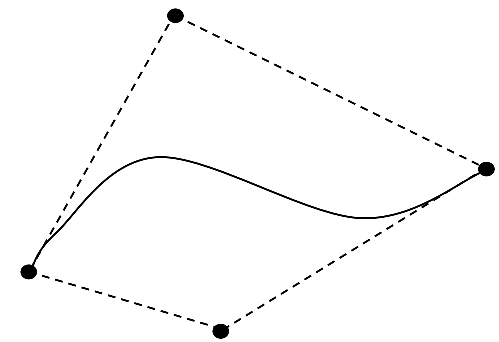
Beizer Curve

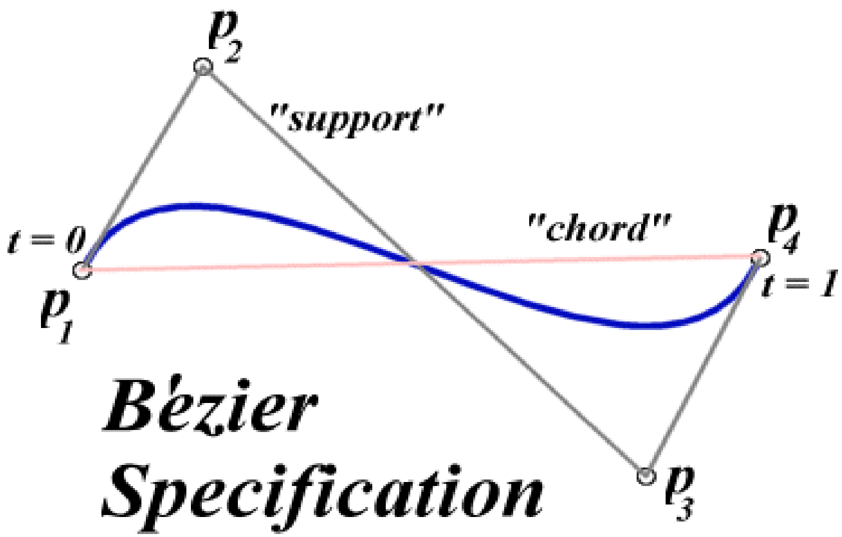
Bézier Curves

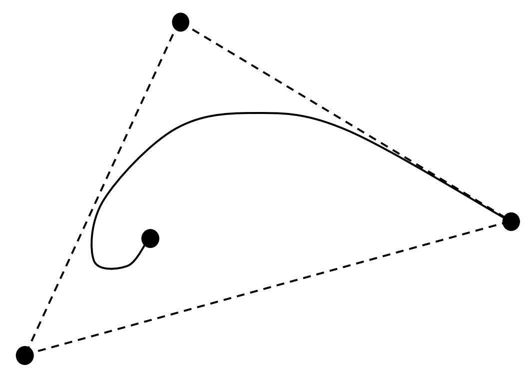
- Hermite cubic curves are difficult to model – need to specify point and gradient.
- More intuitive to only specify points.
- Pierre Bézier (an engineer at Renault) specified 2 endpoints and 2 additional control points to specify the gradient at the endpoints.
- Can be derived from Hermite matrix:
  - Two end control points specify tangent

Bézier Curves

Note the Convex Hull has been shown as a dashed line – used as a bounding extent for intersection purposes.







Bézier Matrix

- The cubic form is the most popular
  $X(t) = t^T M_B q$ (M_B is the Bézier matrix)
- With *n=4* and *r=0,1,2,3* *we get:*

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

- Similarly for Y(t) and Z(t)

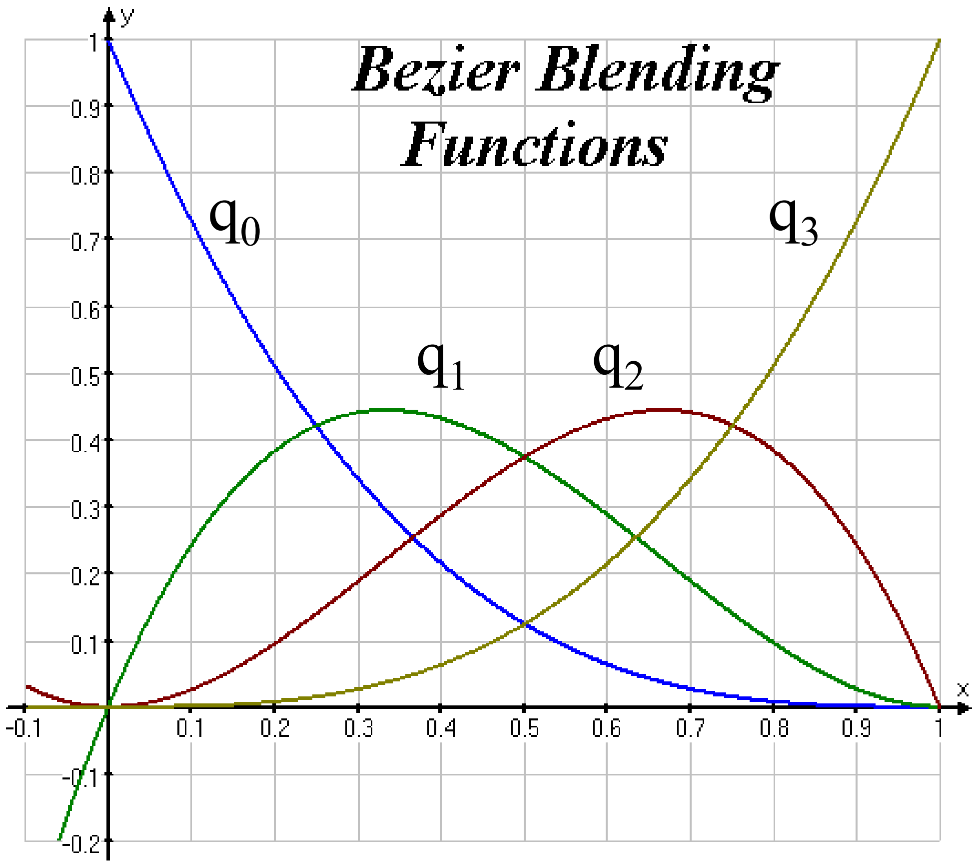
Bézier blending functions

This is how they look –

The functions sum to 1 at any point along the curve.

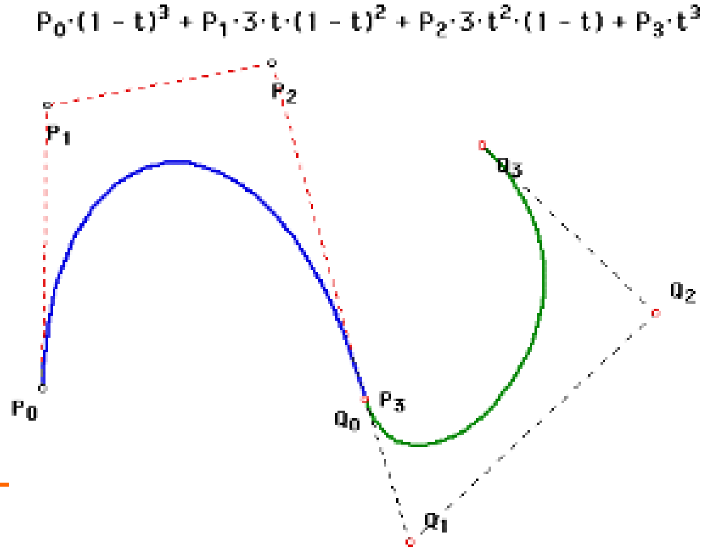
Endpoints have full weight

The weights of each function is clear and the labels show the control



Joining Bezier Curves

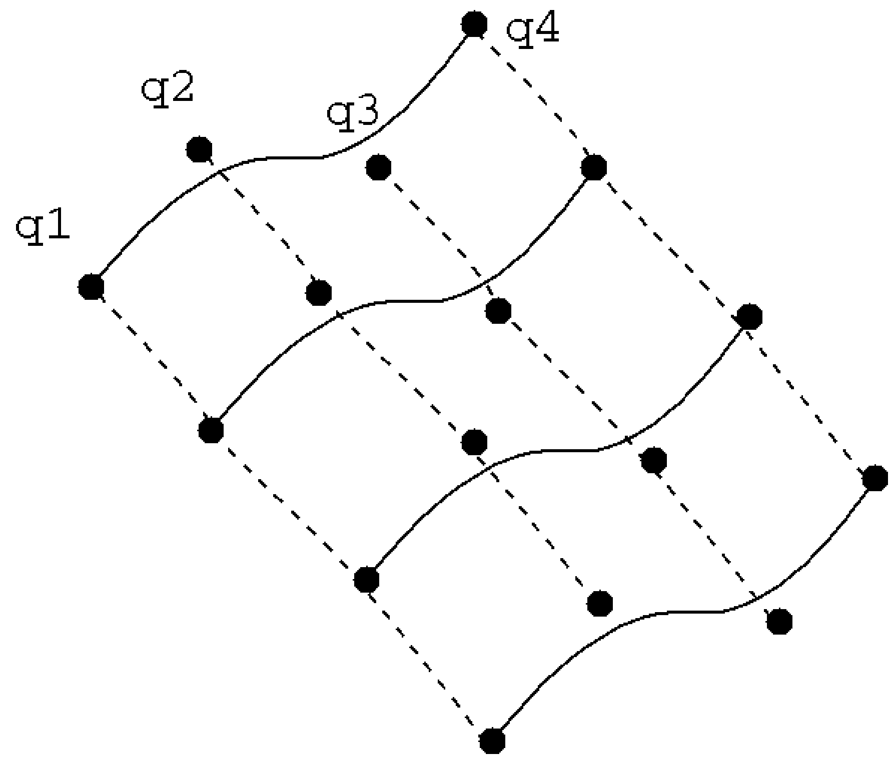
- G^1 continuity is provided at the endpoint when $P_2 - P_3 = k\,(Q_1 - Q_0)$
- if $k=1$, C^1 continuity is obtained



Bicubic patches

- The concept of parametric curves can be extended to surfaces
- The cubic parametric curve is in the form of $Q(t)=\boldsymbol{t}^T\boldsymbol{M}\,\boldsymbol{q}$ where $\boldsymbol{q}=(q_1,q_2,q_3,q_4)$: q_i control points, $\boldsymbol{M}$ is the basis matrix (Hermite or Bezier,…), $\boldsymbol{t}^T=(t^3, t^2, t, 1)$

- Now we assume q_i to vary along a parameter s,
- $Q_i(s,t)=\mathbf{t}^T\mathbf{M}\,[q_1(s),q_2(s),q_3(s),q_4(s)]$
- $q_i(s)$ are themselves cubic curves, we can write them in the form …



Bicubic patches

$$Q(s,t) = t^T.M.(s^T.M.[\mathbf{q}_{11},\mathbf{q}_{12},\mathbf{q}_{13},\mathbf{q}_{14}],\ldots,s^T.M.[\mathbf{q}_{41},\mathbf{q}_{42},\mathbf{q}_{43},\mathbf{q}_{44}])$$

$$= t^T.M.\mathbf{q}.M^T.s$$

where **q** is a 4x4 matrix $\begin{bmatrix} q_{11} & q_{21} & q_{31} & q_{41} \\ q_{12} & q_{22} & q_{32} & q_{42} \\ q_{13} & q_{23} & q_{33} & q_{43} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}$

Each column contains the control points of

q1(s),…,q4(s)

x,y,z computed by

$$x(s,t) = t^T.M.\mathbf{q}_x.M^T.s$$

$$y(s,t) = t^T.M.\mathbf{q}_y.M^T.s$$

$$z(s,t) = t^T.M.\mathbf{q}_z.M^T.s$$

Bézier example

- We compute (x,y,z) by

$$x(s,t) = t^T.M_B.q_x.M_B^T.s$$

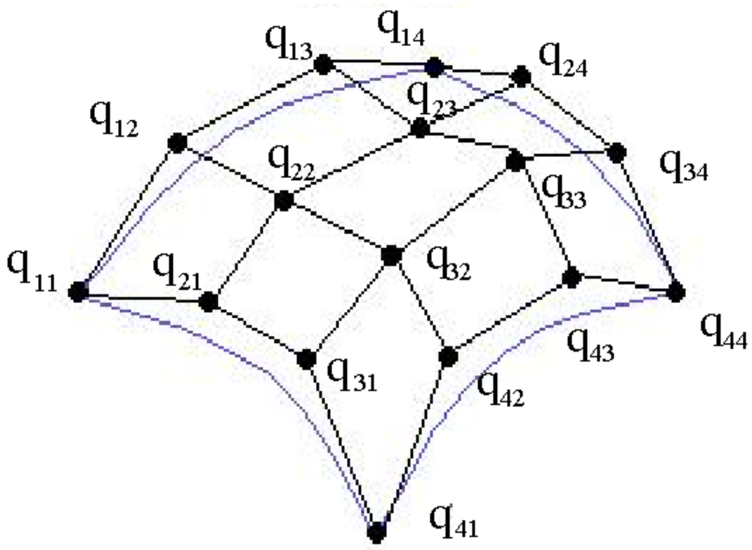
q_x *is* 4×4 *array of x coords*

$$y(s,t) = t^T.M_B.q_y.M_B^T.s$$

q_y *is* 4×4 *array of y coords*

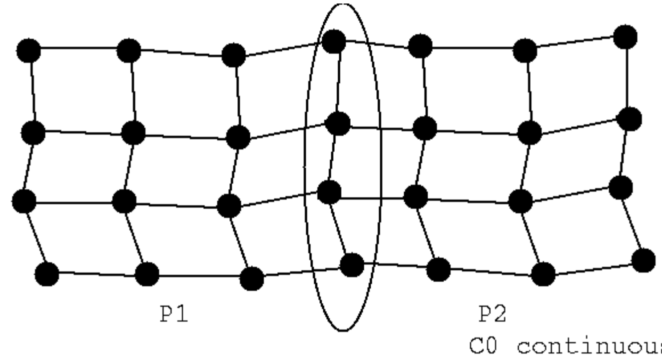
$$z(s,t) = t^T.M_B.q_z.M_B^T.s$$

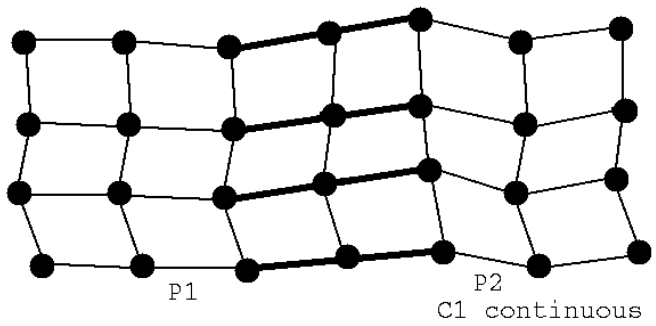
q_z *is* 4×4 *array of z coords*



Continuity of Bicubic patches.

- Hermite and Bézier patches
  - C_0 continuity by sharing 4 control points between patches.
  - C_1 continuity when both sets of control points either side of the edge are collinear with the edge.





Displaying Bicubic patches.

- Need to compute the normals
  - vector cross product of the 2 tangent vectors.
- Need to convert the bicubic patches into a polygon mesh
  - tessellation

Normal Vectors

$$\frac{\partial}{\partial s}Q(s,t)=\frac{\partial}{\partial s}(t^T.M.q.M^T.s)=t^T.M.q.M^T.\frac{\partial}{\partial s}(s)$$

$$=t^T.M.q_x.M^T.[3s^2,2s,1,0]^T$$

$$\frac{\partial}{\partial t}Q(s,t)=\frac{\partial}{\partial t}(t^T.M.q.M^T.s)=\frac{\partial}{\partial t}(t^T).M.q.M^T.s$$

$$=[3t^2,2t,1,0]^T.M.q.M^T.s$$

$$\frac{\partial}{\partial s}Q(s,t)\times\frac{\partial}{\partial t}Q(s,t)=(y_s z_t - y_t z_s, z_s x_t - z_t x_s, x_s y_t - x_t y_s)$$

The surface normal is biquintic (two variables, fifth-degree) polynomial and very expensive
Can use finite difference to reduce the computation

Tessellation

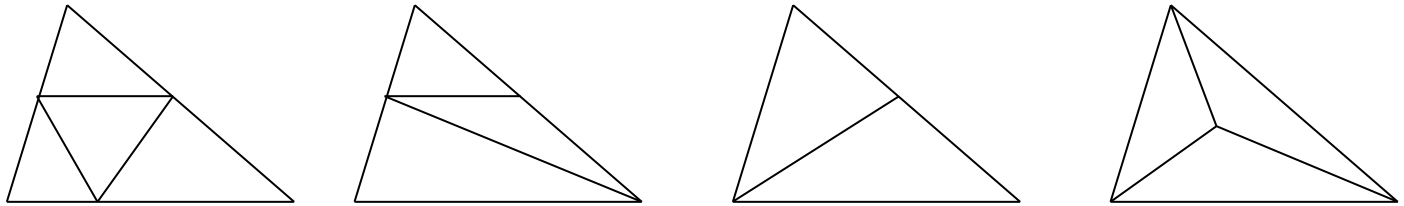
- We need to compute the triangles on the surface
- The simplest way is do uniform tessellation, which samples points uniformly in the parameter space
- Adaptive tessellation – adapt the size of triangles to the shape of the surface
  - i.e., more triangles where the surface bends more
  - On the other hand, for flat areas we do not need many triangles

Adaptive Tessellation

- For every triangle edges, check if each edge is tessellated enough (curveTessEnough())
- If all edges are tessellated enough, check if the whole triangle is tessellated enough as a whole (triTessEnough())
- If one or more of the edges or the triangle's interior is not tessellated enough, then further tessellation is needed

Adaptive Tessellation

- When an edge is not tessellated enough, a point is created halfway between the edge points' uv-values
- New triangles are created and the tessellator is once again called with the new triangles as input

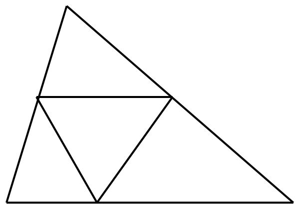


Four cases of further tessellation

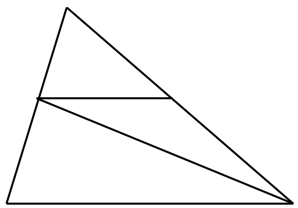
Adaptive Tessellation

AdaptiveTessellate(p,q,r)

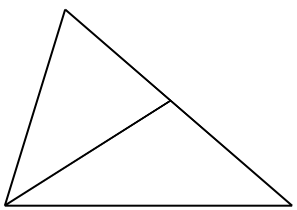
- tessPQ=not curveTessEnough(p,q)
- tessQR=not curveTessEnough(q,r)
- tessRP=not curveTessEnough(r,p)
- If (tessPQ and tessQR and tessRP)
  - AdaptiveTessellate(p,(p+q)/2,(p+r)/2);
  - AdaptiveTessellate(q,(q+r)/2,(q+p)/2);
  - AdaptiveTessellate(r,(r+p)/2,(r+q)/2);
  - AdaptiveTessellate((p+q)/2,(q+r)/2,(r+p)/2);



- else if (tessPQ and tessQR)
  - AdaptiveTessellate(p,(p+q)/2,r);
  - AdaptiveTessellate((p+q)/2,(q+r)/2,r);
  - AdaptiveTessellate((p+q)/2,q,(q+r)/2);

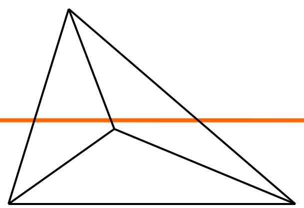


- else if (tessPQ)
  - AdaptiveTessellate(p,(p+q)/2,r);
  - AdaptiveTessellate(q,r,(p+q)/2);



- Else if (not triTessEnough(p,q,r))

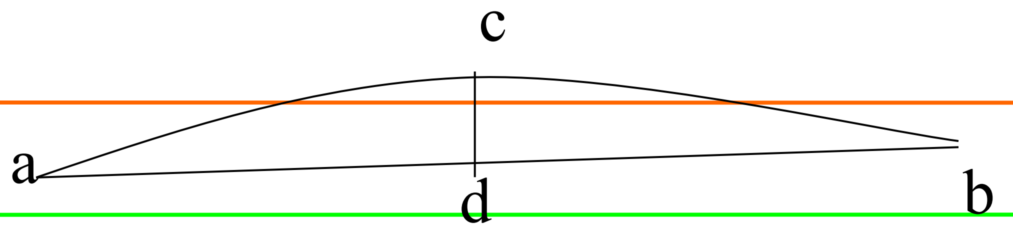
  AdaptiveTessellate((p+q+r)/3,p,q);
  AdaptiveTessellate((p+q+r)/3,q,r);
  AdaptiveTessellate((p+q+r)/3,r,p);

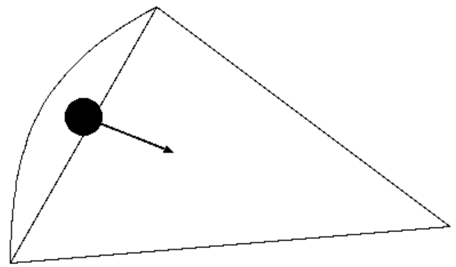


end;

curveTessEnough

- Say you are to judge whether **ab** needs tessellation
- You can compute the midpoint c, and compute its distance *l* from **ab**
- Check if *l*/||**a-b**|| is under a threshold
- Can do something similar for triTessEnough
  - Sample at the mass center and calculate its distance from the triangle





Other factors to evaluate

- Inside the view frustum
- Front facing
- Occupying a large area in screen space
- Close to the sillhouette of the object
- Illuminated by a significant amount of specular lighting