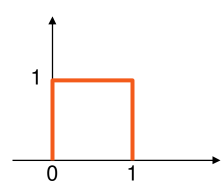
B-splines and Subdivision

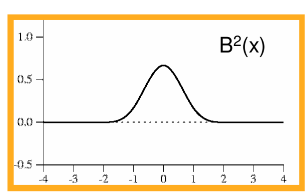
B-Splines (Uniform)

Through repeated integration

Box function $$B_0(x) = \begin{cases} 1 & \text{if } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$



$$B_{m+1}(x) = \int_0^1 B_m(x-t)\,dt$$



Convolution

- definition:

$$g \otimes h(x) = \int g(t)h(x-t)\,dt$$

- translation:

$$g(.-i) \otimes h(.-j)(x) = g \otimes h(x-i-j)$$

- dilation:

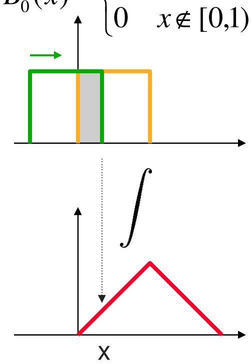
$$g(2\cdot) \otimes h(2\cdot)(x) = \frac{1}{2} g \otimes h(2x)$$

B-Splines

Repeated convolution
with box function

$$\boxed{B_{m+1}(x) = B_0 \otimes B_m(x)}$$

$$B_0(x) = \begin{cases} 1 & x \in [0,1) \\ 0 & x \notin [0,1) \end{cases}$$



B-Splines

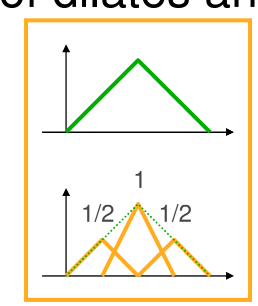
Obvious properties

- piecewise polynomial
- unit integral: $\int_{-\infty}^{+\infty} B_m(x)dx = 1$
- non-negative: $B_m(x) \geq 0$
- partition of unity: $\sum_i B_m(x-i) = 1$
- support: $B_m(x) \neq 0,\ x \in [0, m]$

Refinability of B-Splines

B-Spline refinement equation

- a B-spline can be written as a linear combination of dilates and translates of itself
- example
  - linear B-spline
  - and all others…



Refinability of B-Splines

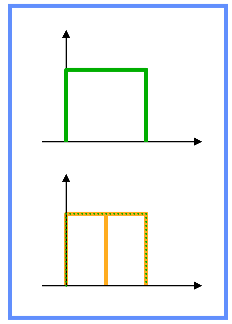
Refinement equation for B-splines

- take advantage of box refinement

$$B_0(x) = B_0(2x) + B_0(2x-1)$$

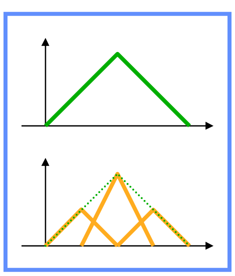
$$B_1(x) = \frac{1}{2}\left(B_1(2x) + 2B_1(2x-1) + B_1(2x-2)\right)$$

$$B_n(x) = \frac{1}{2^n}\sum_{k=0}^{n+1}\binom{n+1}{k}B_n\left(2x-k\right)$$

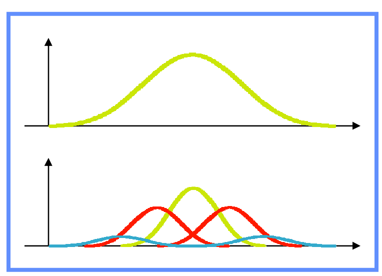


B-Spline Refinement

Examples



1/2(1, 2, 1)



1/8(1, 4, 6, 4, 1)

Spline Curves

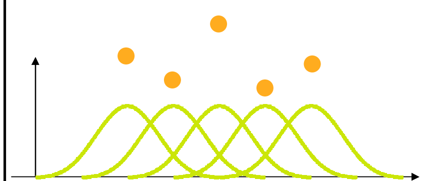
Sum of B-splines

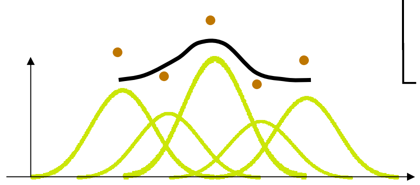
- curve as linear combination

$$\gamma(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_i B(t-i)\,p_i = \mathbf{B}(t)\mathbf{p}$$

control points

$$\mathbf{B}(t) = [\ldots\ \ B(t+1)\ \ B(t)\ \ B(t-1)\ \ \ldots], \quad \mathbf{p} = \begin{bmatrix} \vdots \\ p_0 \\ p_1 \\ \vdots \end{bmatrix}$$

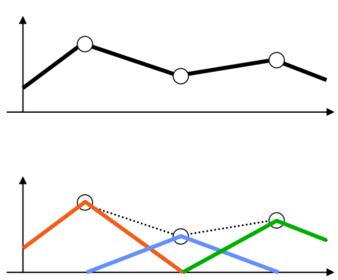


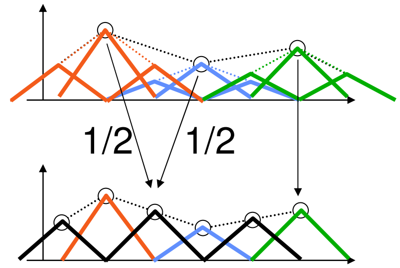


Spline Curves

Refine each B-spline in sum

- example: linear B-spline





Spline Curves

Refinement for curves

- refine each B-spline in sum

$$\gamma(t) = \sum_i p_i B(t-i) = \sum_i p_i \left(\sum_k s_k B\big(2(t-i)-k\big) \right)$$

$$= \underbrace{\boxed{\sum_j B(2t-j)}}_{\text{refined bases}} \underbrace{\boxed{\sum_i s_{j-2i} p_i}}_{\text{refinement of control points}}$$

Refinement of Curves

Linear operation on control points

$$\gamma(t) = \sum_i p_i B(t-i) \qquad \boxed{\gamma(t) = B(t)p}$$

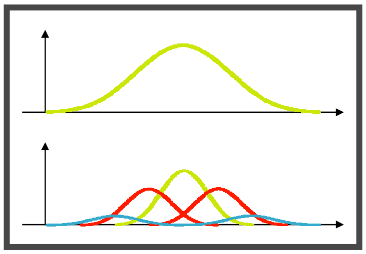
$$B(t) = \sum_k s_k B(2t-k) \qquad \boxed{B(t) = B(2t)S}$$

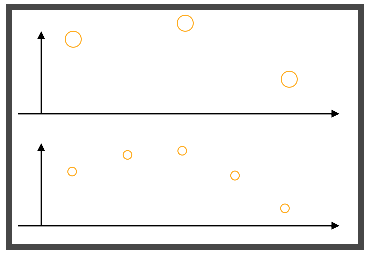
$$\boxed{\gamma(t) = B(t)p = B(2t)Sp}$$

Refinement of Curves

Bases and control points

$$B(t) = B(2t)S \qquad p^1 = Sp^0$$





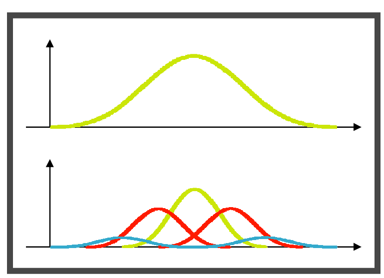
Subdivision Operator

Example

- cubic splines

$$\mathbf{S} = \frac{1}{8}\begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \cdots & 1 & 0 & 0 & 0 & \cdots \\ \cdots & 4 & 0 & 0 & 0 & \cdots \\ \cdots & 6 & \mathbf{1} & 0 & 0 & \cdots \\ \cdots & 4 & \mathbf{4} & 0 & 0 & \cdots \\ \cdots & 1 & \mathbf{6} & 1 & 0 & \cdots \\ \cdots & 0 & \mathbf{4} & 4 & 0 & \cdots \\ \cdots & 0 & \mathbf{1} & 6 & 1 & \cdots \\ \cdots & 0 & 0 & 4 & 4 & \cdots \\ \cdots & 0 & 0 & 1 & 6 & \cdots \\ \cdots & 0 & 0 & 0 & 4 & \cdots \\ \cdots & 0 & 0 & 0 & 1 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$B(t) = B(2t)S$$

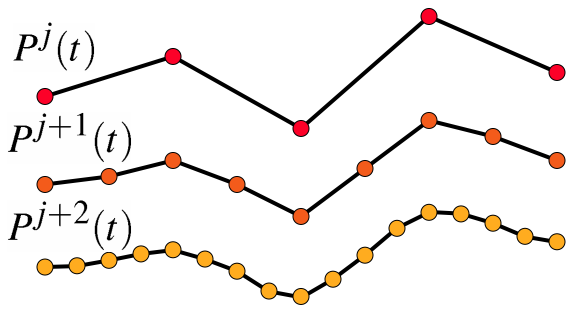


1/8 (1, 4, 6, 4, 1)

Subdivision

Apply subdivision to control points

- draw successive control polygons rather than curve itself



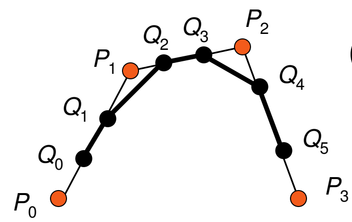
$$p^{j+1} = Sp^j$$

Summary so far

Splines through refinement

- B-splines satisfy refinement eq.
- basis refinement corresponds to control point refinement
- instead of drawing curve, draw control polygon
- subdivision is refinement of control polygon

Chaikin's Algorithm 1974



Corner cutting

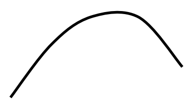
$$Q_0 = \frac{1}{4}P_0 + \frac{3}{4}P_1$$
$$Q_1 = \frac{3}{4}P_0 + \frac{1}{4}P_1$$

$$Q_2 = \frac{1}{4}P_1 + \frac{3}{4}P_2$$
$$Q_3 = \frac{3}{4}P_1 + \frac{1}{4}P_2$$

$$Q_4 = \frac{1}{4}P_2 + \frac{3}{4}P_3$$
$$Q_5 = \frac{3}{4}P_2 + \frac{1}{4}P_3$$

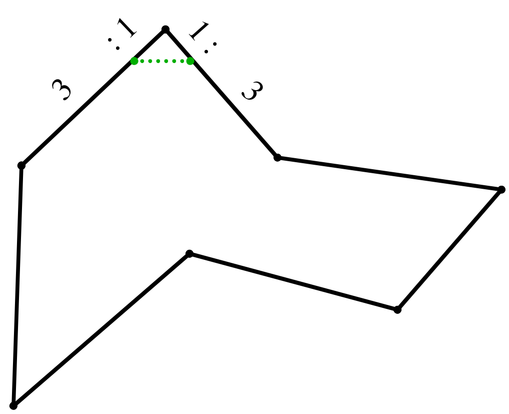
Apply Iterated Function System

$$Q_{2i} = \frac{1}{4}P_i + \frac{3}{4}P_{i+1}$$
$$Q_{2i+1} = \frac{3}{4}P_i + \frac{1}{4}P_{i+1}$$

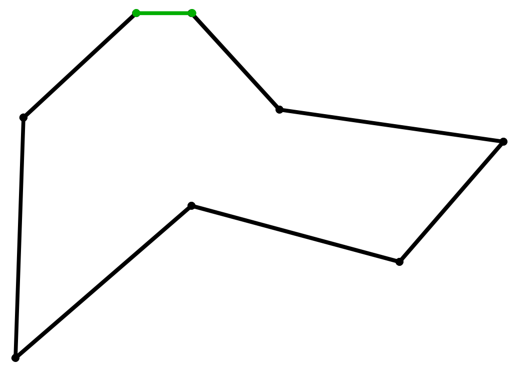


Limit Curve Surface

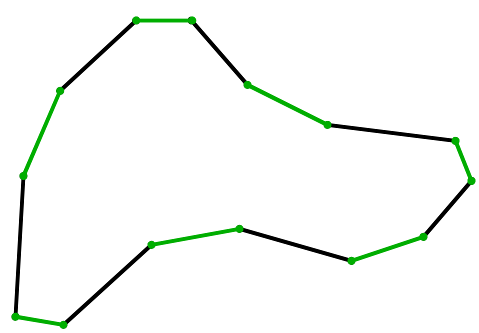
Corner Cutting



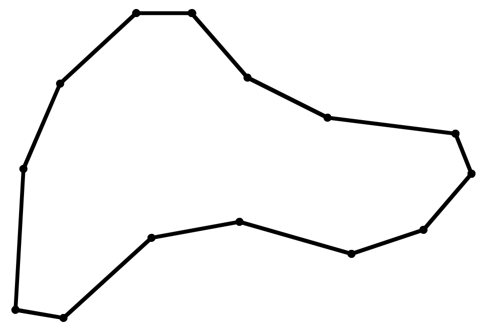
Corner Cutting



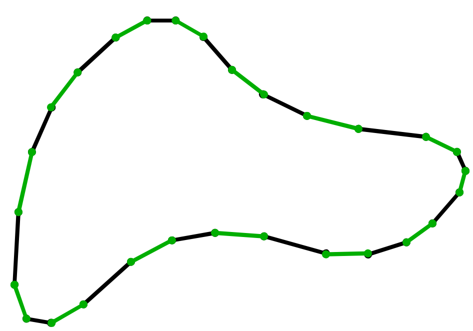
Corner Cutting



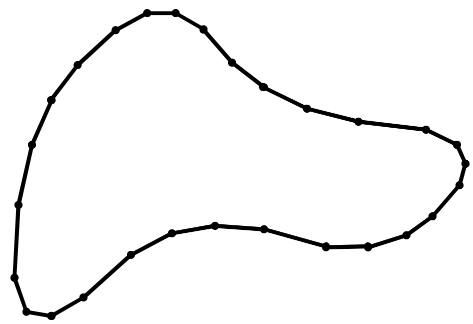
Corner Cutting



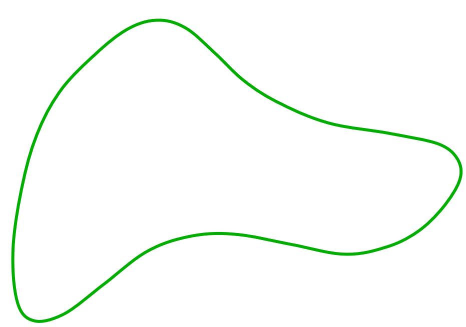
Corner Cutting



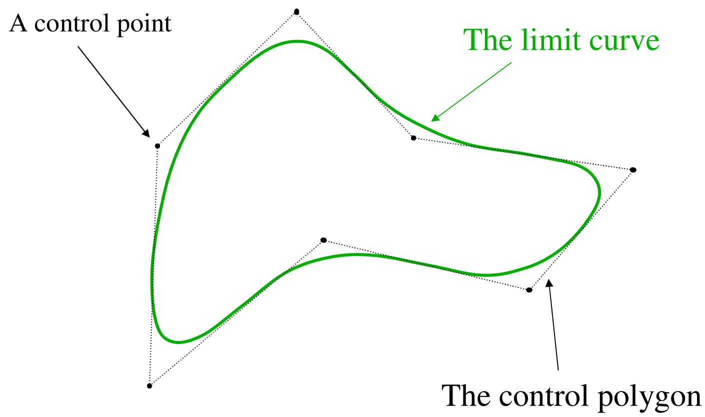
Corner Cutting



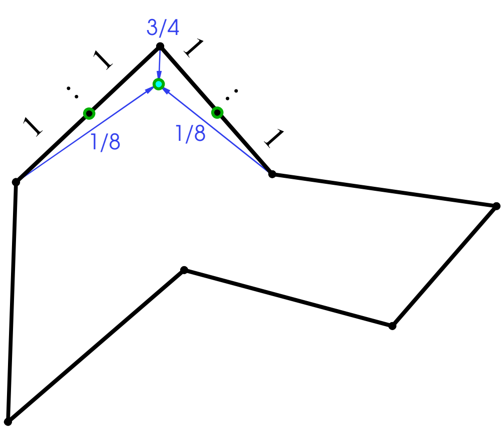
Corner Cutting



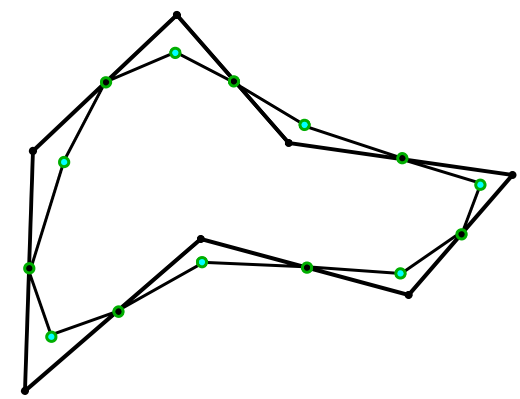
Corner Cutting



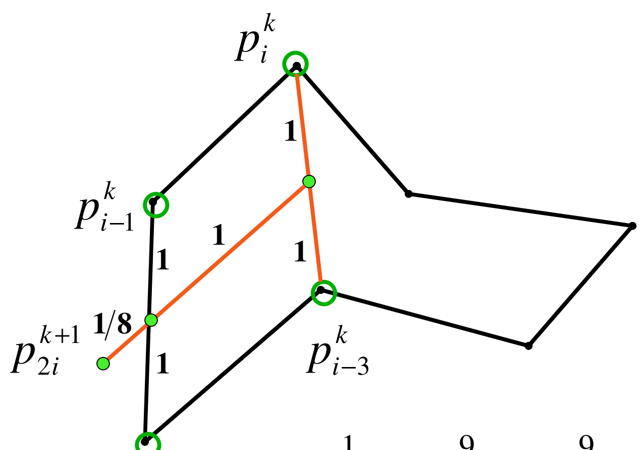
Cubic Corner Cutting



Cubic Corner Cutting

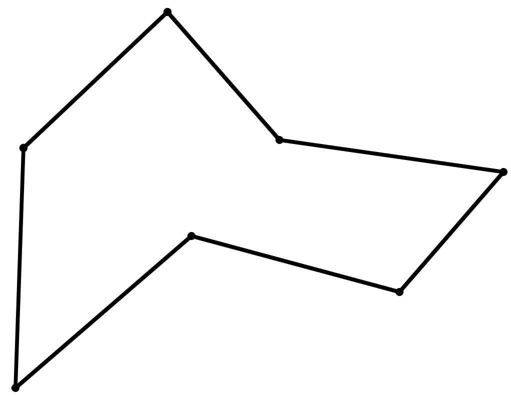


The 4-point scheme
Dyn, Gregory, Levin 1987



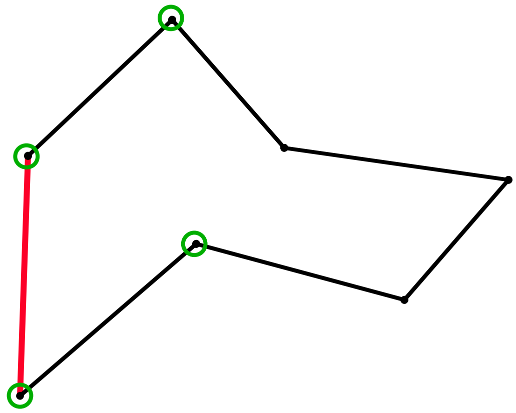
$$p_{2i-1}^{k+1} = p_{i-2}^k \qquad p_{2i}^{k+1} = -\frac{1}{16}p_{i-3}^k + \frac{9}{16}p_{i-2}^k + \frac{9}{16}p_{i-1}^k - \frac{1}{16}p_i^k$$

The 4-point scheme



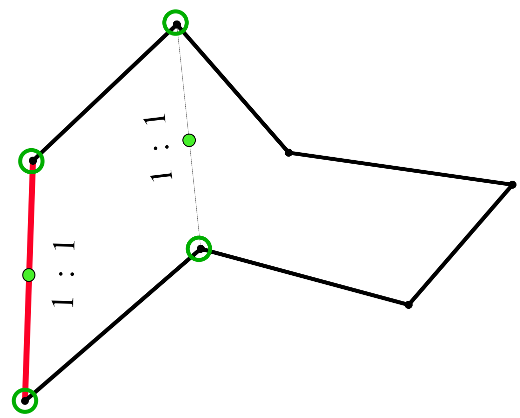
29

The 4-point scheme



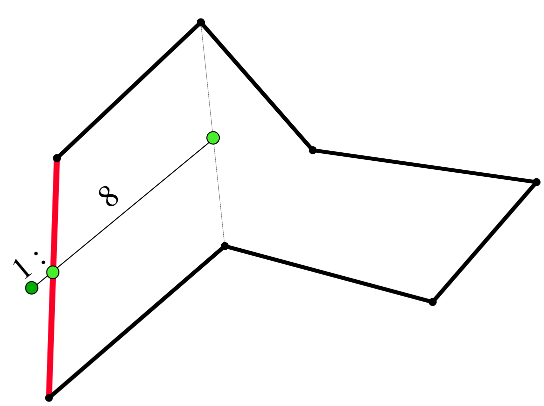
30

The 4-point scheme



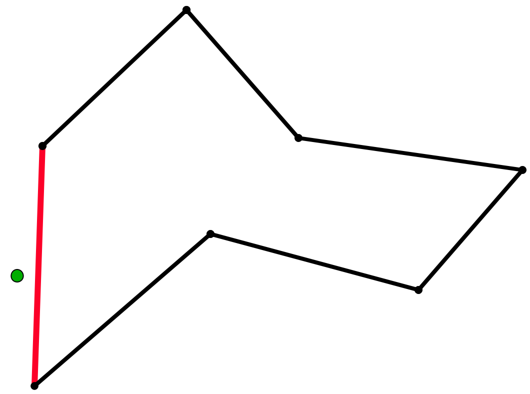
31

The 4-point scheme

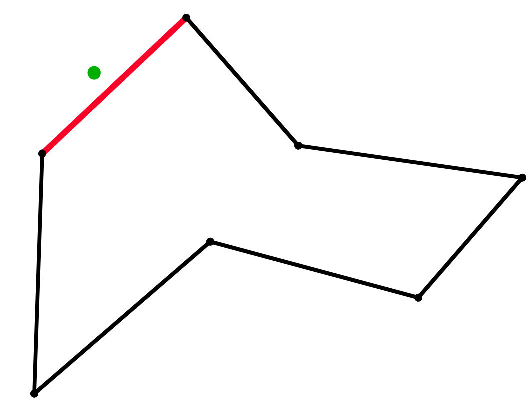


32

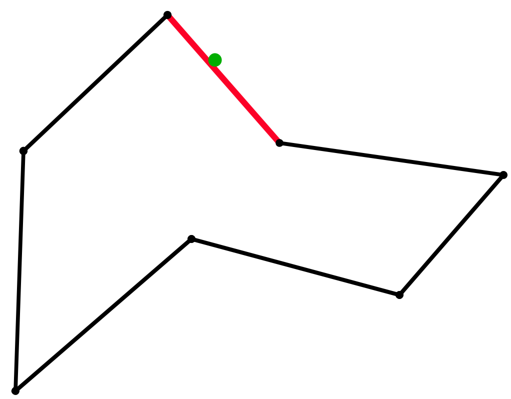
The 4-point scheme



The 4-point scheme

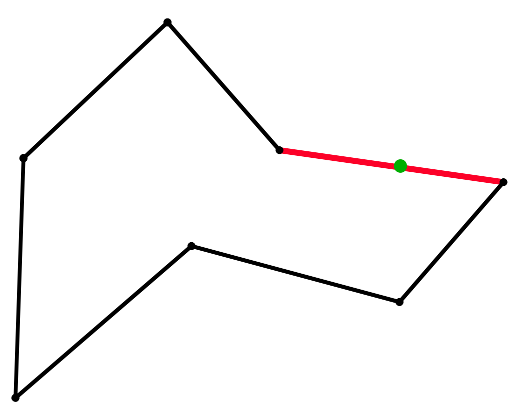


The 4-point scheme



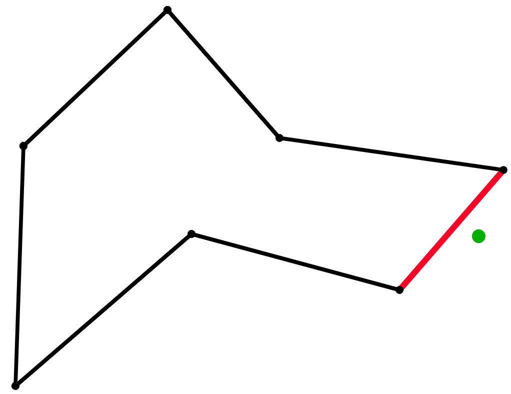
35

The 4-point scheme

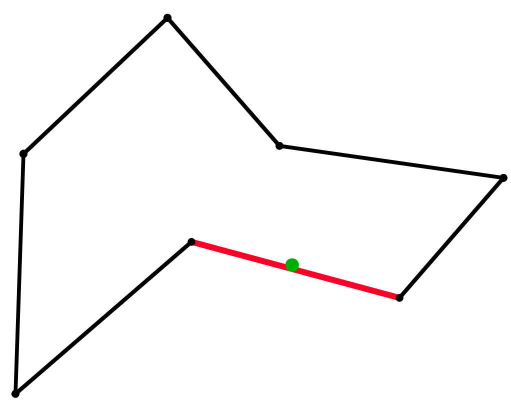


36

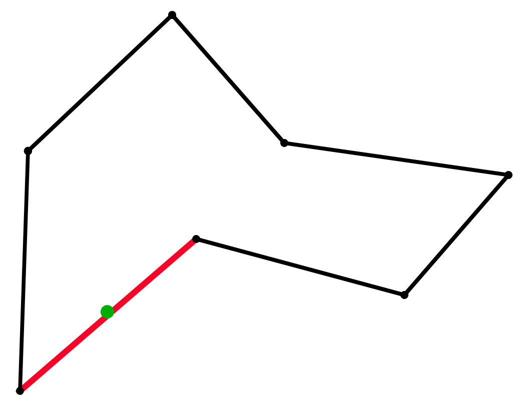
The 4-point scheme



The 4-point scheme

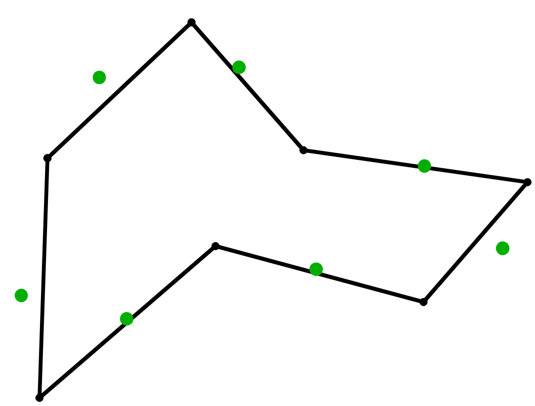


The 4-point scheme



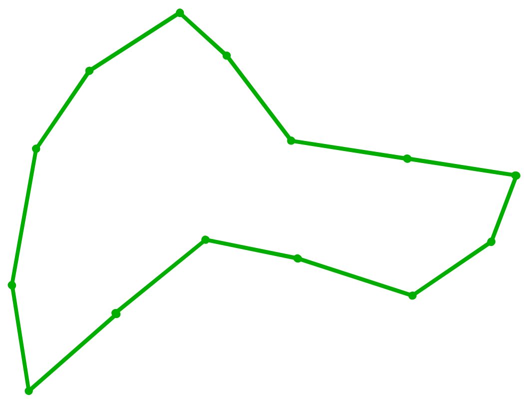
39

The 4-point scheme



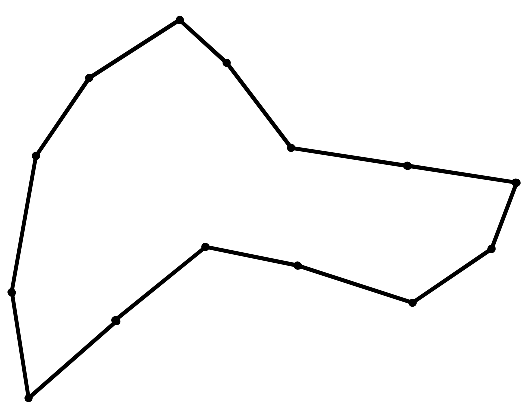
40

The 4-point scheme



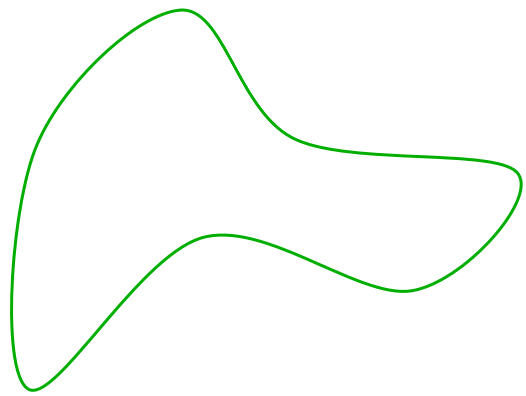
41

The 4-point scheme

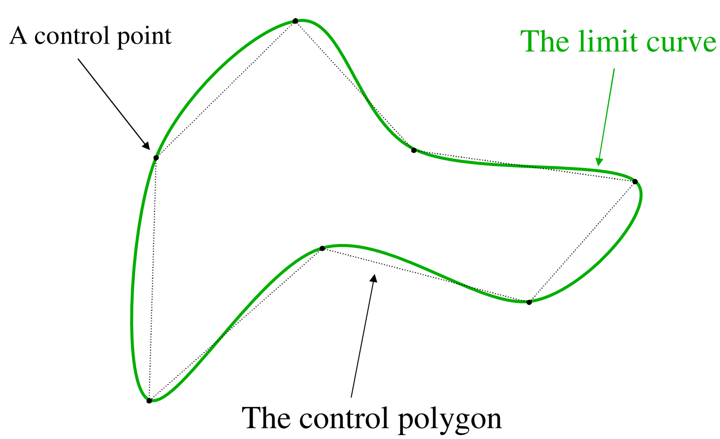


42

The 4-point scheme

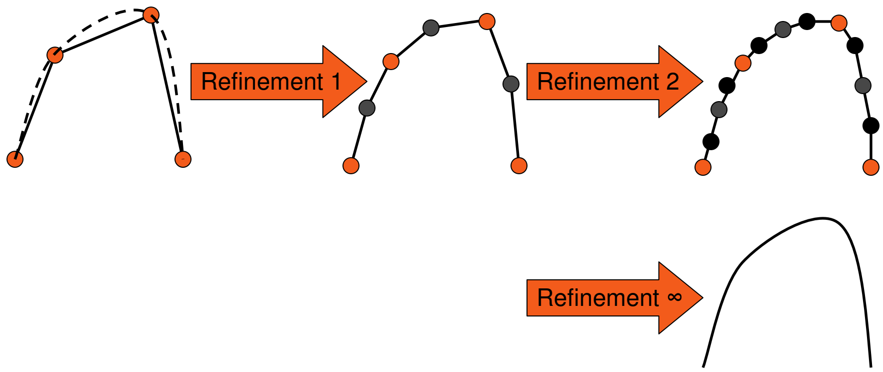


The 4-point scheme



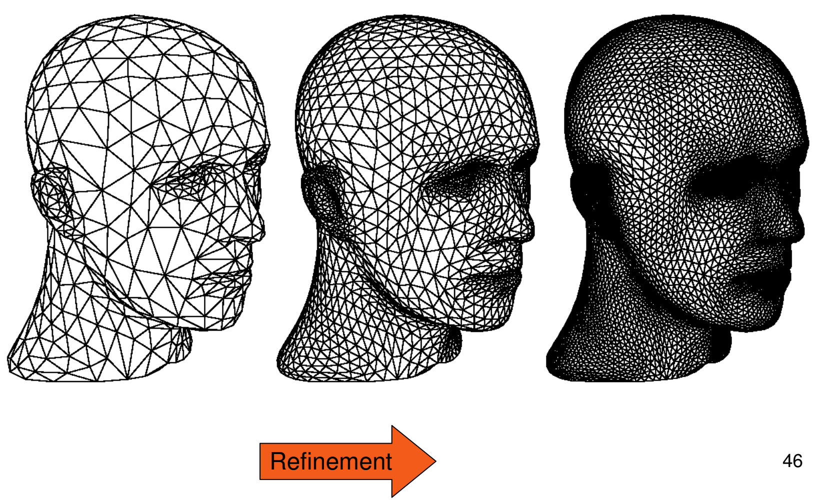
Subdivision Curves

Approach Limit Curve Surface through an Iterative Refinement Process.



Subdivision Surfaces

Same approach works in 3D



Applications of Subdivision Surfaces

Geri's Game (1989) : Pixar Animation Studios