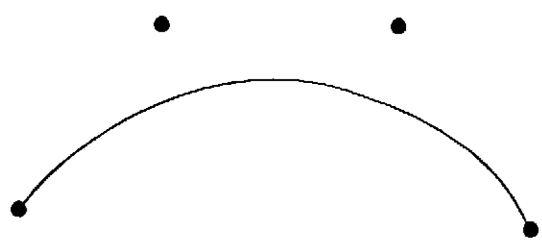
- In general, we want to specify the parametric curve using only a small number of control points

There are two possible ways that the control points define the curve:

- The parametric curve can pass through the control points, in which case it **interpolates** between the points
- Or the parametric curve can pass near the control point, in which case it **approximates** them



Interpolation



Approximation

Bezier Curves

- The particular class parametric curves that we are interested in are the Bezier Curves
- This is a group of approximating polynomial curves based on the Bezier Polynomial which were developed by Pierre Bezier a Renault design engineer.

Given, $n+1$ control points designated by the vectors

$$\vec{p_k} = (x_k, y_k, z_k) \qquad k = 0, \ldots, n$$

The order n Bezier curve $P(u)$ is given by

$$P(u) = \sum_{k=0}^{n} \vec{p_k} B_{k,n}(u)$$

- $B_{k,n}(u)$ are called **blending functions** (or basis functions) since they blend the control points together at each particular value of u and they are given by:

$$B_{k,n}(u) = C(n,k)\, u^k (1-u)^{n-k}$$

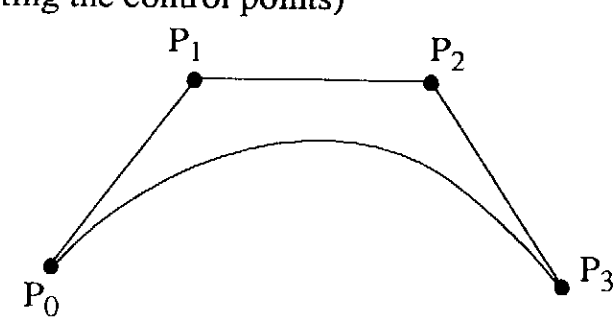
where $C(n,k) = \binom{n}{k} = \dfrac{n!}{k!\,(n-k)!}$

Properties of Bezier Curves:

1. The degree of the bezier curve is one less than the number of control point defining it.

   i.e. $n = \#\,\text{control points} - 1$

2. The curve generally follows the shape of the defining polygon (i.e. the polygon formed by connecting the control points)



3. The curve is contained within the convex hull of the defining polygon (i.e. it has the same curvature of as the defining polygon)

4. The first and last endpoints of the curve are coincident with the endpoint of the defining polygon.

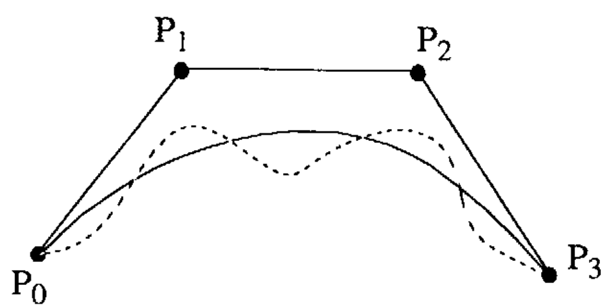
$$P(u)\big|_{u=0} = \vec{p}_0$$

and

$$P(u)\big|_{u=1} = \vec{p}_k$$

5. The tangent vectors at the endpoints of the curve have the same direction as the first and last polygon segments, respectively. (i.e. the slope of the curve at the endpoints is the same as the slope of the respective polygon segment)

6. The curve exhibits the variation diminishing property. Basically, the curve does not oscillate about any straight line more often than the defining polygon. (i.e. the shape of the curve follows the shape of the defining polygon - the curve can follow the solid line but it can't follow the dotted line)



Example:

- Find Bezier curve which is given by the following control points

$$\vec{P_0} = (0,0)\ ,\ \vec{P_1} = (1,1)\ ,\ \vec{P_2} = (2,1)\ , \text{and}\ \vec{P_3} = (3,0)$$

Step 1: Determine the order of the Bezier curve

Since there are 4 control point, $n = 4 - 1 = 3$

Step 2: Calculate the blending functions $B_{k,n}(u)$ for $k = 0, \ldots, n$

For $k = 0$, $B_{0,3}(u) = \frac{3!}{0!(3-0)!}u^0(1-u)^{3-0} = (1-u)^3$

$k = 1$, $B_{1,3}(u) = \frac{3!}{1!(3-1)!}u^1(1-u)^{3-1} = 3u(1-u)^2$

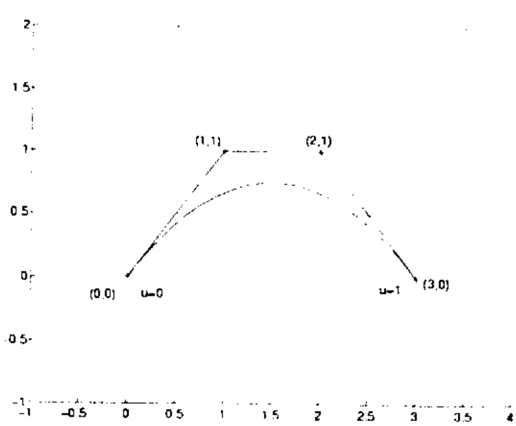
$k = 2$, $B_{2,3}(u) = \frac{3!}{2!(3-2)!}u^2(1-u)^{3-2} = 3u^2(1-u)$

$k = 3$, $B_{3,3}(u) = \frac{3!}{3!(3-3)!}u^3(1-u)^{3-3} = u^3$

Step 3: Substitute the blending functions and control points into the Bezier polynomial function $P(u) = \sum_{k=0}^{n} \vec{p_k}B_{k,n}(u)$

$$\begin{aligned} P(u) &= \vec{p_o}B_{0,3}(u) + \vec{p_1}B_{1,3}(u) + \vec{p_2}B_{2,3}(u) + \vec{p_3}B_{3,3}(u) \\ &= \begin{bmatrix}0 & 0\end{bmatrix}(1-u)^3 + \begin{bmatrix}1 & 1\end{bmatrix}3u(1-u)^2 + \begin{bmatrix}2 & 1\end{bmatrix}3u^2(1-u) + \begin{bmatrix}3 & 0\end{bmatrix}u^3 \end{aligned}$$

Step 4: Evaluate $P(u)$ for $u \in [0, 1]$



In-Class Problems:

Problem 1:

- Generate the Bezier curve given by the following set of control points

$P_0 = [3\ 3\ 0]$

$P_1 = [17\ 6\ 0]$

$P_2 = [7\ 9\ 0]$

$P_3 = [20\ 12\ 0]$

- Evaluate the curve at u=0, 0.2, 0.4, 0.5, 0.8, 1.0
- Sketch the curve, control points, and defining polygon.

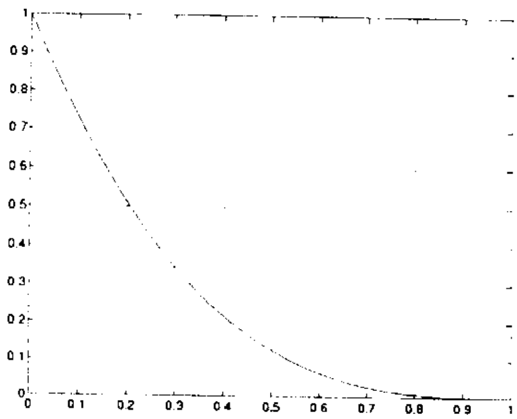
Problem 2:

- Develop the expression P(u) for the bezier curve with n=4.

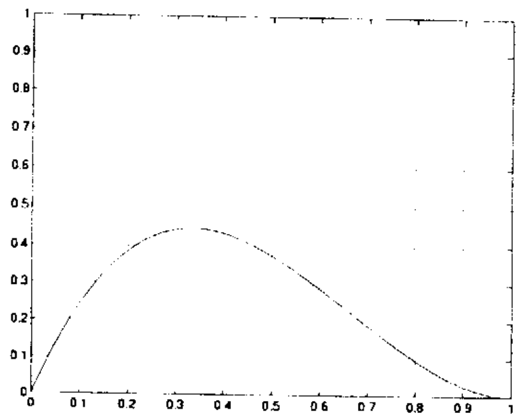
Bezier Curves Cont.

Blending Functions

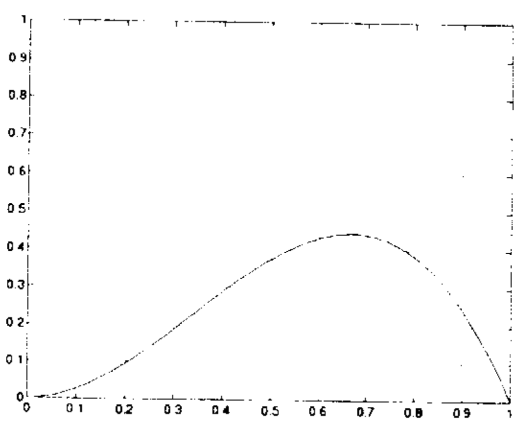
- Last class we introduced the concept of the Bezier blending functions, let's take a closer look at the actual form of these functions.
- Below are the blending function for a 3rd order Bezier curve



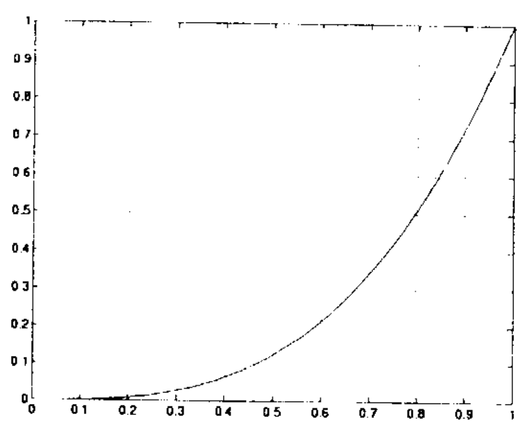
$B_{0,3}(u)$



$B_{1,3}(u)$



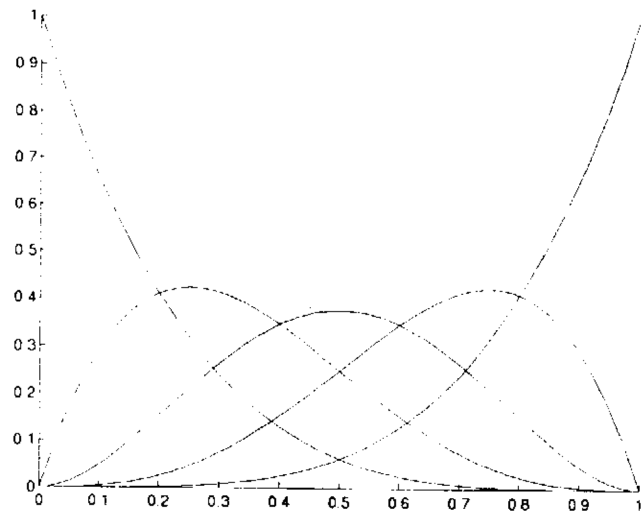
$B_{2,3}(u)$



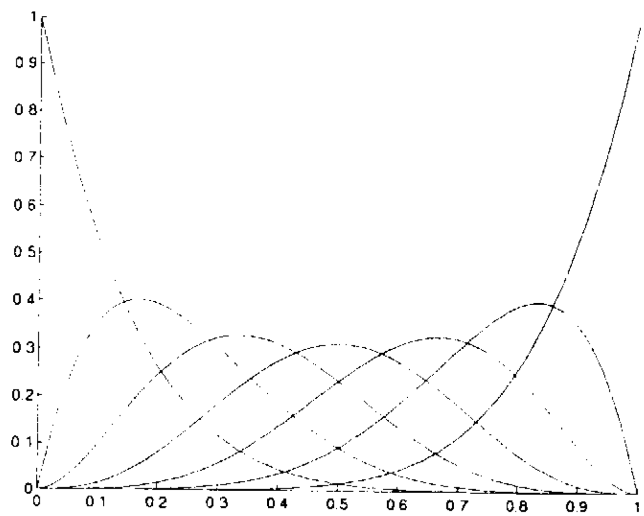
$B_{3,3}(u)$

- These functions determine the weighting of the control point for the various values of u.
- It can been seen that when $u = 0$ only $B_{0,3}(u)$ is non-zero. Hence only $\vec{p}_1$ contributes to the curve for this u. In fact since, $B_{0,3}(u{=}0) = 1$ the curve passes through $\vec{p}_1$ at $u = 0$.
- Similarly at $u = 1$ only $B_{3,3}(u)$ is non-zero, and the curve passes through $\vec{p}_3$.

- As the order of the Bezier curve changes, the number and shape of the blending functions also change
- The following figures show the blending functions for 4^{th} and 6^{th} order Bezier curves.



4^{th} order Bezier basis functions



6^{th} order Bezier basis functions

Bezier Matrix Notation

- Last class we presented the Bezier curve as follows

$$P(u) = \sum_{k=0}^{n} \vec{p}_k B_{k,n}(u)$$

- We can re-write this function in matrix form as:

$$P(u) = [F][G]$$

$$\text{where } [F] = \left[B_{0,n}(u)\ B_{1,n}(u)\ \dots\ B_{n,n}(u)\right]$$

$$[G] = \left[\vec{p}_o\ \vec{p}_1\ \dots\ \vec{p}_n\right]^T$$

- For four control points ($n = 3$), the cubic Bezier curve is given by:

$$P(u) = \left[(1-u)^3\ 3u(1-u)^2\ 2u^2(1-u)\ u^3\right]\begin{bmatrix}\vec{p}_0\\ \vec{p}_1\\ \vec{p}_2\\ \vec{p}_3\end{bmatrix}$$

Or,

$$P(u) = \left[u^3\ u^2\ u\ 1\right]\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 3 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\vec{p}_0\\ \vec{p}_1\\ \vec{p}_2\\ \vec{p}_3\end{bmatrix} = \mathbf{u}^T B\mathbf{p}$$

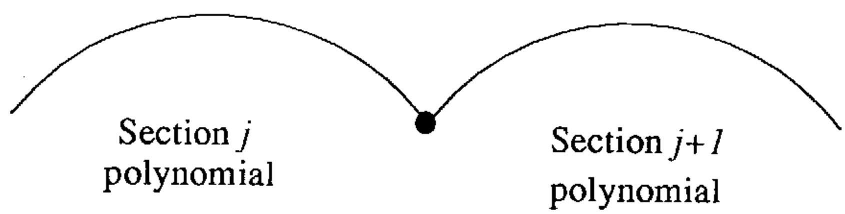
- Note: Matrix B is $(n+1) \times (n+1)$ and specifies the coefficients for the blending functions. A bezier curve can be fully specified by the control point vector and matrix B since the form of ***u*** is known once the order of the curve is known.

Continuity

- Up to this point we have been dealing with just a single parametric curve.
- But, it is difficult to devise a single set of parametric equations which completely defines a "free-form" curve.
- Any curve, though, can be approximated by using different sets of parametric functions over different parts of the curve.
- These approximations are formed with polynomial function (in our case Bezier curves)
- But, we now must ensure that there is a smooth transition from one polynomial section to then next.
- The smoothness criteria is described by the **CONTINUITY** between the sections.

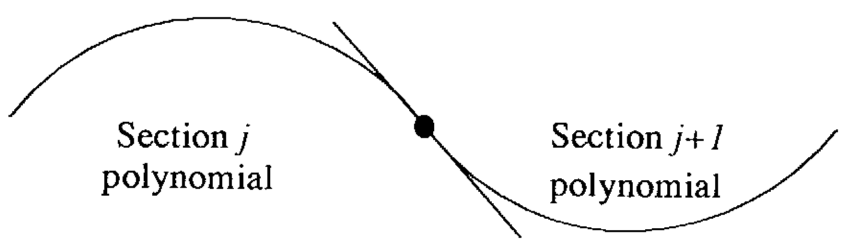
1. 0-order Continuity, C^0

- The curves have a common end/start point (i.e. $P_j(u=1) = P_{j+1}(u=0)$)



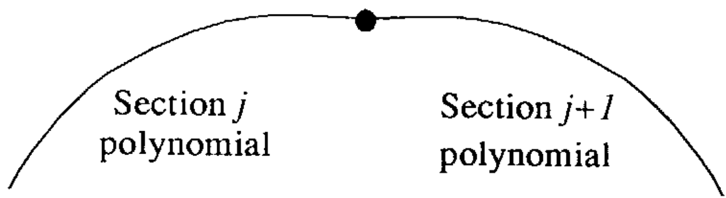
2. 1-order Continuity, C^1

- The tangent lines a (first derivatives of each section) of two adjoining curve sections are equal at the meeting point. (i.e. $P_j'(u=1) = P'_{j+1}(u=0)$)



3. 2-order Continuity, C^2

- The curvature (second derivatives of each section) of two adjoining curve sections are equal at the meeting point. (i.e. $P_j''(u=1) = P''_{j+1}(u=0)$)



Note: $C^n \rightarrow C^{n-1} \rightarrow \ldots \rightarrow C^3 \rightarrow C^2 \rightarrow C^1 \rightarrow C^0$

(i.e if you have C^1 continuity, you also have C^0 continuity)

Derivatives of Bezier Curves

- Obviously, to make use of the concept of continuity with respect to Bezier curves we must know how to take their derivatives.
- From the example given last class, a Bezier curve defined by four control points is given by:

$$P(u) = \vec{p_o}B_{0,3}(u) + \vec{p_1}B_{1,3}(u) + \vec{p_2}B_{2,3}(u) + \vec{p_3}B_{3,3}(u)$$

- the first derivative (w.r.t. u) is:

$$P'(u) = \vec{p_o}B'_{0,3}(u) + \vec{p_1}B'_{1,3}(u) + \vec{p_2}B'_{2,3}(u) + \vec{p_3}B'_{3,3}(u)$$

- From before we have that

$$B_{0,3}(u) = (1-u)^3 \ , B_{1,3}(u) = 3u(1-u)^2 \ , B_{2,3}(u) = 3u^2(1-u) \ , B_{3,3}(u) = u^3$$

$$\therefore \quad B'_{0,3}(u) = -3(1-u)^2$$

$$B'_{1,3}(u) = 3(1-u)^2 - 6u(1-u)$$

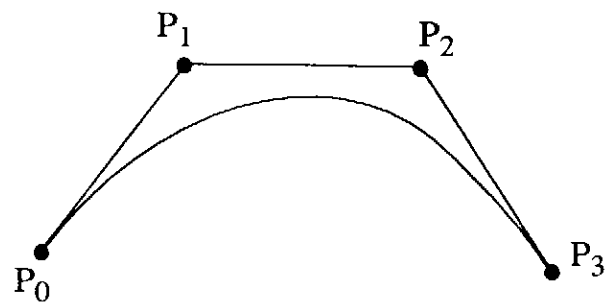
$$B'_{2,3}(u) = 6u(1-u) - 3u^2$$

$$B'_{3,3}(u) = 3u^2$$

- at $u = 0$,

$$P'(u{=}0) = \vec{p}_o(-3) + \vec{p}_1(3) + \vec{p}_2(0) + \vec{p}_3(0) = 3\left(\vec{p}_1 - \vec{p}_0\right)$$

- $\vec{p}_1 - \vec{p}_0$ = the slope of the polygon segment from p_0 to p_1



$\therefore$ the direction of the tangent vector at the curve beginning is the same as the polygon span.

In-Class Problems:

Problem 1:

- Calculate the maximum of each Bezier basis function for n=5.

Problem 2:

- How many control points does a 4^{th} order Bezier curve have?

Problem 3:

- Describe the circumstances for 2 adjoining Bezier curves to have C^0 and C^1 continuity.

Problem 4:

- It is desired to connect 2 Bezier curves with C^1 continuity. The first curve $\vec{P}(t)$ is defined by 4 control points $\vec{B_o}, \vec{B_1}, \vec{B_2}$ and $\vec{B_3}$. The second curve $\vec{Q}(s)$ is defined by the by the 4 control points $\vec{D_o}, \vec{D_1}, \vec{D_2}$ and $\vec{D_3}$.
- Given that

$$\vec{B}_0 = [2\ 1\ 0]^T$$

$$\vec{B}_1 = [3\ 4\ 0]^T$$

$$\vec{B}_2 = [3.5\ 7\ 0]^T$$

$$\vec{B}_3 = [5\ 5\ 0]^T$$

- Find points $\vec{D_0}$ and $\vec{D_1}$.

Review

Orthographic Views:

- Top, Front and Profile views
- Line Precedence (Visible/Hidden/Centreline)
- Generating Missing Views
- Hidden Lines (Space if vertex known, no space if hidden line defines vertex)

Dimensioning:

- Dimension feature once where it shows
- Dimension outside diameters in non-circular view
- Dimensioning of repetitive features

Sectioning:

- Full Sections
- Half Sections
- Revolved/Removed Sections
- Label Section and cutting plane
- Fins, objects on centreline of assembly drawing left unsectioned
- GOAL: Clarity

3-D Viewing Formats:

- Isometric - all axis equal angles apart
- Oblique - cabinet/cavalier
- Perspective - 1 point, 2 point, 3 point

Auxiliary Views:

- know how to generate
- (i.e. TS view of an inclined plane)

Assembly Drawings:

Spatial Analysis:

- TL of a line (aux.view/rotation)
- PV of a line
- EV of a plane (aux.view/rotation)
- TS of a plane (aux.view/rotation)
- Slope/Grade/Bearing of a line
- Slope of a plane
- angle between 2 lines
- angle between 2 planes

Tests for:

- Parallelism of 2 lines
- Perpendicularity of 2 lines
- intersection of 2 lines
- Perpendicularity of a line and a plane

Shortest distance:
- between two lines
- between a plane and a point
- between a plane and a line

Constructing a line perpendicular to a plane

Interactions between:
- line and plane
- plane and a plane
- line and a solid
- plane and a solid
- solid and solid

Methods for determining points of intersection:
- auxiliary view method
- cutting plane method

Visibility

Labeling: subscripts, TL, PV, EV, TS, fold lines, etc.

Computer Graphics

- 2-D and 3-D transformation matrices
  - Translation, scaling, rotation, reflection
- Composite transforms
- Performing transforms about arbitrary points/lines/planes
- Bezier Curves
  - developing curves from set of control points
  - Continuity conditions
  - Matrix notation for Bezier function