Properties of Bezier Curves 1

Suppose you have a 2D Bezier curve defined by the control points b_0, b_1, and b_2. Which of the follow is NOT true of that curve?

- (a) It will be a cubic polynomial curve.
- (b) It will be completely contained within the triangle formed by b_0, b_1, and b_2.
- (c) Changing the order of the control points to be b_2, b_1, and b_0 will result in the same curve.
- (d) It will pass through the points b_0 and b_2.

Save & Grade Save only New variant

See the lecture Bezier Curves:

https://github.com/illinois-cs415/illinois-cs415.github.io/raw/main/img/slides/415-GeometryBezier1.pdf

Bezier Patches 1

Suppose we have a cubic Bezier patch in 3D space defined by the control net of points $b_{0,0}$,..., $b_{3,3}$. Which set of points does the patch interpolate?

- (a) All the control points on the boundary of the patch
- (b) The corner points $b_{0,0}$ $b_{3,0}$ $b_{0,3}$ and $b_{3,3}$
- (c) A Bezier patch does not interpolate any of the points in the control net.
- (d) All the boundary points of the patch

Save & Grade | Save only | New variant

See the lecture Bezier Patches page 16

https://github.com/illinois-cs415/illinois-cs415.github.io/raw/main/img/slides/415-GeometryBezier2.pdf

Bezier Patches 2

Suppose we have a quadratic Bezier patch in 3D. We wish to render the patch using a set of triangles. What is the maximum resolution at which the patch can be rendered in terms of the number of triangles?

- (a) We can render the patch using a grid with 16 vertices, generating 18 triangles.
- (b) We can render the patch using a grid with 16 vertices, generating 32 triangles.
- (c) We can render the patch using a grid with 9 vertices, generating 8 triangles.
- (d) We can render the patch using a grid with 9 vertices, generating 18 triangles.
- (e) A Bezier patch can only be used to generate quadrilaterals.
- (f) We can render the patch using as many triangles as we wish; there is no inherent limit to the resolution at which you can approximate a Bezier patch.

Save & Grade Save only New variant

We can generate any number of points on the patch the we wish...imagine we want to generate a grid of n by n points we simply generate a point B(u,v) where u and v go from 0 to 1 using a stepsize of 1/(n-1). This grid consists of quadrilateral cells, each of which can be broken into 2 triangles. So...how many triangles can we generate?

Mesh Simplification and Edge Collapses

Assume you have a closed triangulated mesh surface mesh with 2973 triangles. After you perform 744 edge collapses how many triangles remain in the mesh?

Answer = integer

Save & Grade | Save only | New variant

How many triangles are removed by an edge collapse?
See the lecture LOD Generation and Mesh Simplification pages 13-14

https://github.com/illinois-cs415/illinois-cs415.github.io/raw/main/img/slides/415-LOD.pdf

Points

A cubic Bézier curve is a parametric polynomial curve given by:

$$X(t) = (1-t)^3 b_0 + 3(1-t)^2 t b_1 + 3(1-t)t^2 b_2 + t^3 b_3$$

where b_i are the control points. Suppose the control points are $b_0 = (-1, 0)$, $b_1 = (0, 1)$, $b_2 = (0, -1)$, and $b_3 = (1, 0)$. Use de Casteljau's algorithm to find the coordinates of $X(0.25)$, and check it with the polynomial equation above.

$X(0.25) =$ [ ] [ ]

Save & Grade Save only New variant

See the lecture Bezier Curves pages 19-21, example on page 21

Bilinear Patch

Suppose our control points are

$b_{0,0} = (0,0,0), b_{1,0} = (1,0,0), b_{0,1} = (0,1,0), b_{1,1} = (1,1,1)$

What point on the patch is found by evaluating X(0.95, 0.55)

[___ ___ ___] ?

Save & Grade | Save only | New variant

$$[1-0.95 \quad 0.95] \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} \begin{bmatrix} 1-0.55 \\ 0.55 \end{bmatrix}$$

actually a row vector

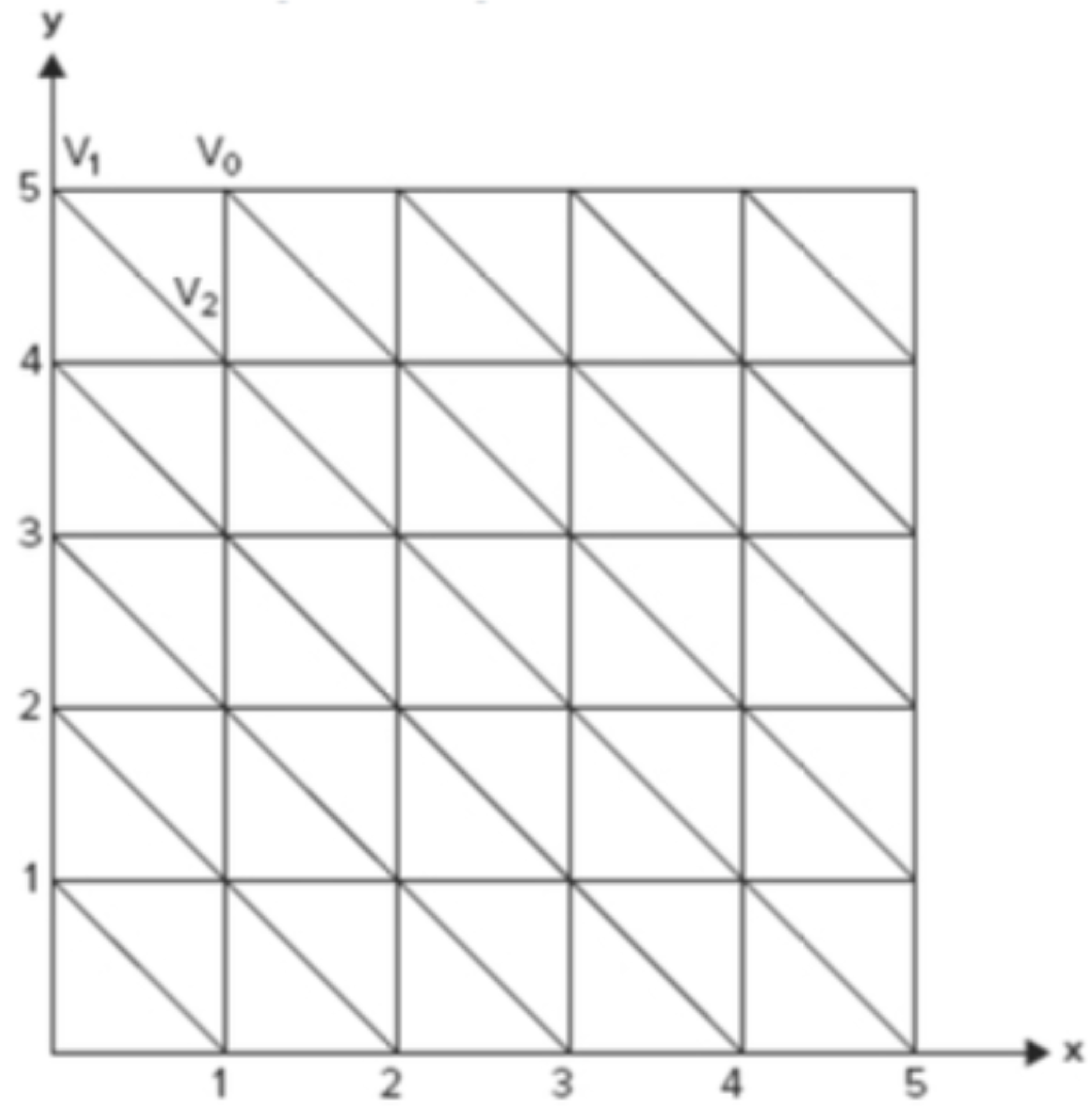
$$\begin{bmatrix} 0.05\,(0,0,0) + 0.95\,(1,0,0) \\ 0.05\,(0,1,0) + 0.95\,(1,1,1) \end{bmatrix}^T \begin{bmatrix} 0.45 \\ 0.55 \end{bmatrix}$$

$$\begin{bmatrix} (0.95,0,0) \\ (0.95,1,0.95) \end{bmatrix}^T \begin{bmatrix} 0.45 \\ 0.55 \end{bmatrix}$$

$$0.45\,(0.95,0,0) + 0.55\,(0.95,1,0.95)$$

. . . . and you can finish

Indexed Face Format for Meshes



Imagine we generate a rectangular mesh with a boundary using an $n+1$ by $n+1$ grid of vertices.
In the example above, it is using $n = 5$. We have $(5+1)*(5+1)$ grid of vertices and $5*5*2$ triangles.
Triangles are generated as shown, cutting each grid square into 2 triangles.

The mesh is stored using an **indexed face format** (like an OBJ file, for example).
Assuming:
1. each vertex is specified by **3 floating point coordinates** (the image doesn't show z-axis)
2. floating point numbers use **4 bytes** of space
3. integer indices use **2 bytes** of space.

Suppose you need to store a mesh with $n = 15$. How much space is used by the mesh?

bytes = integer

For $n=15$
16^2 vertices each needing 3 coords $\times$ 4 bytes
15^2 quads in grid... each split into 2 triangles

each Δ has 3 indices

Quadric Error Metric 1

We saw that the squared distance from a point p to a plane q in 3D space can be found in the following manner:

$$p = (x, y, z, 1)^T, \quad q = (a, b, c, d)^T$$

$$dist(q, p)^2 = (q^T p)^2 = p^T (q q^T) p =: p^T Q_q p$$

$$Q_q = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Suppose we wish to find the sum of the squared distances from p to a set of N planes $q_1,\ldots,q_n$ and we already have access to the matrices $Q_1,\ldots,Q_n$. How many scalar multiplications would need to be performed to find the sum? Disregard any addition operations.

- ○ (a) 20
- ○ (b) 32N
- ○ (c) 16N
- ○ (d) 32
- ○ (e) 20N

Save & Grade | Save only | New variant

Can sum all n Q_i matrices to a single matrix using only addition

$$p^T Q_q p = [x, y, z, 1] \underbrace{\begin{bmatrix} q_{00} & \cdots & q_{03} \\ \vdots & & \vdots \\ q_{30} & \cdots & q_{33} \end{bmatrix}}_{\to 16 \text{ mults}} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\to 4 \text{ mults}}$$

The De Casteljau Algorithm

The function `lerp(t,a,b)` can be used to linearly interpolate between two points in space. If we use the De Casteljau algorithm to evaluate 100 points on a cubic Bezier curve, how many times will the `lerp(t,a,b)` function be called?

Answer = integer

Save & Grade | Save only | New variant

See the lecture Bezier Curves pages 19-21, example on page 21

Evaluating Bezier Curves

Suppose we have a 2D quadratic Bezier curve with control points

$b_0 = (-3, -2)$
$b_1 = (0, -1)$
$b_2 = (2, 4)$

Recall that the Berstein polynomials for a quadratic Bezier curve are:

$$B_0^2(t) = (1-t)^2$$
$$B_1^2(t) = (1-t)2t$$
$$B_2^2(t) = t^2$$

What is the y coordinate of the point on the curve at $t = 0.1$
Your answer is expected to be correct to 2 significant digits.

$y =$ number (2 significant figures)

Save & Grade | Save only | New variant

Using only y coords we have
$(1-0.1)^2(-2) + (1-0.1)(2\times0.1)(-1) + (0.1)^2 4$