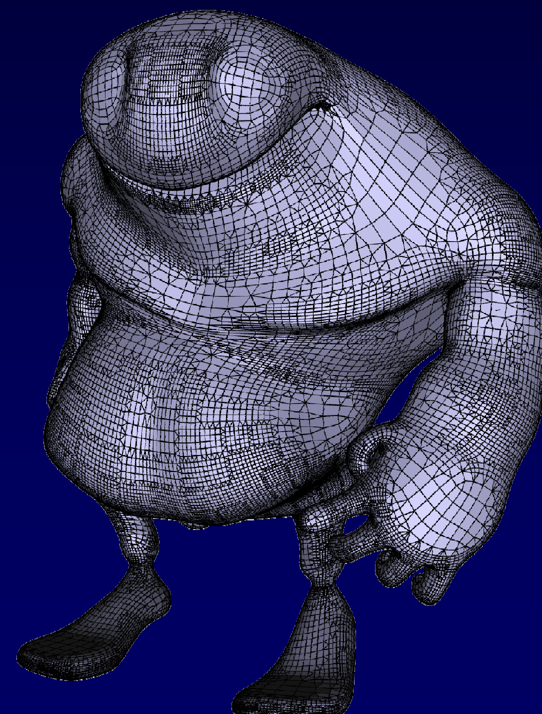
Exact Evaluation of Non-Polynomial Subdivision Schemes at Rational Parameter Values

Scott Schaefer

Joe Warren



TAMU
Computer
Science

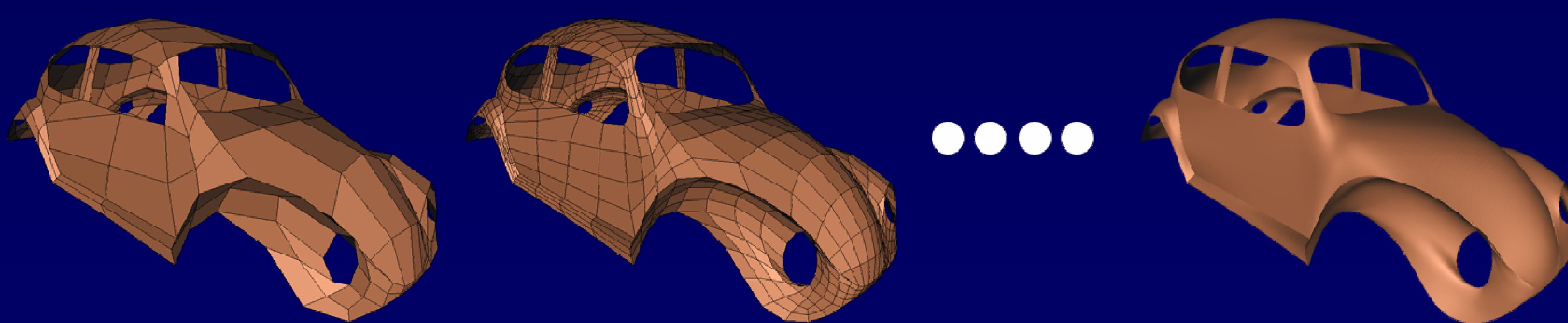
RICE
COMPUTER
SCIENCE

Subdivision Surfaces

- Set of rules S that recursively act on a shape p^0

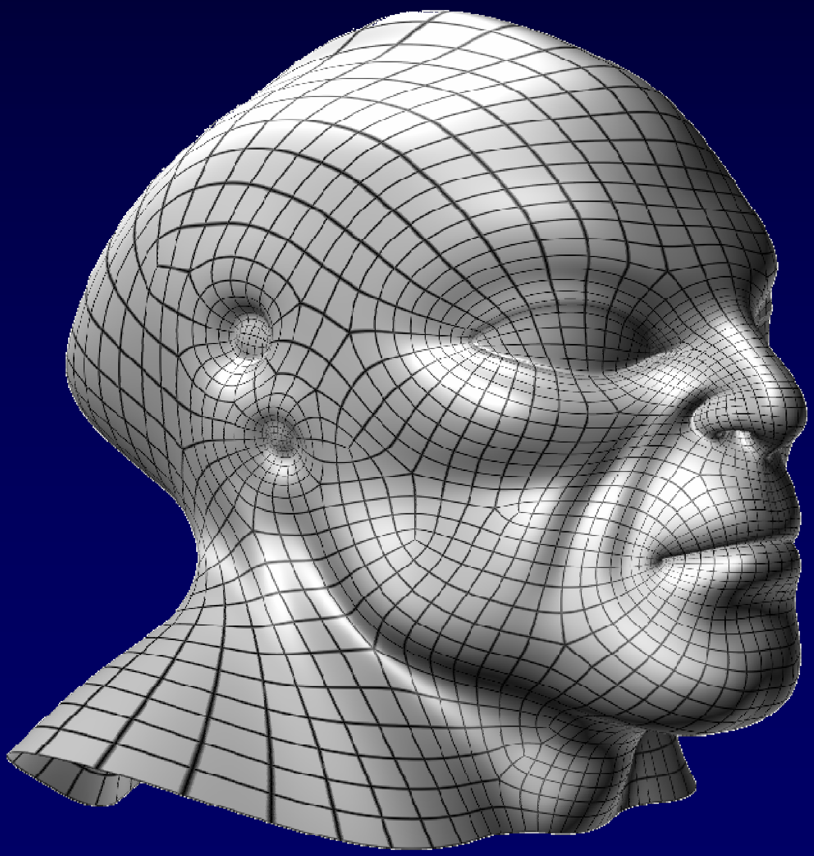
$$p^{k+1} = S\,p^k$$

- Arbitrary topology surfaces
- Smooth everywhere



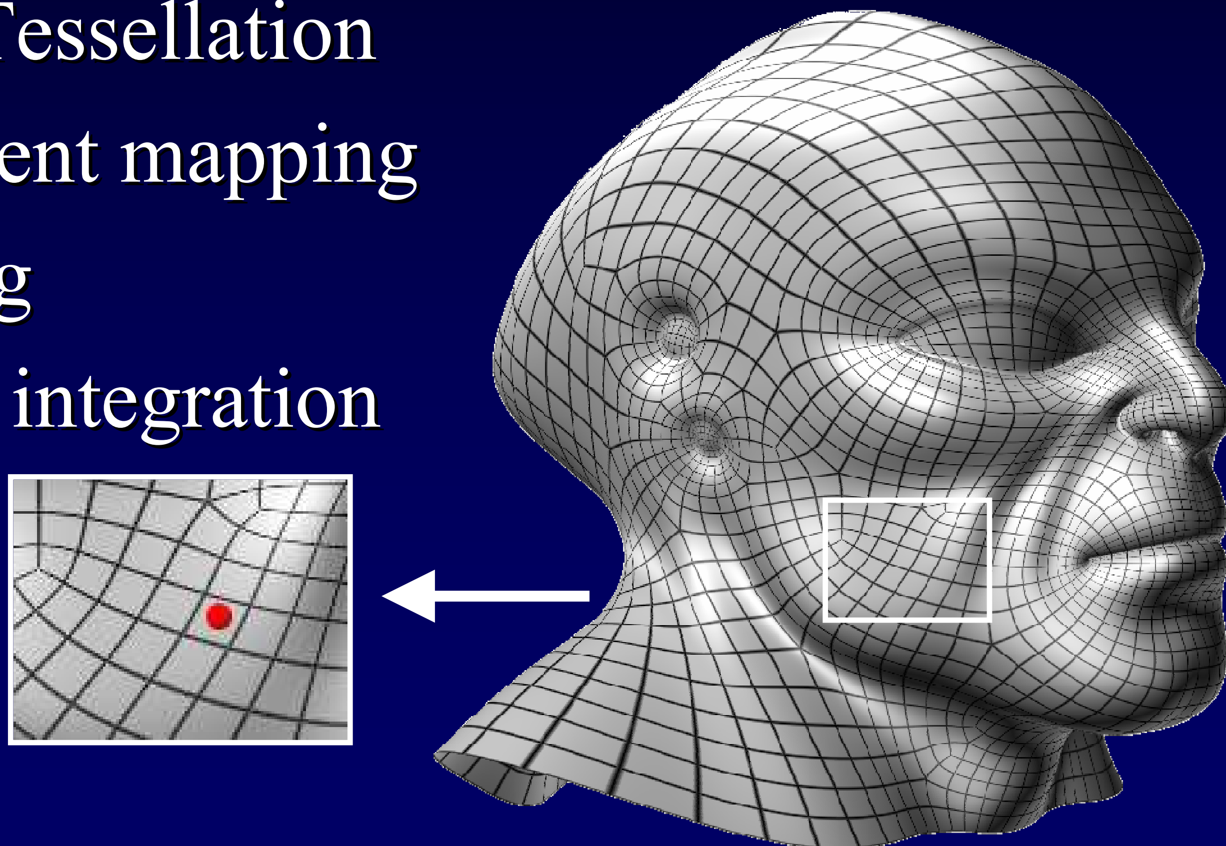
Exact Evaluation

- Important in
  - Adaptive Tessellation
  - Displacement mapping
  - Ray-tracing
  - Numerical integration
  - …

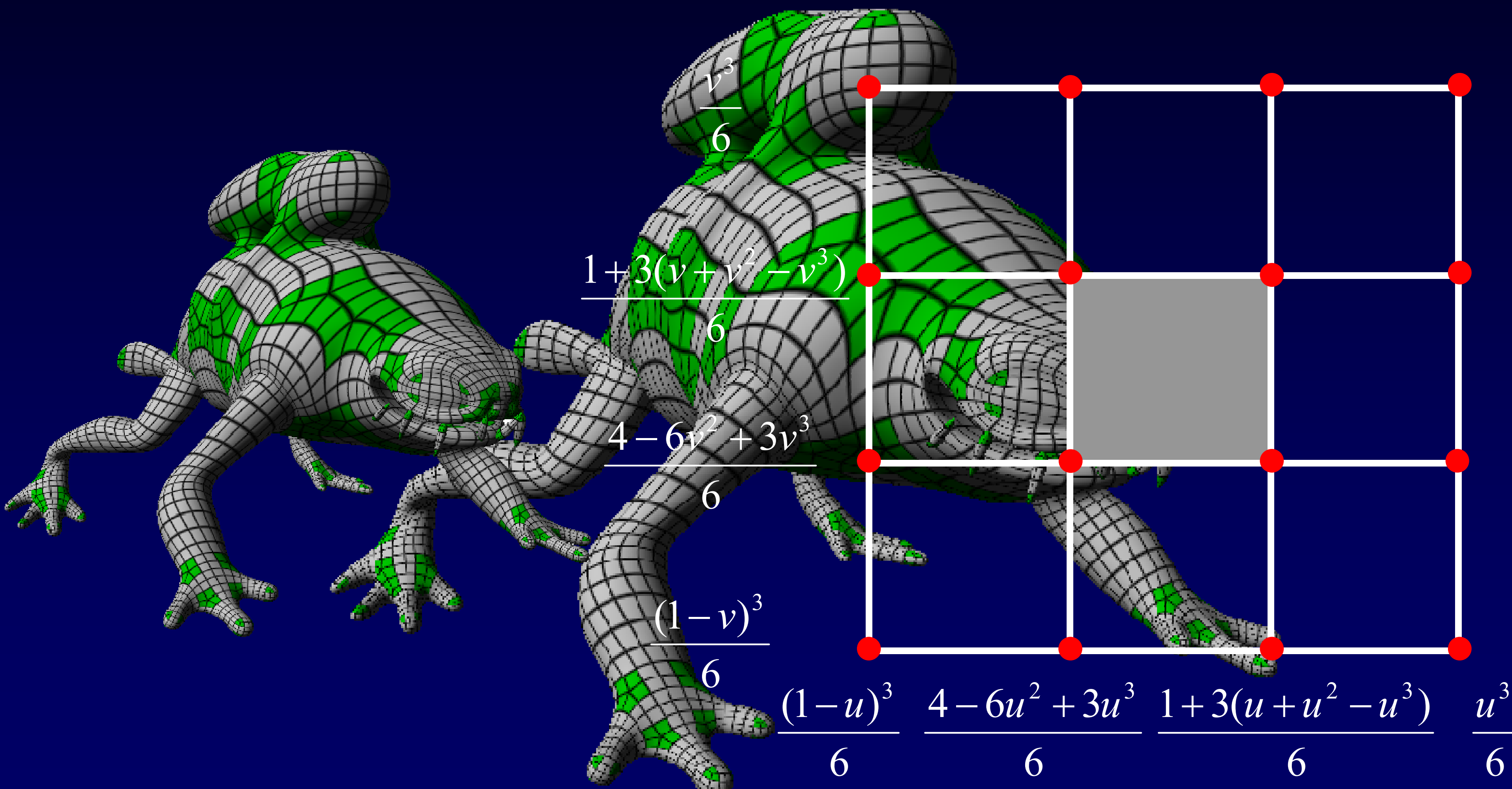


Exact Evaluation

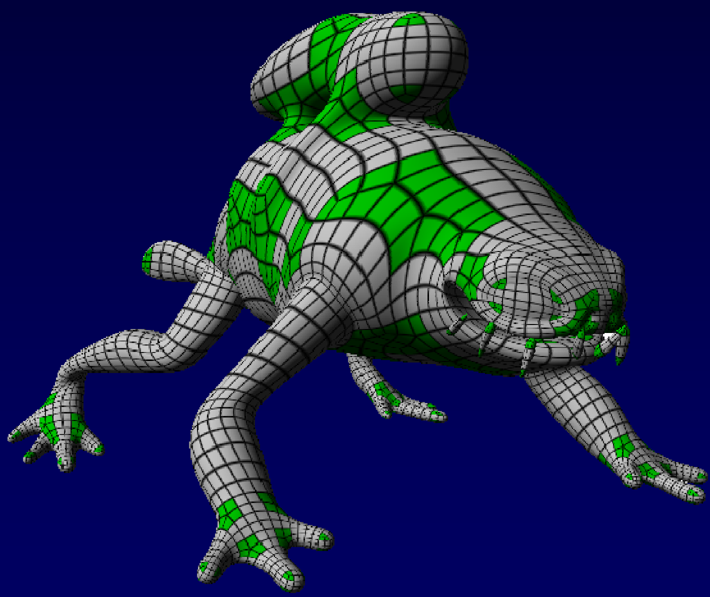
- Important in
  - Adaptive Tessellation
  - Displacement mapping
  - Ray-tracing
  - Numerical integration
  - …

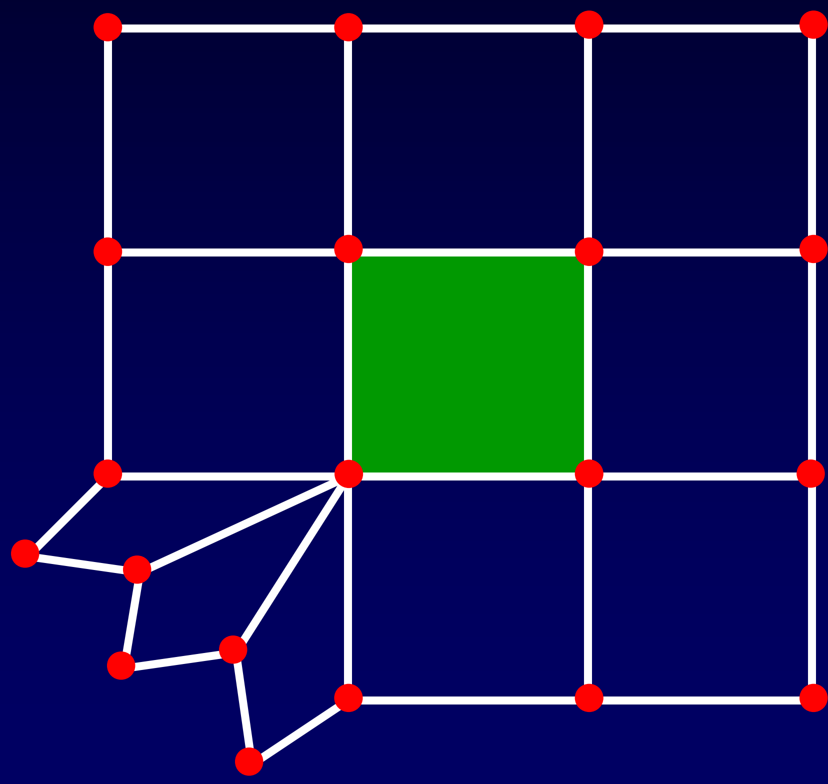


Polynomial Surfaces

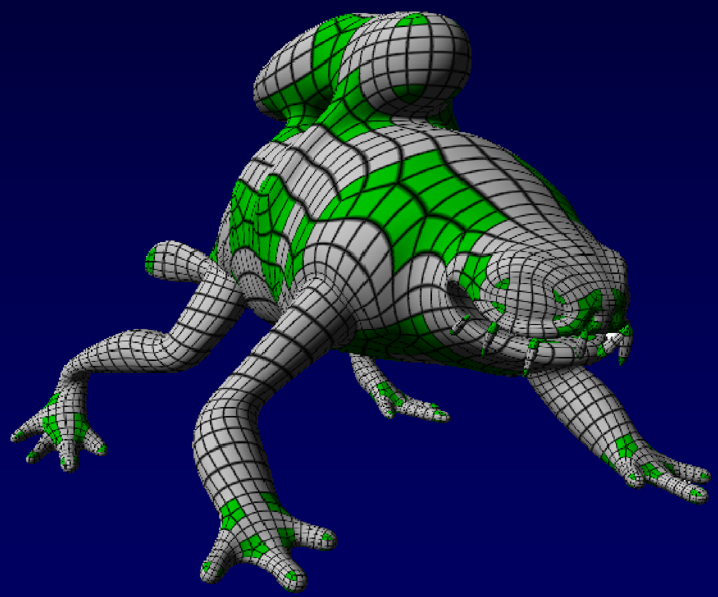


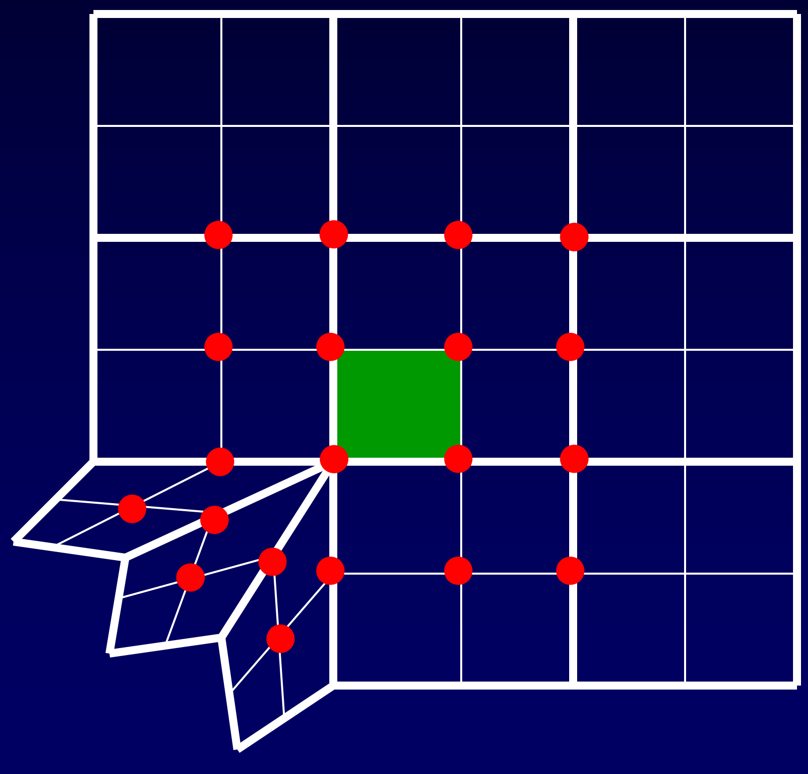
Polynomial Surfaces



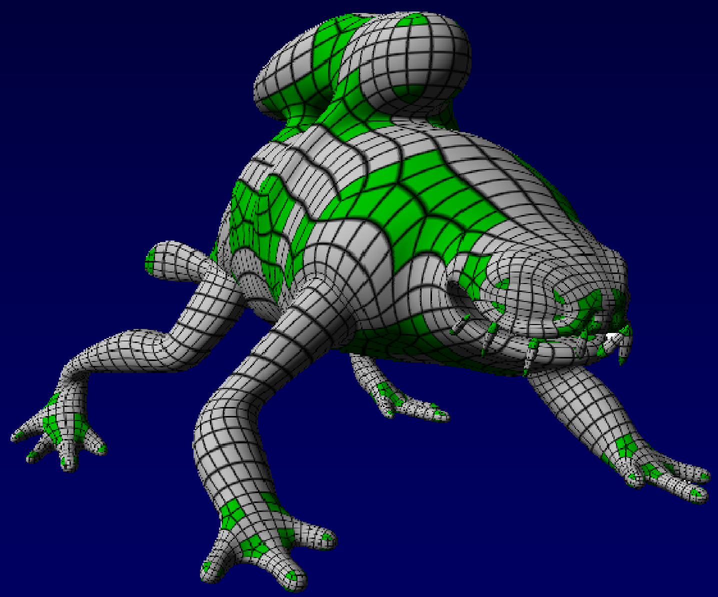


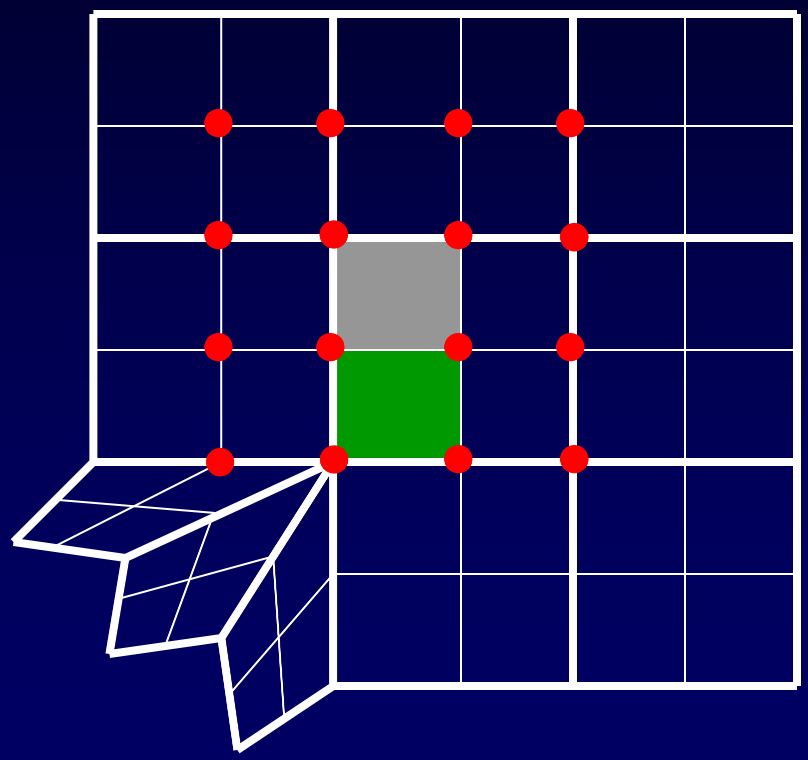
Polynomial Surfaces



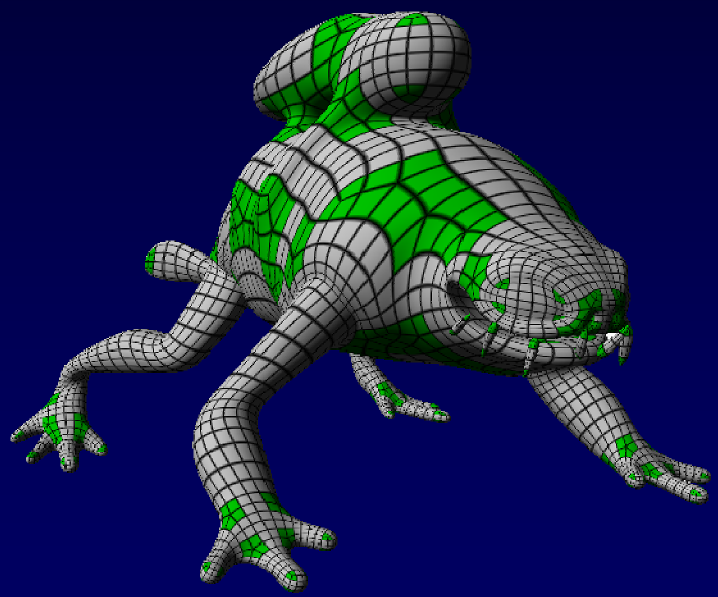


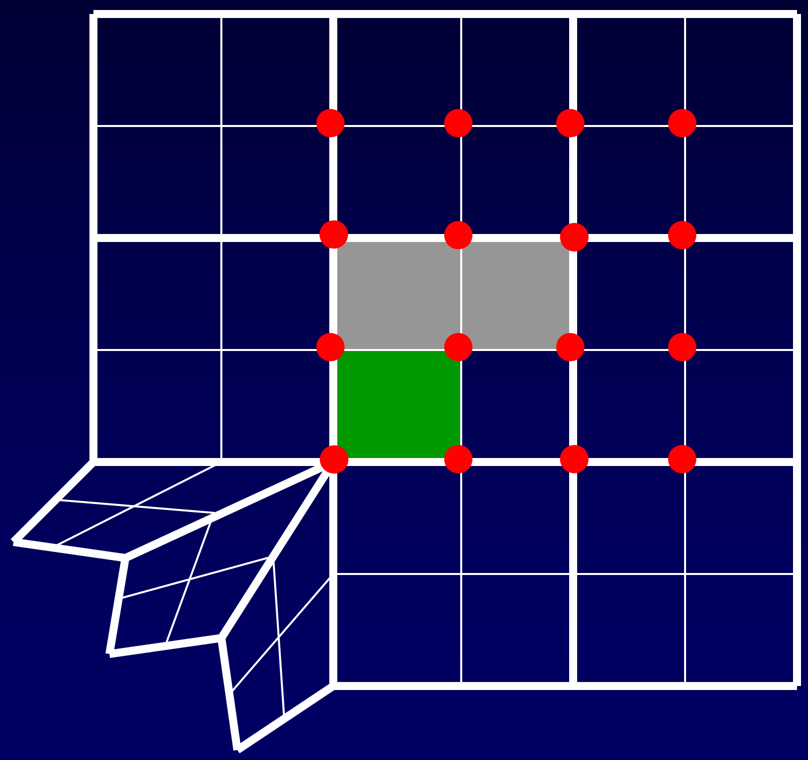
Polynomial Surfaces



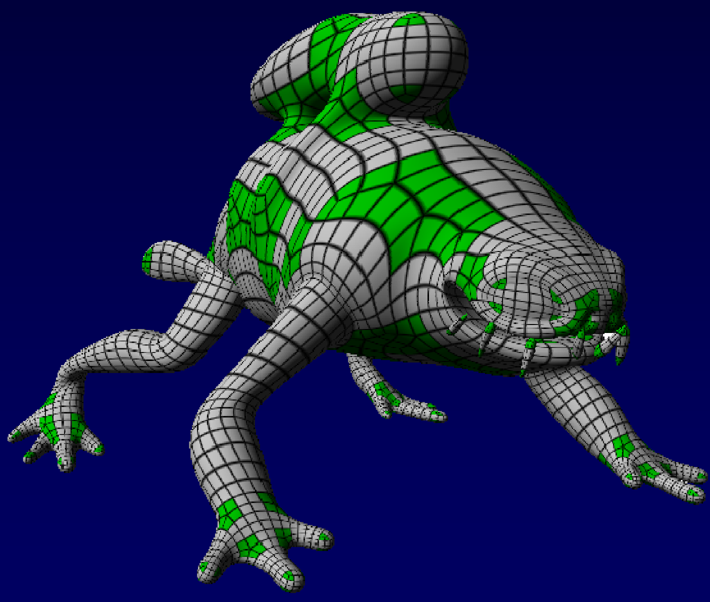


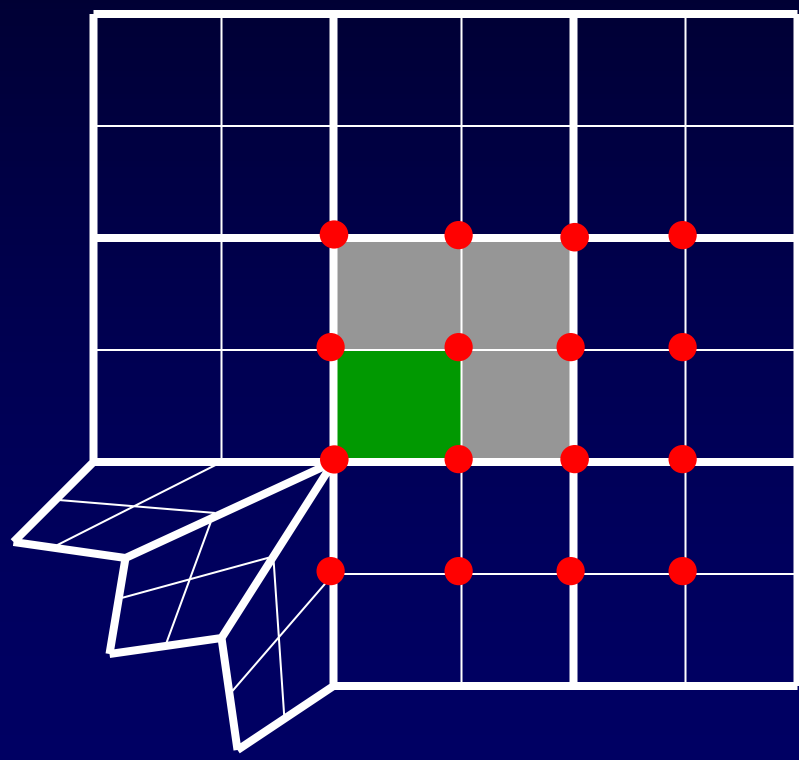
Polynomial Surfaces



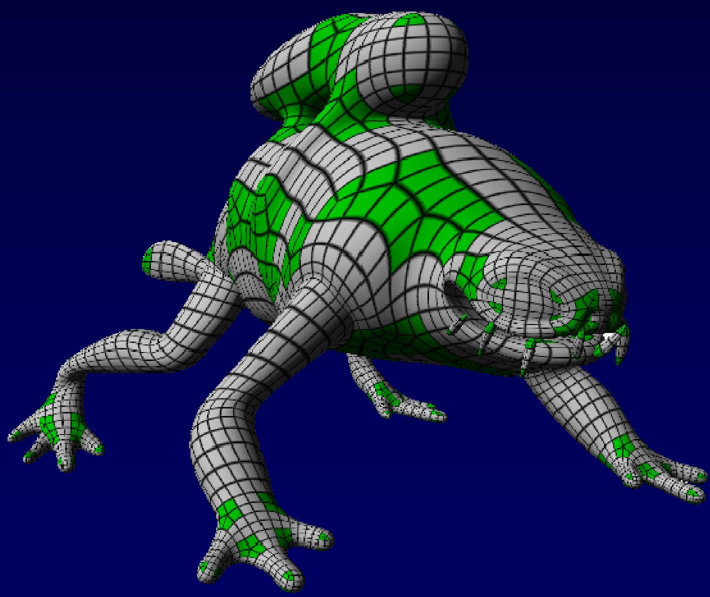


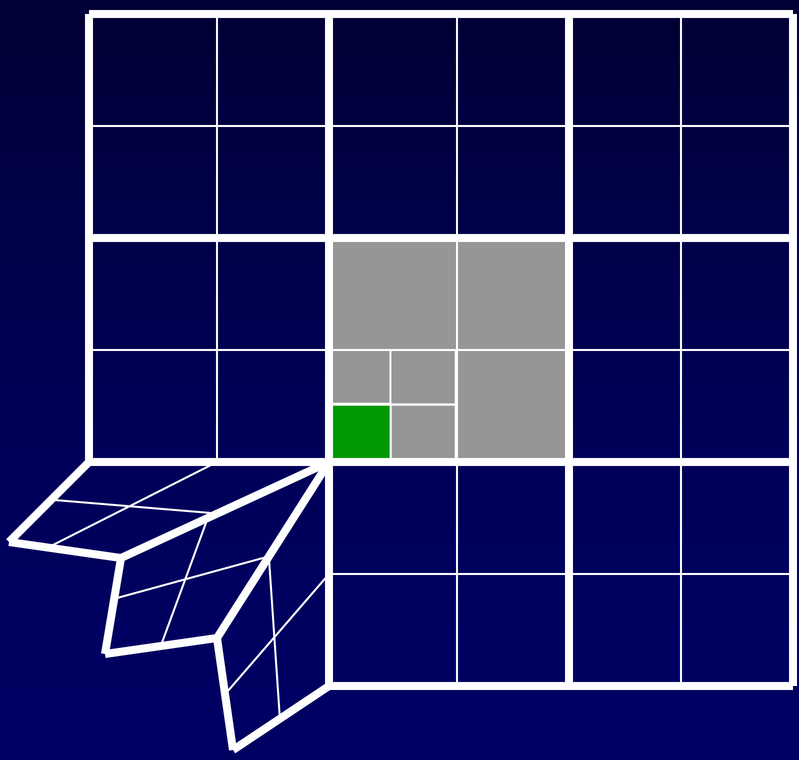
Polynomial Surfaces



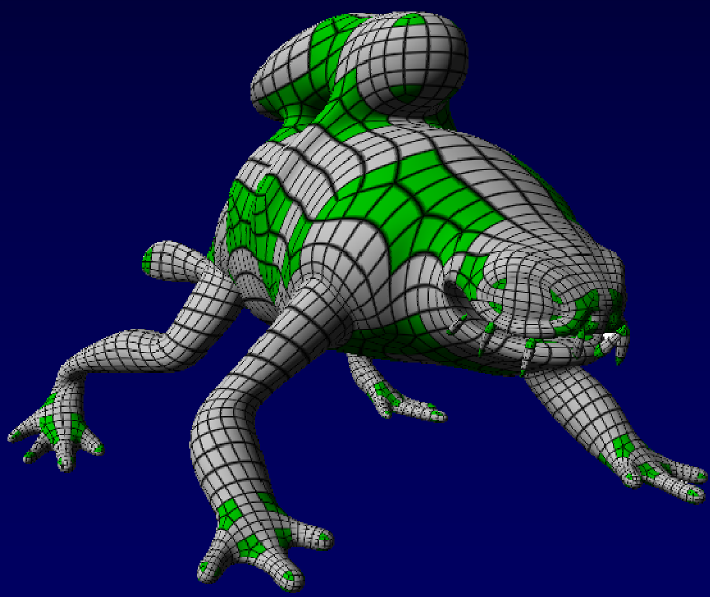


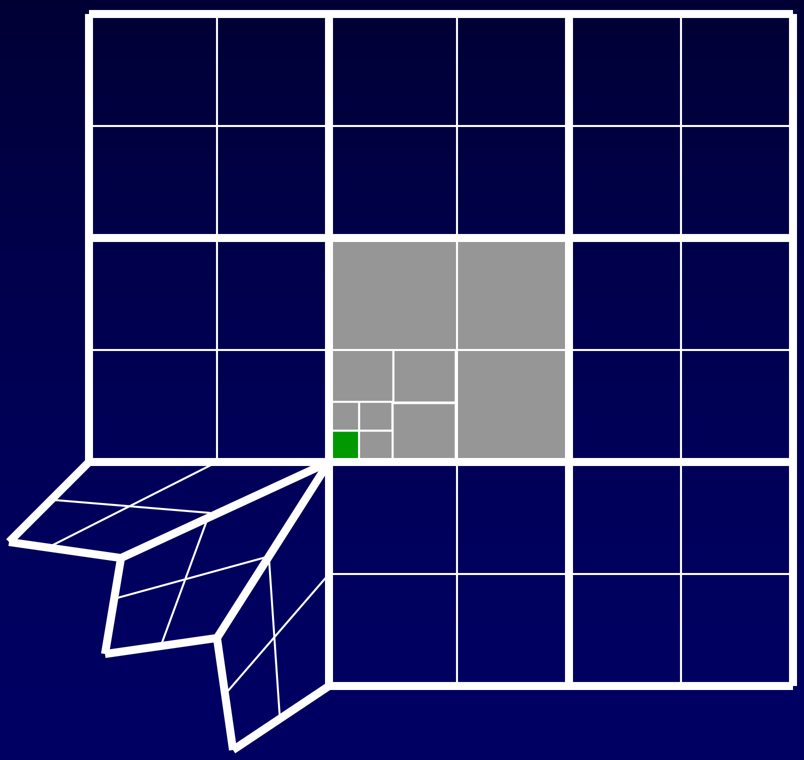
Polynomial Surfaces



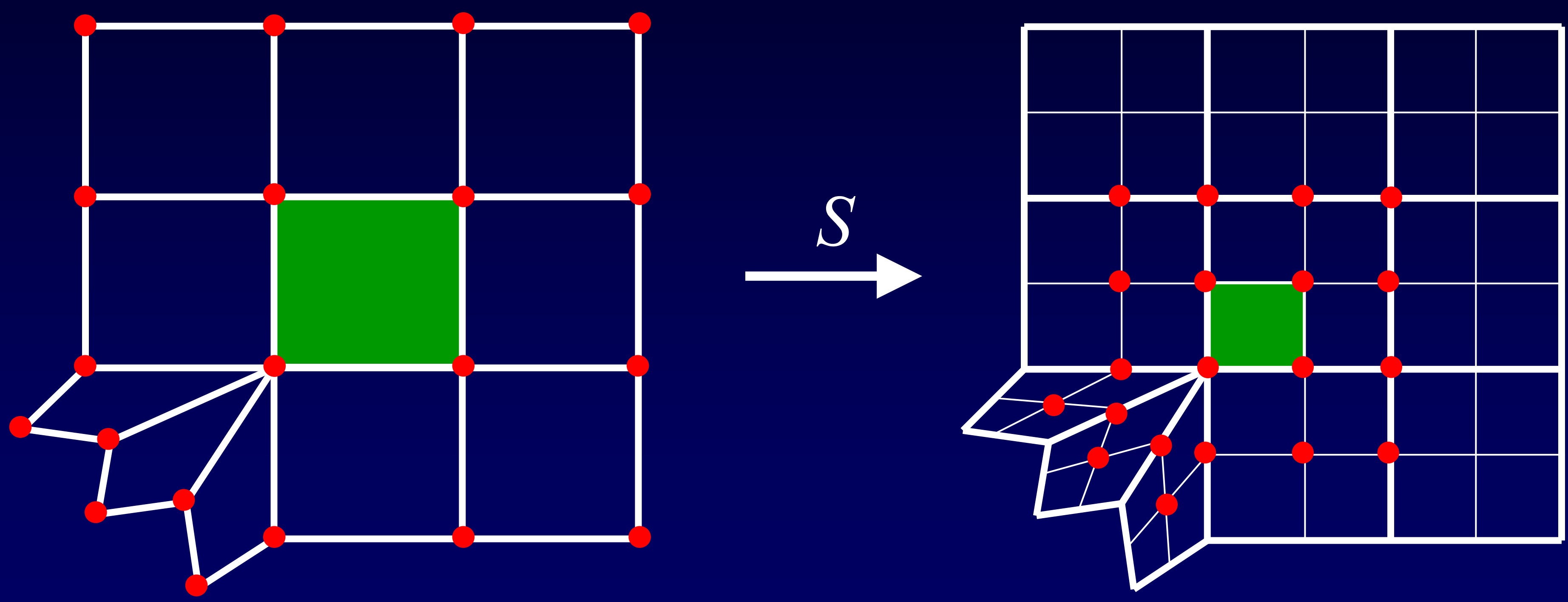


Polynomial Surfaces





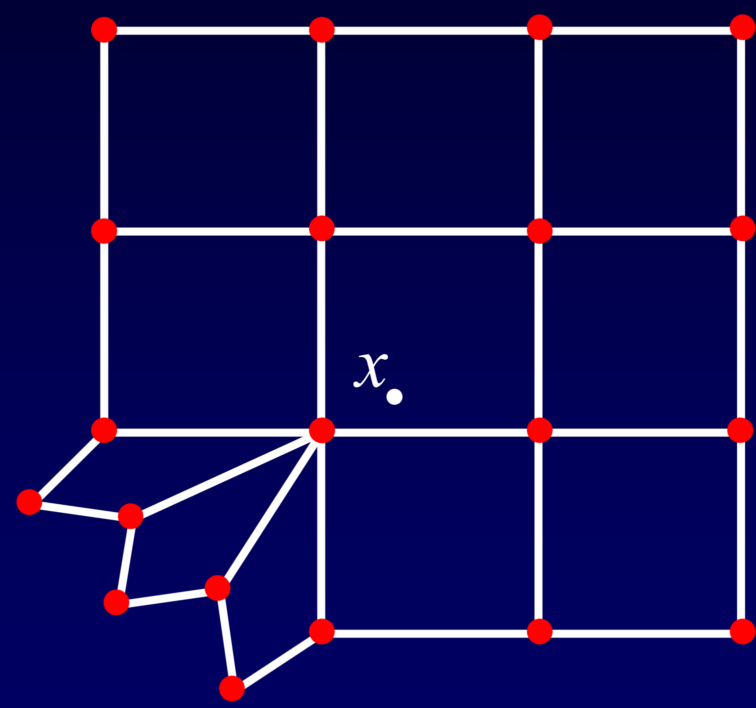
Stam's Exact Evaluation Algorithm



Stam's Exact Evaluation Algorithm

- Subdivide until *x* is in ordinary region

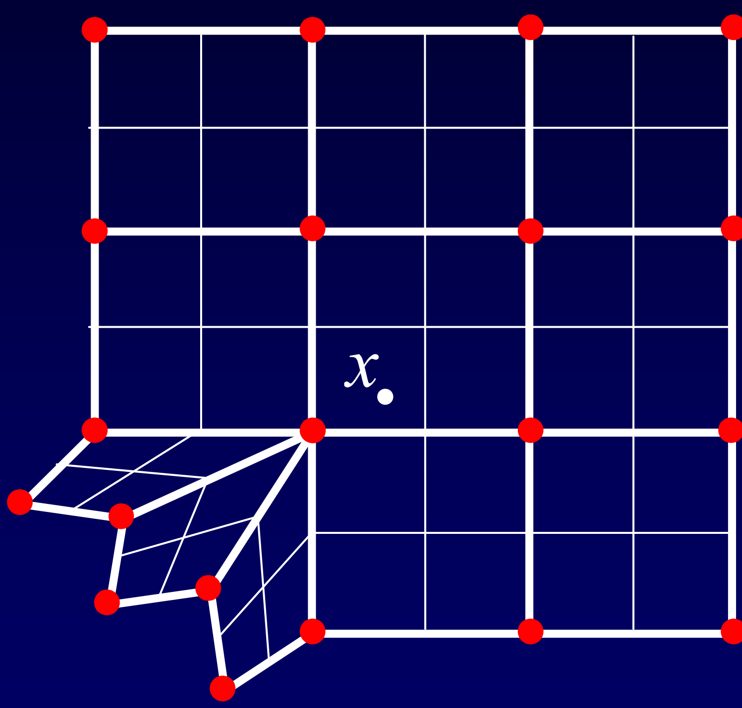
$$S^i P$$



Stam's Exact Evaluation Algorithm

- Subdivide until x is in ordinary region

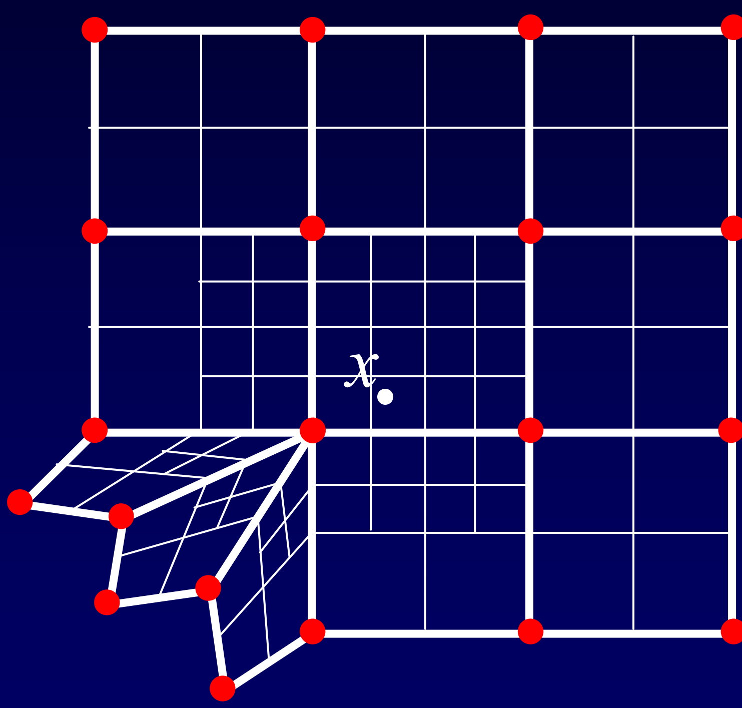
$$S^i P$$



Stam's Exact Evaluation Algorithm

- Subdivide until x is in ordinary region

$$S^i P$$

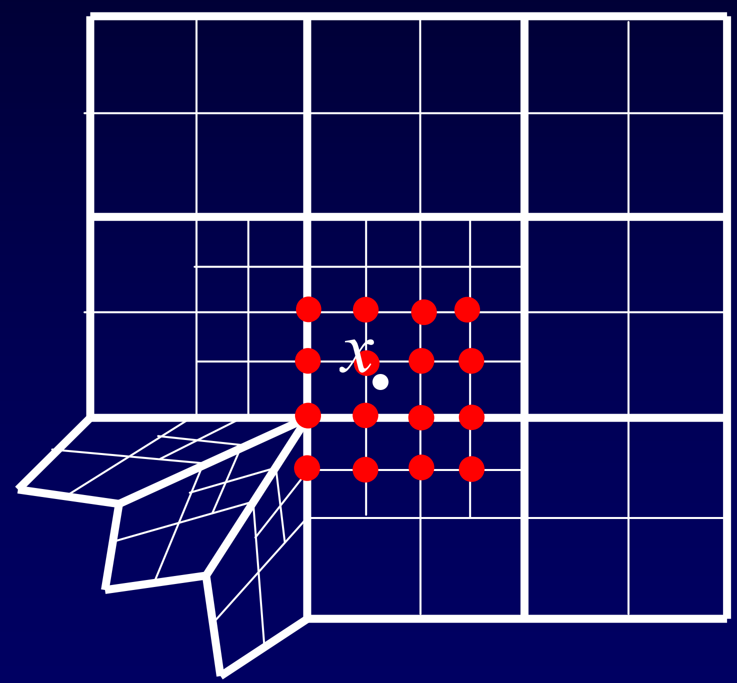


Stam's Exact Evaluation Algorithm

- Subdivide until *x* is in ordinary region

$$S^i P$$

- Extract B-spline control points and evaluate at *x*

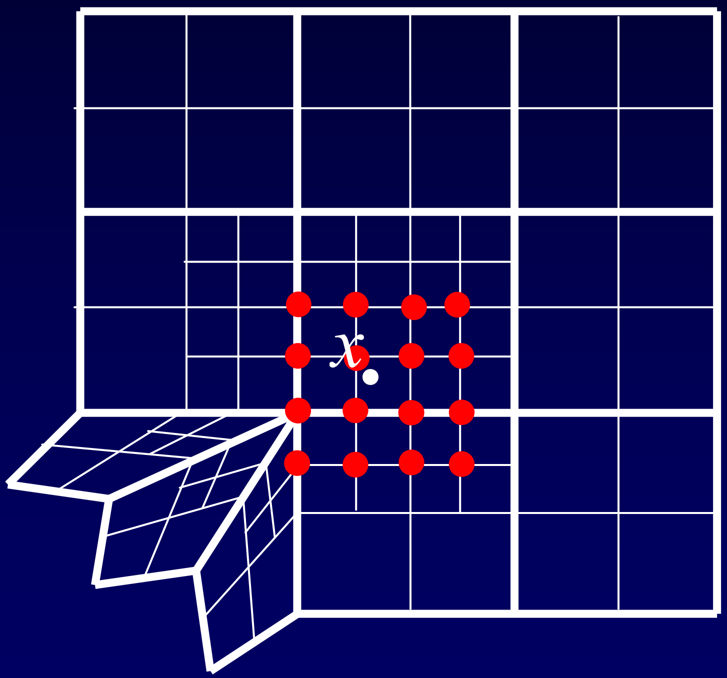


Stam's Exact Evaluation Algorithm

- Subdivide until *x* is in ordinary region

$$V\Lambda^i V^{-1} P$$

- Extract B-spline control points and evaluate at *x*



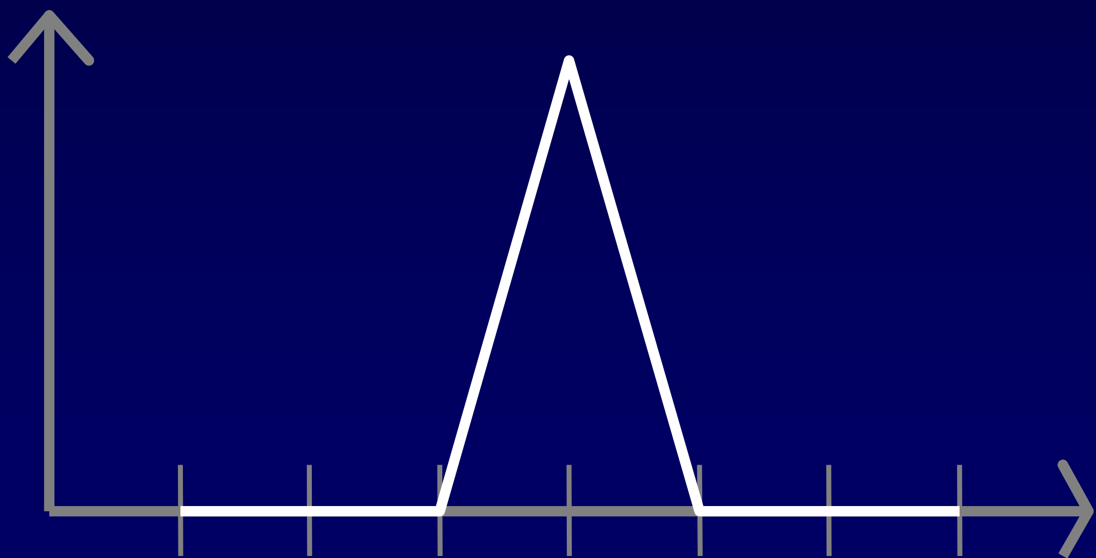
Limitations

Not all subdivision schemes are polynomial!

- Butterfly subdivision
- Kobbelt's Interpolatory Quad Scheme
- Sqrt(3) subdivision
- …

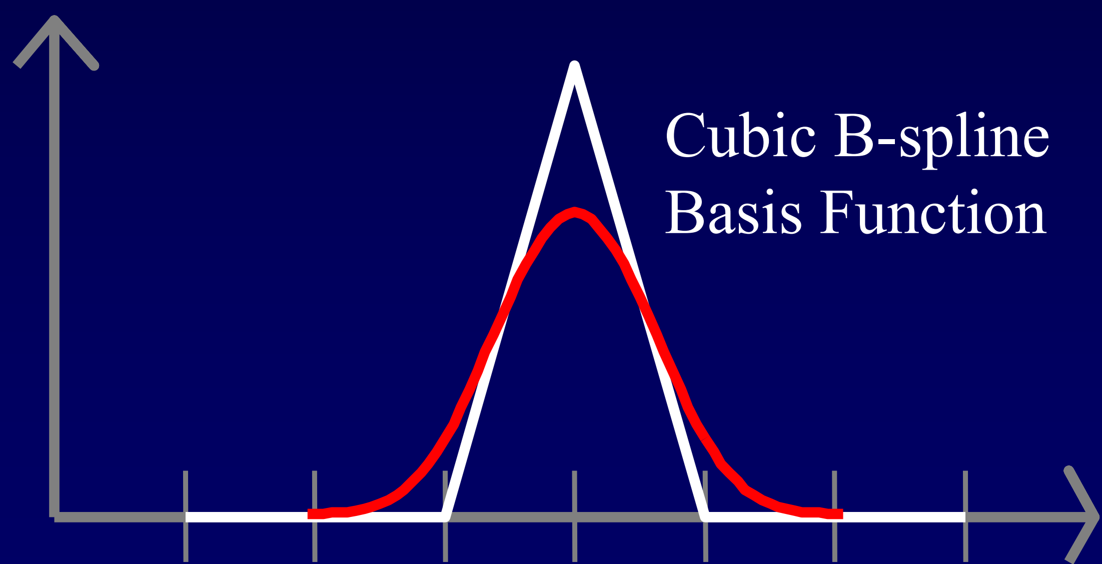
Subdivision as Basis Function Refinement

$$p(t) = \sum_i N(t-i)\, p_i$$



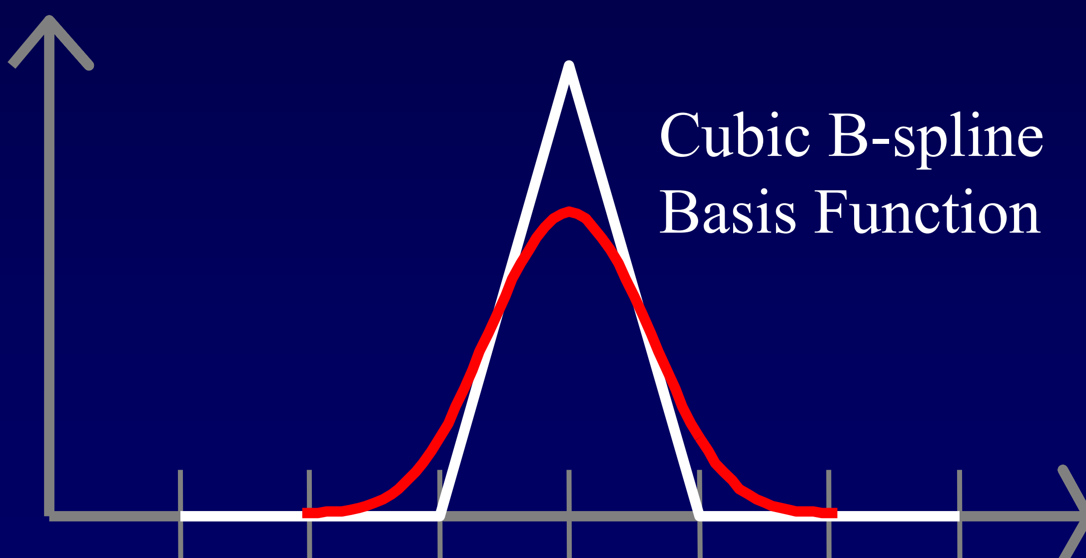
Subdivision as Basis Function Refinement

$$p(t) = \sum_i N(t-i)\, p_i$$



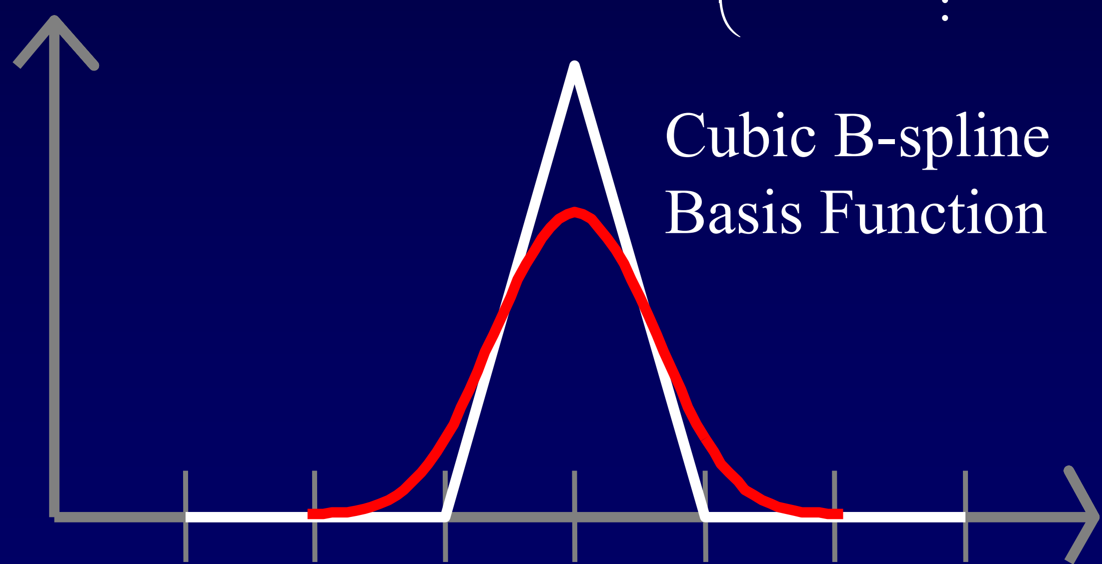
Subdivision as Basis Function Refinement

$$p(t) = \begin{pmatrix} \dots & N(t+1) & N(t) & N(t-1) & \dots \end{pmatrix} \begin{pmatrix} \vdots \\ p_1 \\ p_0 \\ p_{-1} \\ \vdots \end{pmatrix}$$



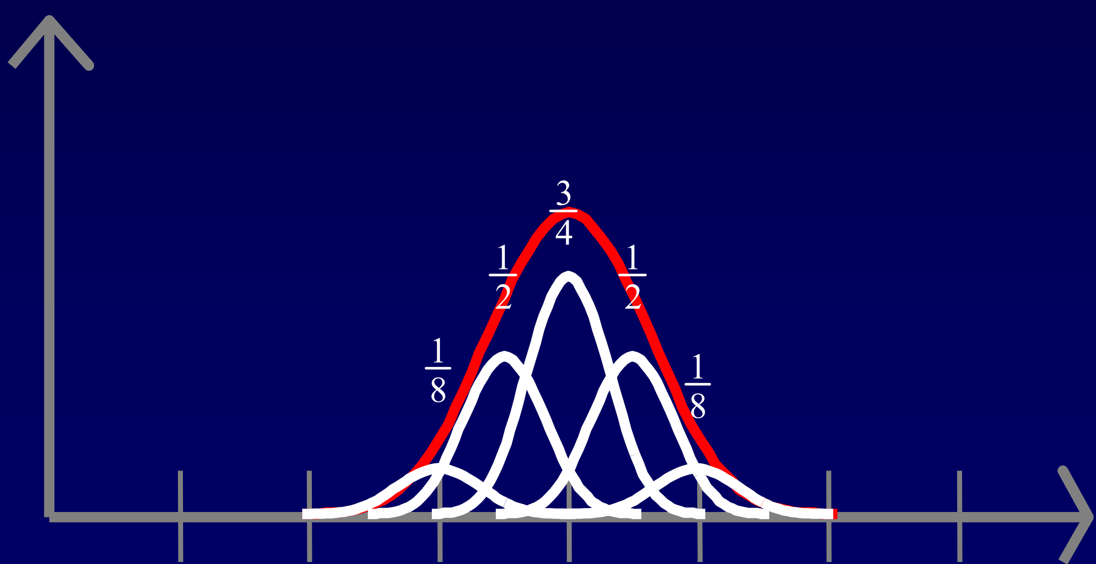
Subdivision as Basis Function Refinement

$$p(t) = \begin{pmatrix} \ldots & N(2t+1) & N(2t) & N(2t-1) & \ldots \end{pmatrix} \begin{pmatrix} & & & \vdots & & & \\ & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \\ & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \\ \ldots & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & \ldots \\ & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \\ & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\ & & & \vdots & & & \end{pmatrix} \begin{pmatrix} \vdots \\ p_1 \\ p_0 \\ p_{-1} \\ \vdots \end{pmatrix}$$



Subdivision as Basis Function Refinement

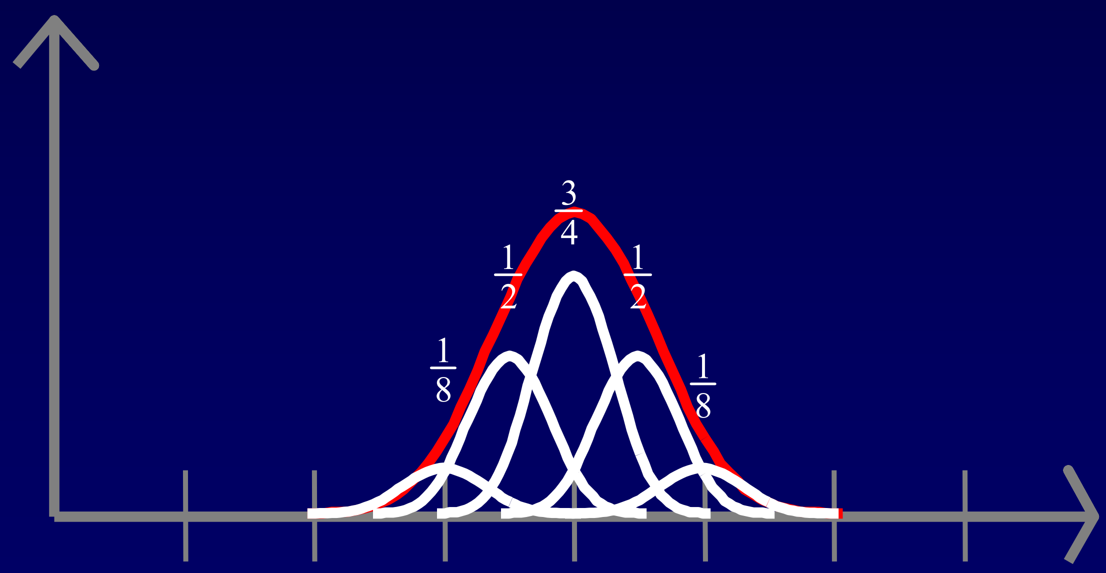
$$N(t) = \tfrac{1}{8}N(2t-2) + \tfrac{1}{2}N(2t-1) + \tfrac{3}{4}N(2t) + \tfrac{1}{2}N(2t+1) + \tfrac{1}{8}N(2t+2)$$



Subdivision as Basis Function Refinement

$$N(t) = \tfrac{1}{8}N(2t-2) + \tfrac{1}{2}N(2t-1) + \tfrac{3}{4}N(2t) + \tfrac{1}{2}N(2t+1) + \tfrac{1}{8}N(2t+2)$$

$$N(0) = \tfrac{1}{8}N(-2) + \tfrac{1}{2}N(-1) + \tfrac{3}{4}N(0) + \tfrac{1}{2}N(1) + \tfrac{1}{8}N(2)$$

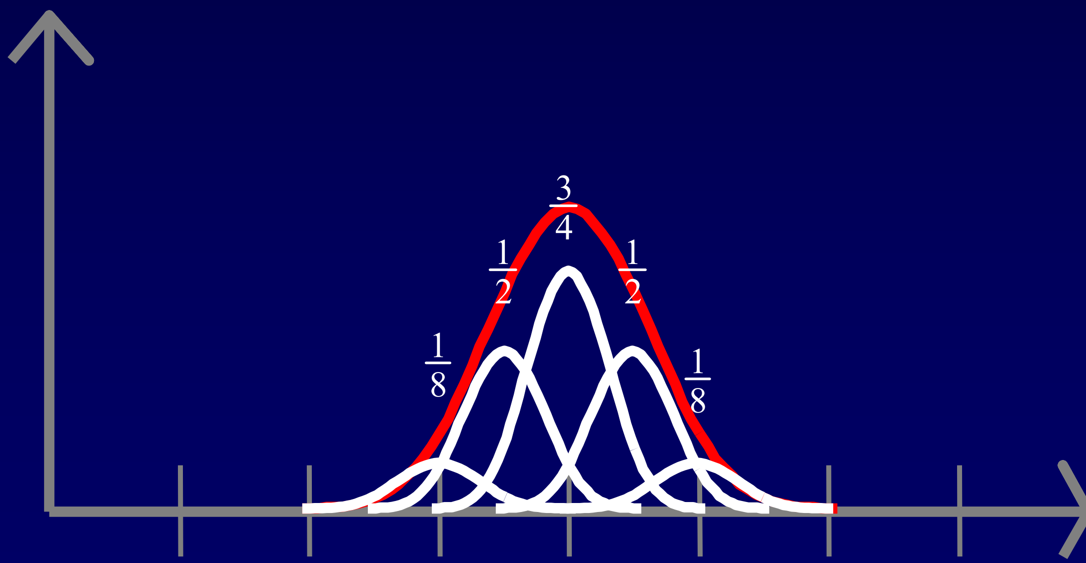


Subdivision as Basis Function Refinement

$$N(t)=\tfrac{1}{8}N(2t-2)+\tfrac{1}{2}N(2t-1)+\tfrac{3}{4}N(2t)+\tfrac{1}{2}N(2t+1)+\tfrac{1}{8}N(2t+2)$$

$$N(0)=\tfrac{1}{8}N(-2)+\tfrac{1}{2}N(-1)+\tfrac{3}{4}N(0)+\tfrac{1}{2}N(1)+\tfrac{1}{8}N(2)$$

$$N(1)=\tfrac{1}{8}N(0)+\tfrac{1}{2}N(1)+\tfrac{3}{4}N(2)$$



Subdivision as Basis Function Refinement

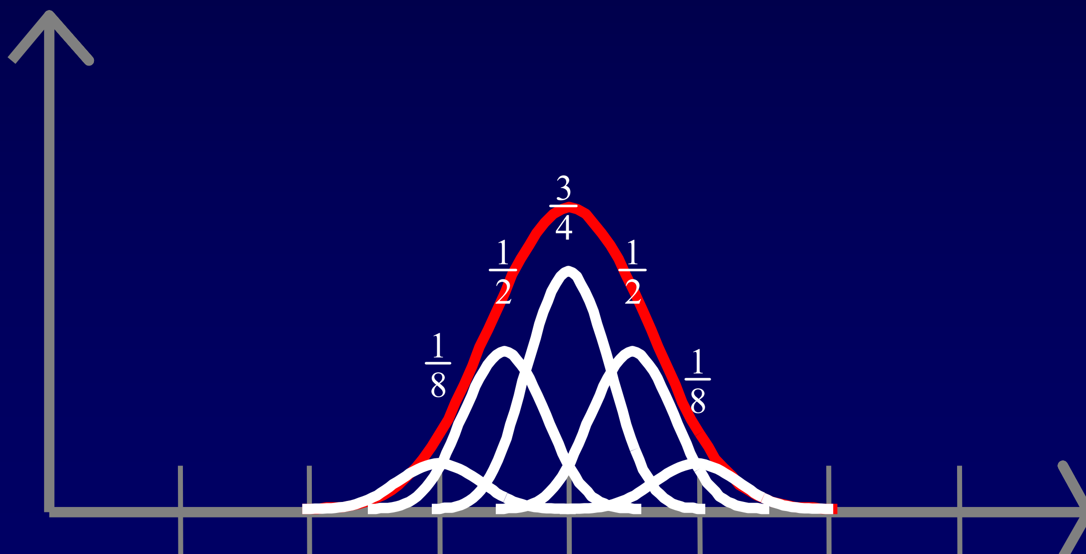
$$N(t) = \tfrac{1}{8}N(2t-2) + \tfrac{1}{2}N(2t-1) + \tfrac{3}{4}N(2t) + \tfrac{1}{2}N(2t+1) + \tfrac{1}{8}N(2t+2)$$

$$N(0) = \tfrac{1}{8}N(-2) + \tfrac{1}{2}N(-1) + \tfrac{3}{4}N(0) + \tfrac{1}{2}N(1) + \tfrac{1}{8}N(2)$$

$$N(1) = \tfrac{1}{8}N(0) + \tfrac{1}{2}N(1) + \tfrac{3}{4}N(2)$$

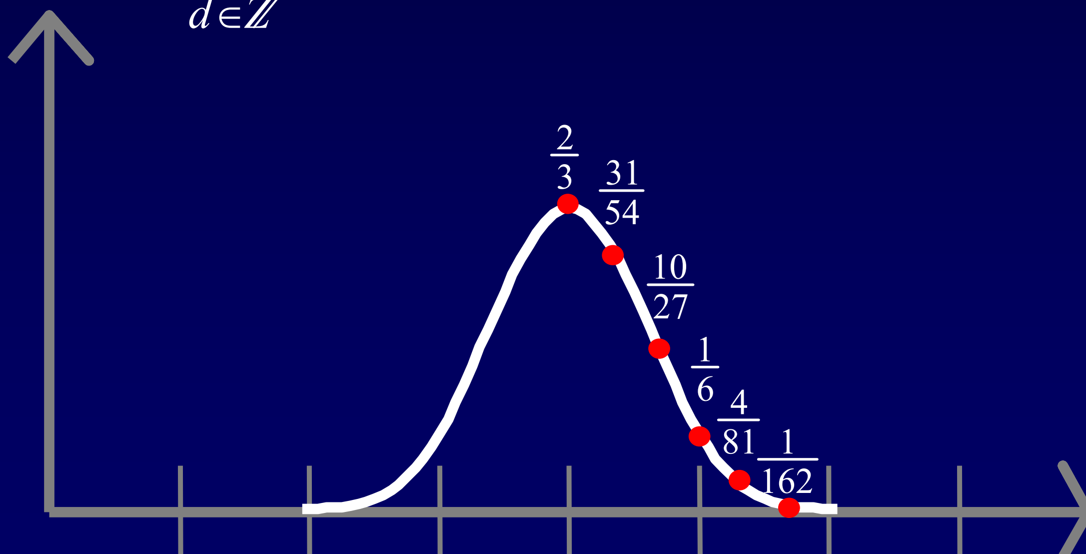
$$N(2) = \tfrac{1}{8}N(2)$$

$$\vdots$$



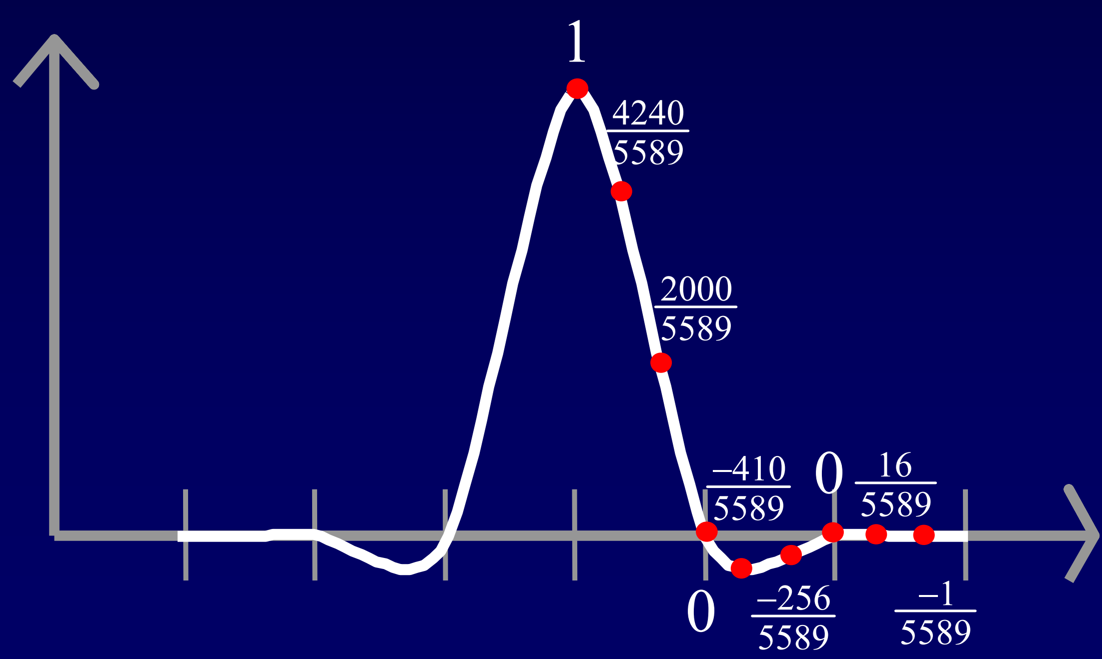
Exact Evaluation at Rational Values

- Evaluate scaling relationship at $N\left(\frac{i}{n}\right)$
- Solve linear system of equations with constraint $\sum_{d\in\mathbb{Z}} N\left(\frac{i}{n}-d\right)=1$



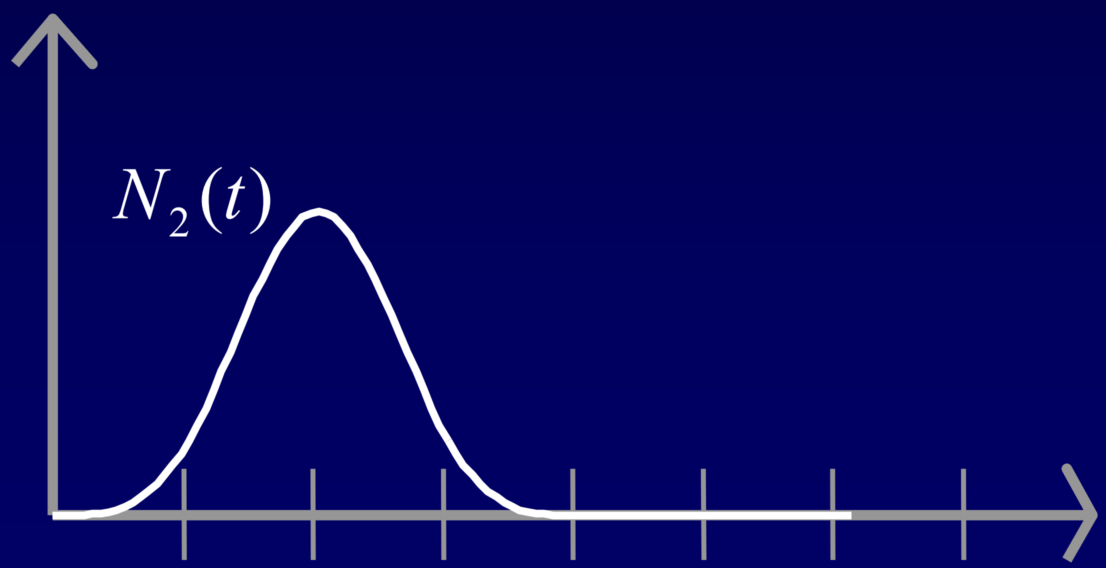
Evaluating Non-Polynomial Schemes

$$N(t) = \tfrac{-1}{16} N(2t-3) + \tfrac{9}{16} N(2t-1) + N(2t) + \tfrac{9}{16} N(2t+1) + \tfrac{-1}{16} N(2t+3)$$



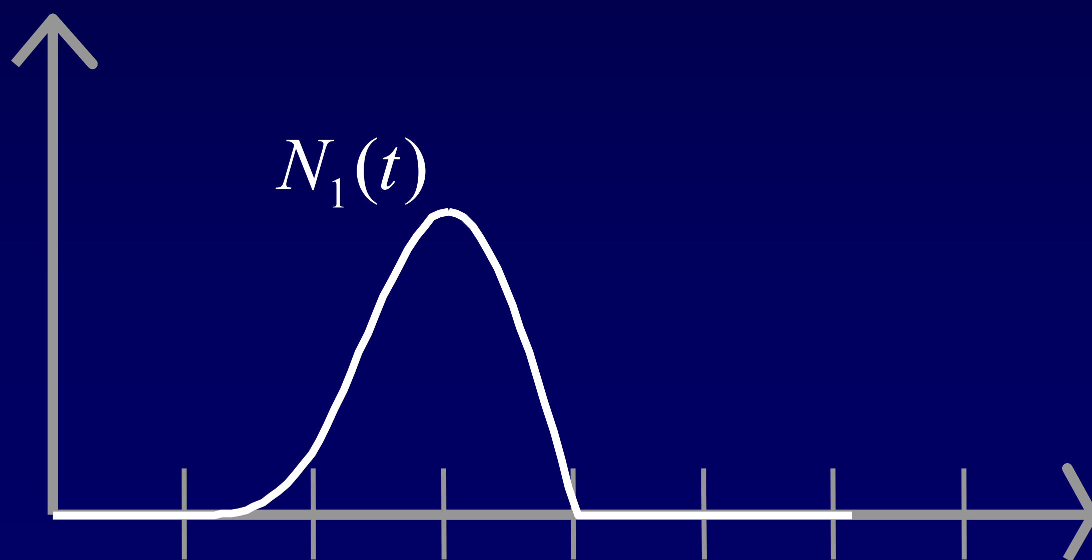
Extraordinary Vertices

$$p(t) = \begin{pmatrix} \ldots & N_1(2t+1) & N_0(2t) & N_{-1}(2t-1) & \ldots \end{pmatrix} \begin{pmatrix} & & \vdots & & & \\ & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \\ & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \\ \ldots & 0 & 0 & 1 & 0 & 0 & \ldots \\ & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \\ & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\ & & & \vdots & & & \end{pmatrix} \begin{pmatrix} \vdots \\ p_1 \\ p_0 \\ p_{-1} \\ \vdots \end{pmatrix}$$



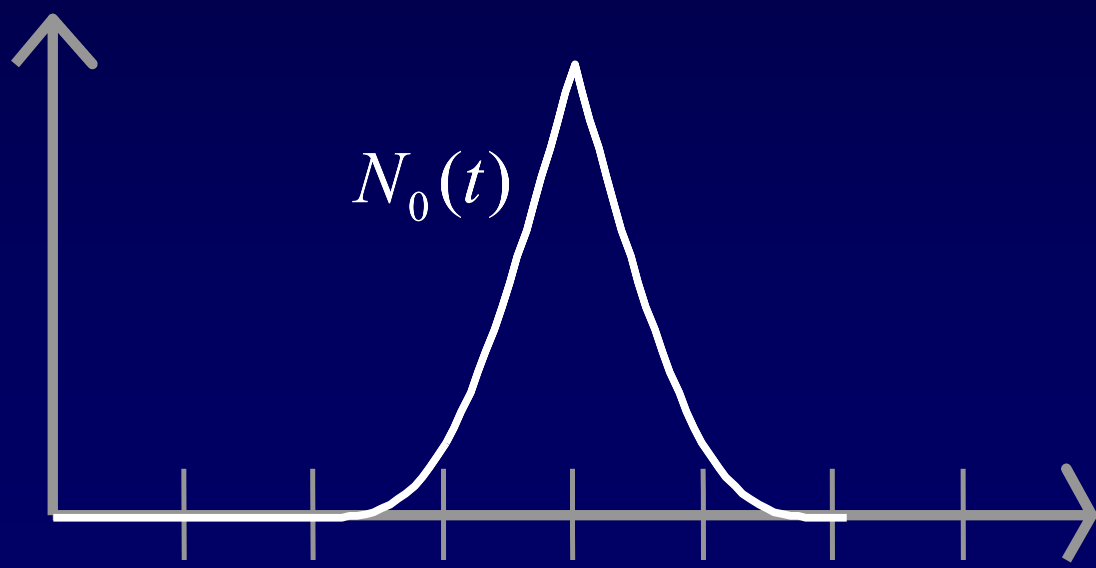
Extraordinary Vertices

$$p(t) = \begin{pmatrix} \ldots & N_1(2t+1) & N_0(2t) & N_{-1}(2t-1) & \ldots \end{pmatrix} \begin{pmatrix} & & \vdots & & & & \\ & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \\ & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \\ \ldots & 0 & 0 & 1 & 0 & 0 & \ldots \\ & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \\ & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\ & & & \vdots & & & \end{pmatrix} \begin{pmatrix} \vdots \\ p_1 \\ p_0 \\ p_{-1} \\ \vdots \end{pmatrix}$$



Extraordinary Vertices

$$p(t) = \begin{pmatrix} \ldots & N_1(2t+1) & N_0(2t) & N_{-1}(2t-1) & \ldots \end{pmatrix} \begin{pmatrix} & & & \vdots & & & \\ & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \\ & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \\ \ldots & 0 & 0 & 1 & 0 & 0 & \ldots \\ & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \\ & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\ & & & \vdots & & & \end{pmatrix} \begin{pmatrix} \vdots \\ p_1 \\ p_0 \\ p_{-1} \\ \vdots \end{pmatrix}$$

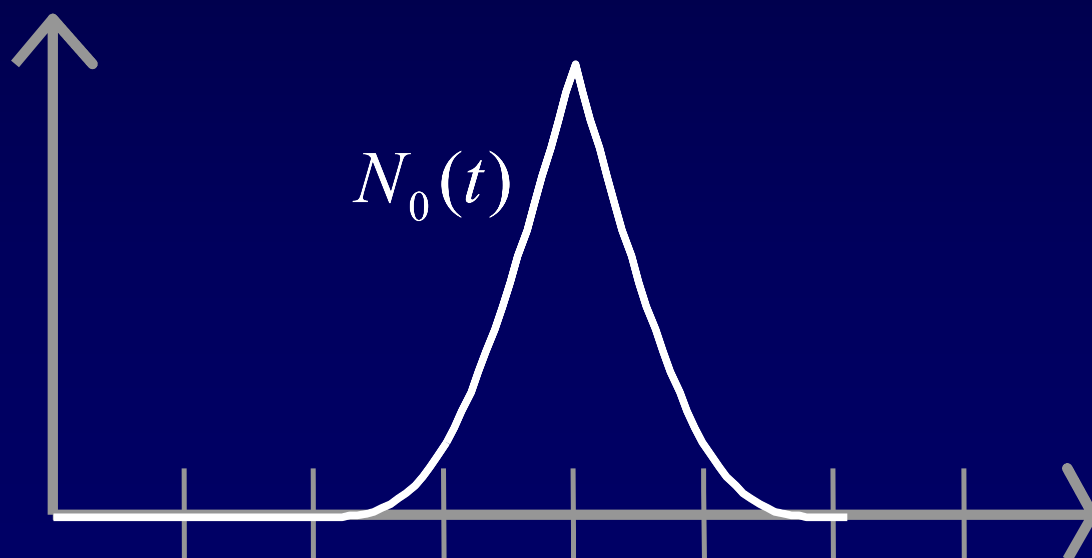


Extraordinary Vertices

$$N_0(t) = \tfrac{1}{8}N_{-2}(2t-2) + \tfrac{1}{2}N_{-1}(2t-1) + N_0(2t) + \tfrac{1}{2}N_1(2t+1) + \tfrac{1}{8}N_2(2t+2)$$

$$N_{-1}(t-1) = \tfrac{1}{8}N_{-2}(2t-4) + \tfrac{1}{2}N_{-2}(2t-3) + \tfrac{3}{4}N_{-2}(2t-2) + \tfrac{1}{2}N_{-1}(2t-1)$$

$$N_{-2}(t-2) = \tfrac{1}{8}N_{-2}(2t-6) + \tfrac{1}{2}N_{-2}(2t-5) + \tfrac{3}{4}N_{-2}(2t-4) + \tfrac{1}{2}N_{-2}(2t-3) + \tfrac{1}{8}N_{-2}(2t-2)$$

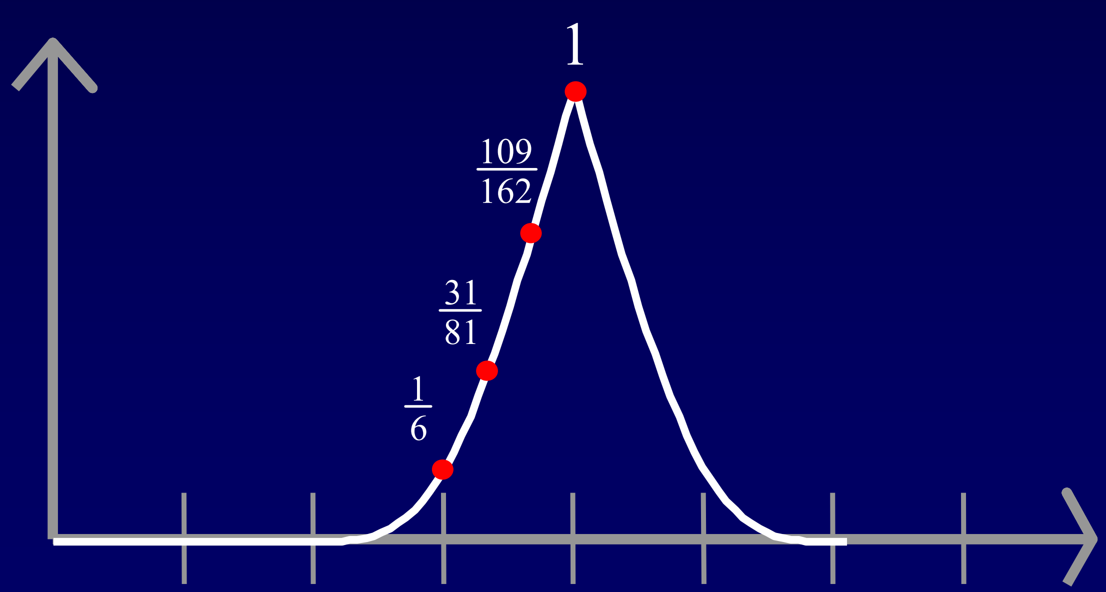


Extraordinary Vertices

$$N_0(t) = \tfrac{1}{8} N_{-2}(2t-2) + \tfrac{1}{2} N_{-1}(2t-1) + N_0(2t) + \tfrac{1}{2} N_1(2t+1) + \tfrac{1}{8} N_2(2t+2)$$

$$N_{-1}(t-1) = \tfrac{1}{8} N_{-2}(2t-4) + \tfrac{1}{2} N_{-2}(2t-3) + \tfrac{3}{4} N_{-2}(2t-2) + \tfrac{1}{2} N_{-1}(2t-1)$$

$$N_{-2}(t-2) = \tfrac{1}{8} N_{-2}(2t-6) + \tfrac{1}{2} N_{-2}(2t-5) + \tfrac{3}{4} N_{-2}(2t-4) + \tfrac{1}{2} N_{-2}(2t-3) + \tfrac{1}{8} N_{-2}(2t-2)$$

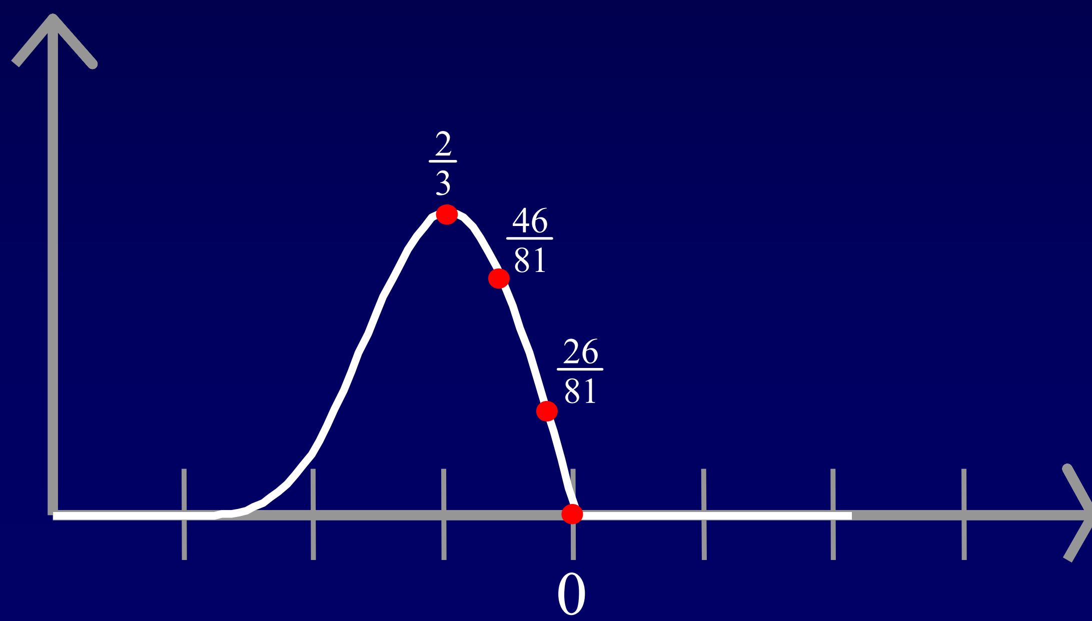


Extraordinary Vertices

$$N_0(t) = \tfrac{1}{8} N_{-2}(2t-2) + \tfrac{1}{2} N_{-1}(2t-1) + N_0(2t) + \tfrac{1}{2} N_1(2t+1) + \tfrac{1}{8} N_2(2t+2)$$

$$N_{-1}(t-1) = \tfrac{1}{8} N_{-2}(2t-4) + \tfrac{1}{2} N_{-2}(2t-3) + \tfrac{3}{4} N_{-2}(2t-2) + \tfrac{1}{2} N_{-1}(2t-1)$$

$$N_{-2}(t-2) = \tfrac{1}{8} N_{-2}(2t-6) + \tfrac{1}{2} N_{-2}(2t-5) + \tfrac{3}{4} N_{-2}(2t-4) + \tfrac{1}{2} N_{-2}(2t-3) + \tfrac{1}{8} N_{-2}(2t-2)$$

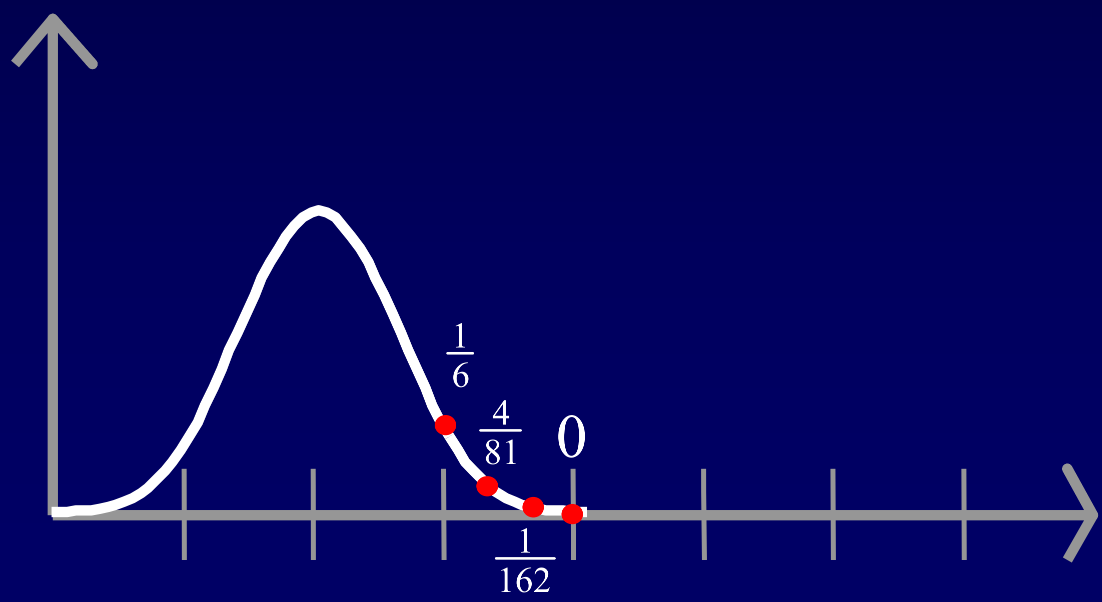


Extraordinary Vertices

$$N_0(t) = \tfrac{1}{8}N_{-2}(2t-2) + \tfrac{1}{2}N_{-1}(2t-1) + N_0(2t) + \tfrac{1}{2}N_1(2t+1) + \tfrac{1}{8}N_2(2t+2)$$

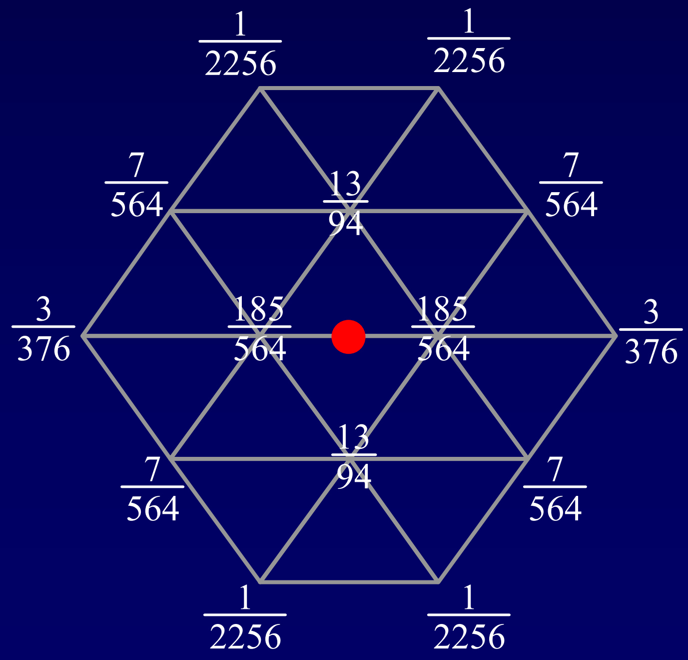
$$N_{-1}(t-1) = \tfrac{1}{8}N_{-2}(2t-4) + \tfrac{1}{2}N_{-2}(2t-3) + \tfrac{3}{4}N_{-2}(2t-2) + \tfrac{1}{2}N_{-1}(2t-1)$$

$$N_{-2}(t-2) = \tfrac{1}{8}N_{-2}(2t-6) + \tfrac{1}{2}N_{-2}(2t-5) + \tfrac{3}{4}N_{-2}(2t-4) + \tfrac{1}{2}N_{-2}(2t-3) + \tfrac{1}{8}N_{-2}(2t-2)$$



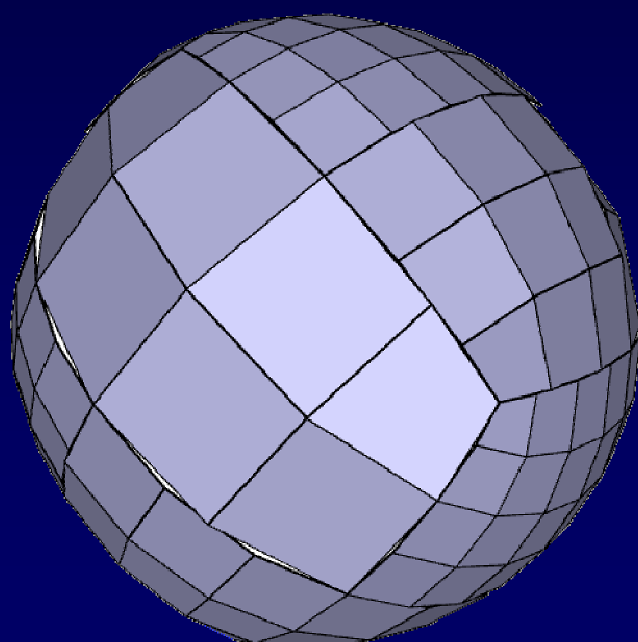
Exact Evaluation For Surfaces

- Same as for curves with more basis functions
- Take advantage of symmetry to reduce number of variables



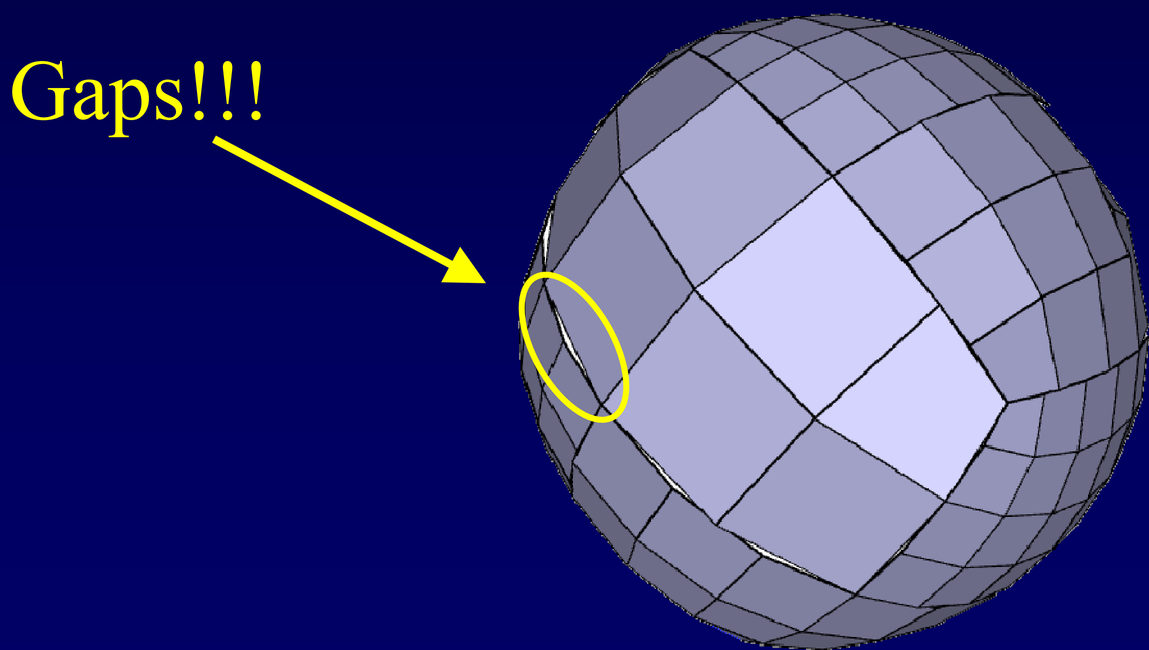
Adaptive Tessellation

- Perform evaluation on uniform grid per patch
- Choose grid resolution based on flatness, pixel size, etc…



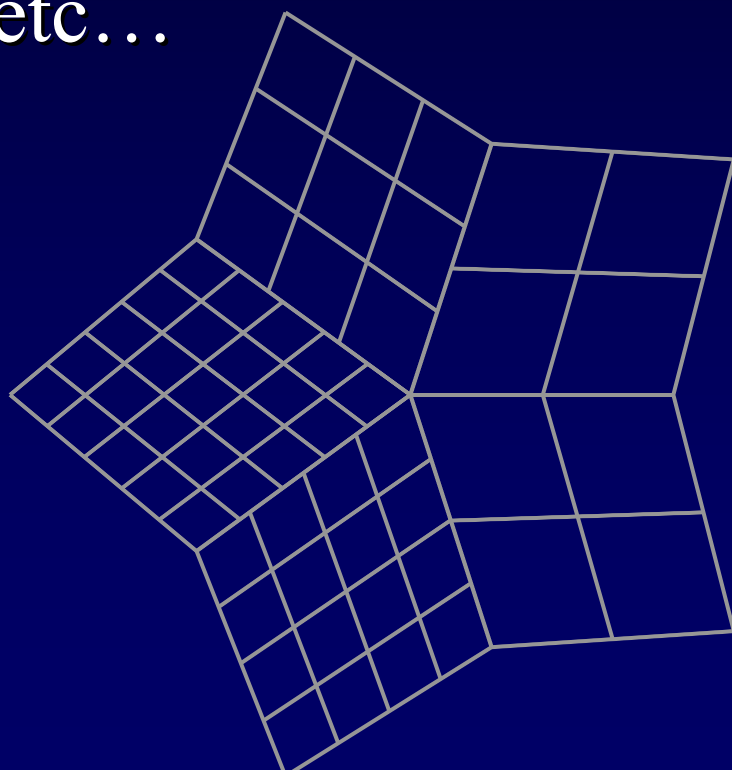
Adaptive Tessellation

- Perform evaluation on uniform grid per patch
- Choose grid resolution based on flatness, pixel size, etc…



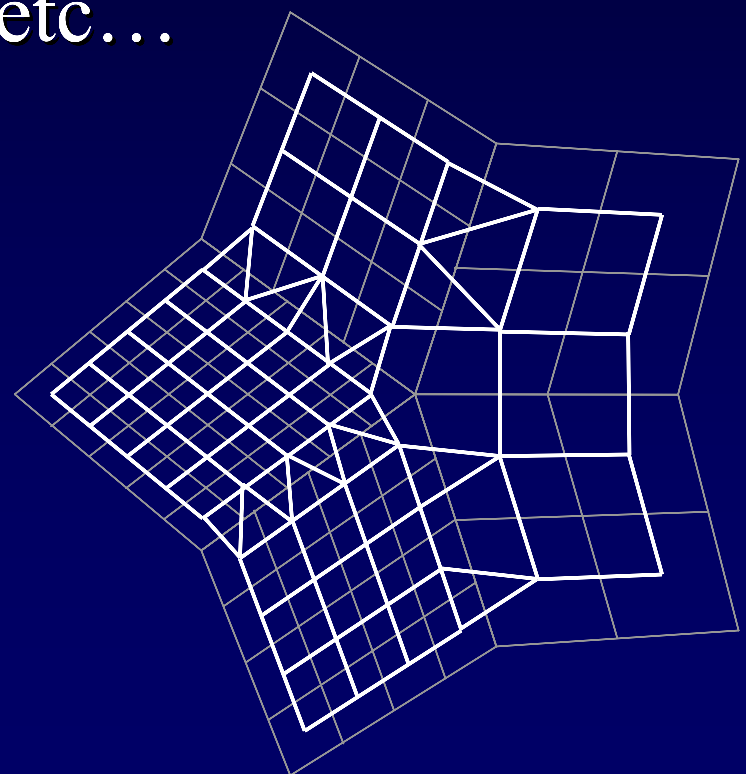
Adaptive Tessellation

- Perform evaluation on uniform grid per patch
- Choose grid resolution based on flatness, pixel size, etc…



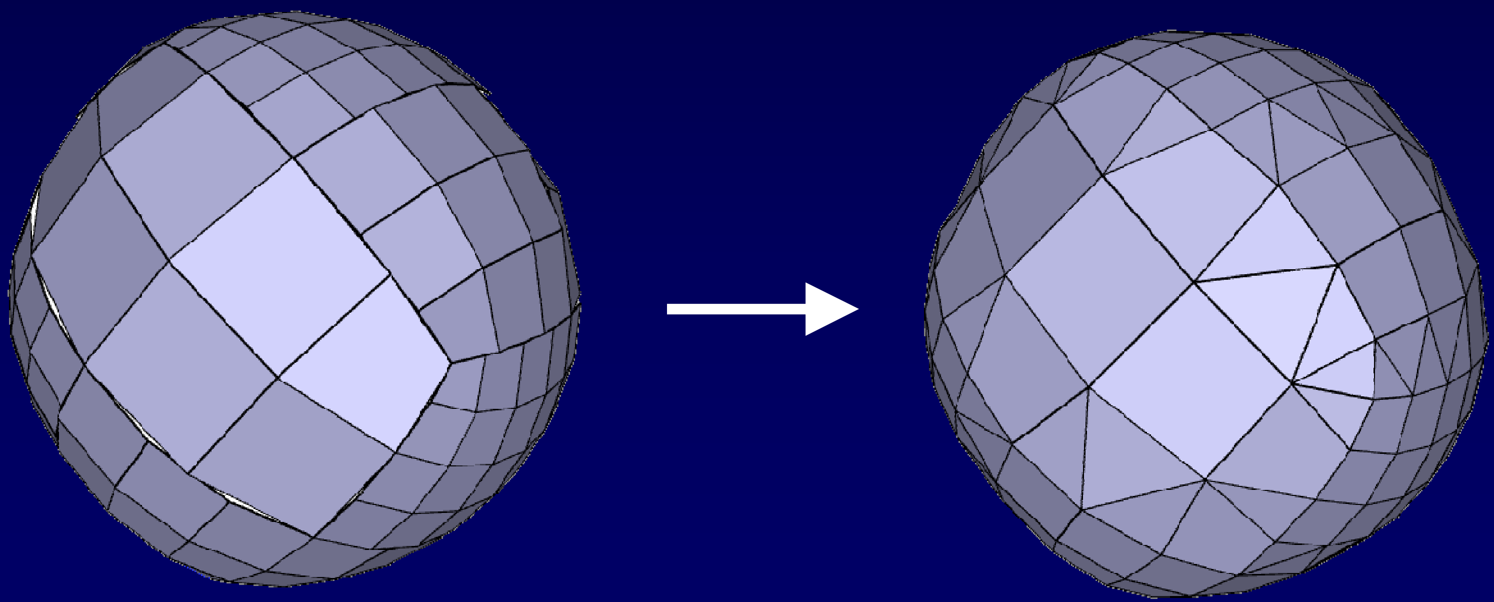
Adaptive Tessellation

- Perform evaluation on uniform grid per patch
- Choose grid resolution based on flatness, pixel size, etc…



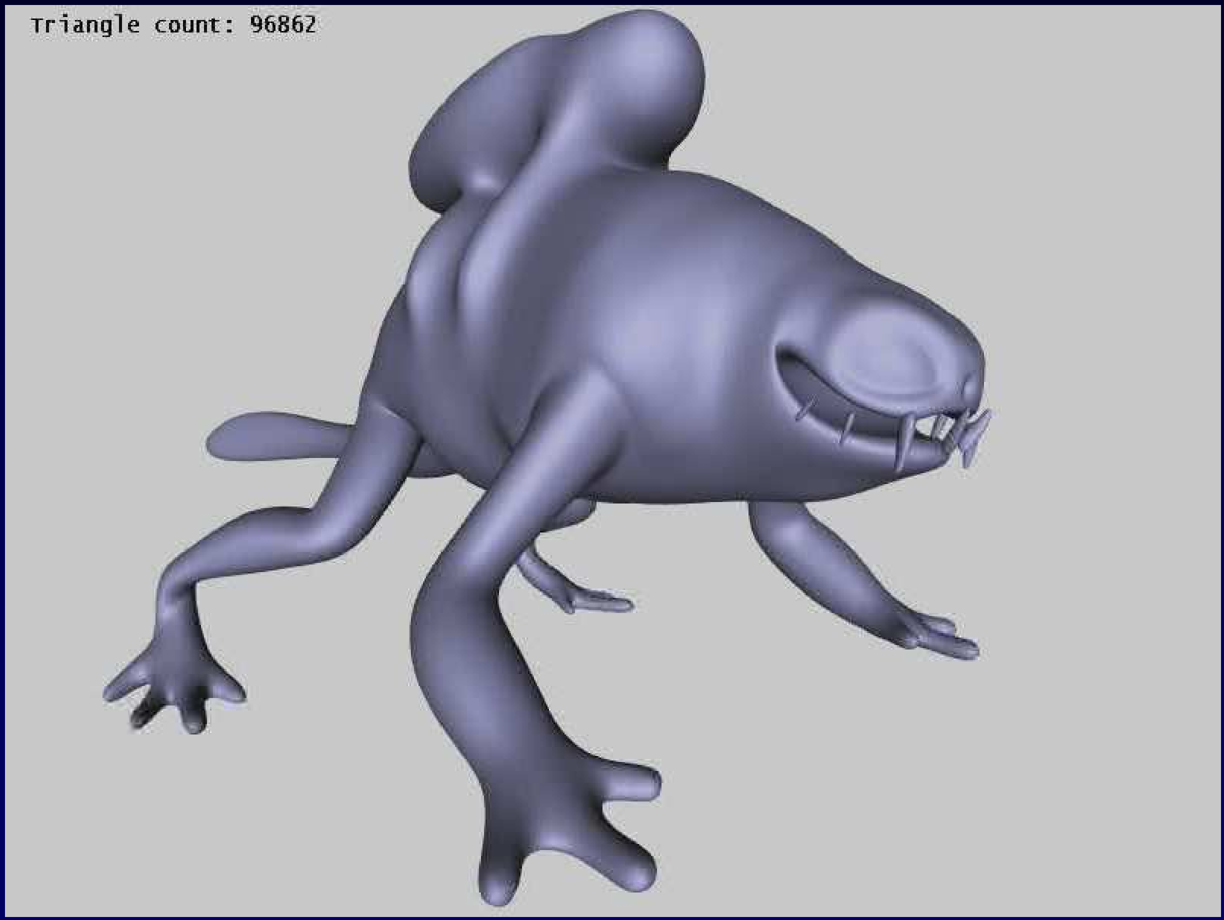
Adaptive Tessellation

- No restrictions on patch tessellation levels
- Guarantees water-tight surface
- No pixel dropouts (due to inexact arithmetic)



Results

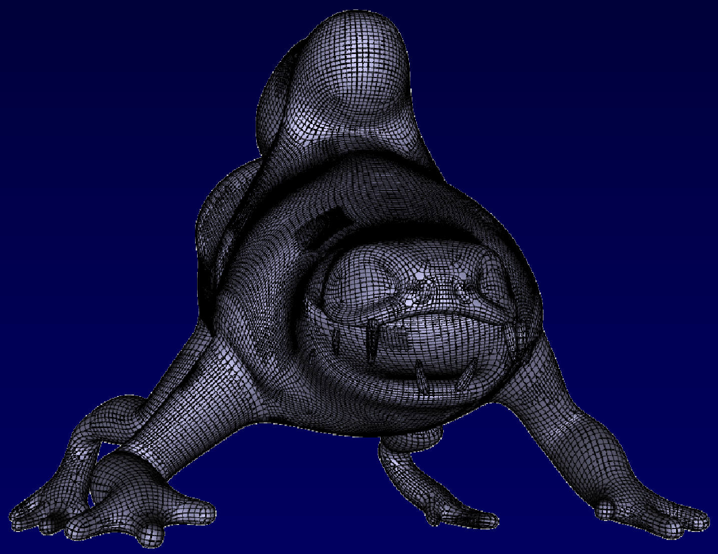
Results



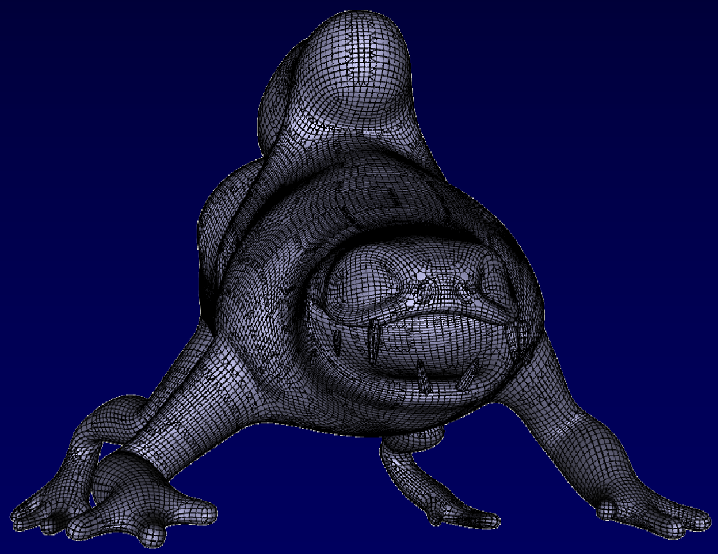
33.5 million triangles/second on CPU

Results

Subdivision Sampling

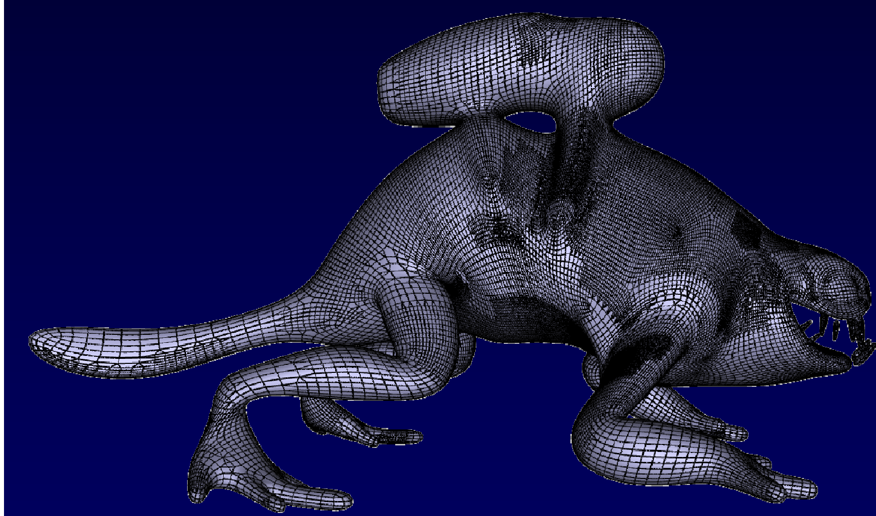


Rational Sampling



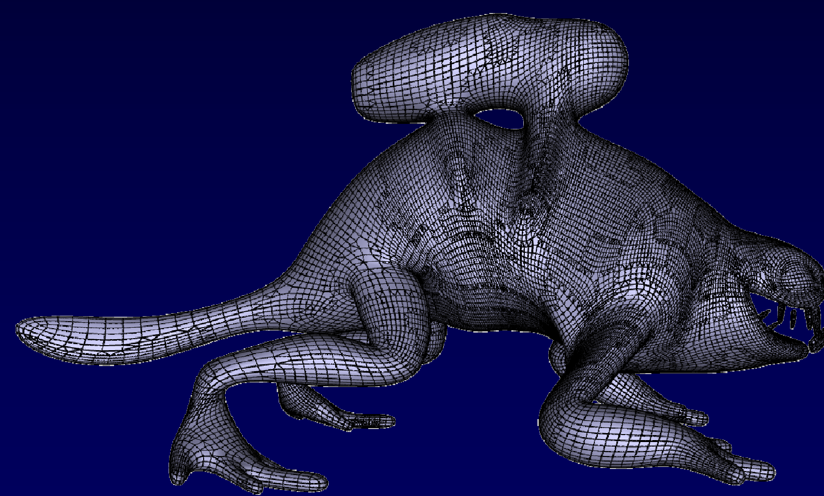
Results

Subdivision Sampling



40% more polygons

Rational Sampling



Conclusions

- Evaluation algorithm for arbitrary subdivision schemes at rational values
- Finds exact values $N\left(\frac{k}{n}\right)\ \forall k \in \mathbb{Z}$ at same time
- Fast, water-tight adaptive tessellation algorithm