计算机辅助几何设计
2023秋学期

Subdivision Curves and Surfaces

陈仁杰

中国科学技术大学

Subdivision Surfaces

Problem with Spline Patches

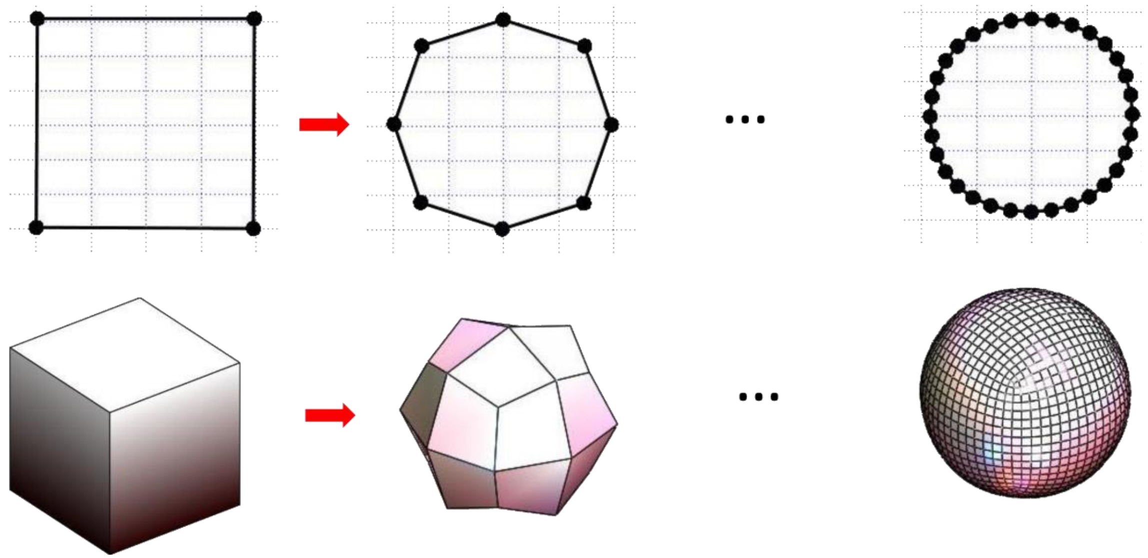
- A continuous tensor product spline surface is only defined on a regular grid of quads as parametrization domain
- Thus, the topology of the object is restricted
- Assembling multiple parameter domains to a single surface is tedious, hard to get continuity guarantees
- Handling trimming curves is not that straightforward

Question: can we do better?

Subdivision Surfaces

Wish list:

- Provide a very coarse representation of the geometry
- Obtain a fine and smooth representation
- Preferably by means of a simple set of rules which can be recursively applied (subdivision rules or subdivision scheme)



Subdivision Surfaces

Bigger goals:

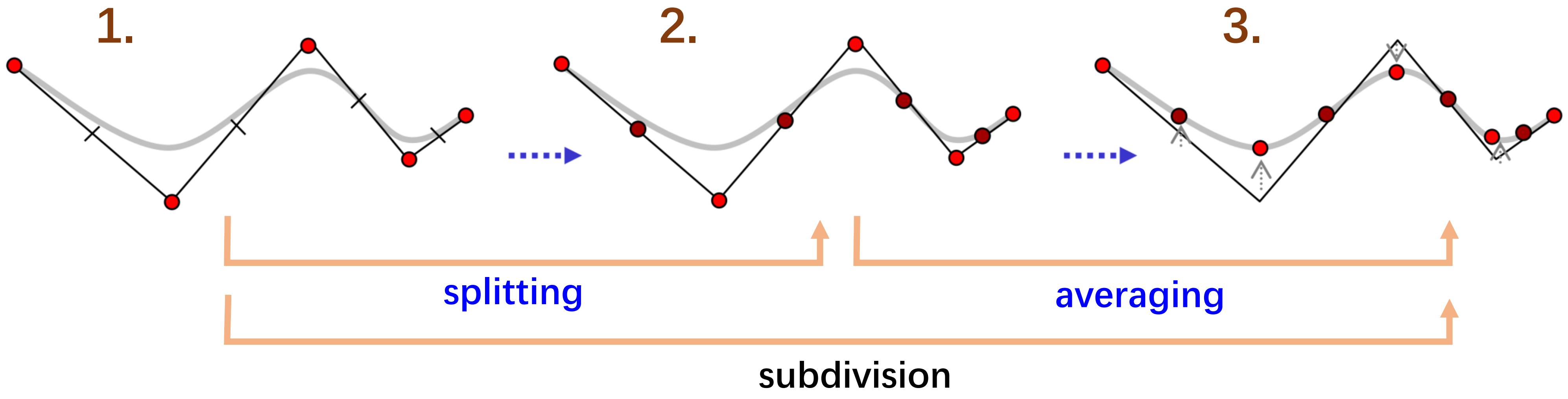
- Simplify the creation of smooth refined geometric models (especially in feature film industry)

- What's lost? Parametric representation …

Basic Scheme

Subdivision Curves & Surfaces: Three Steps

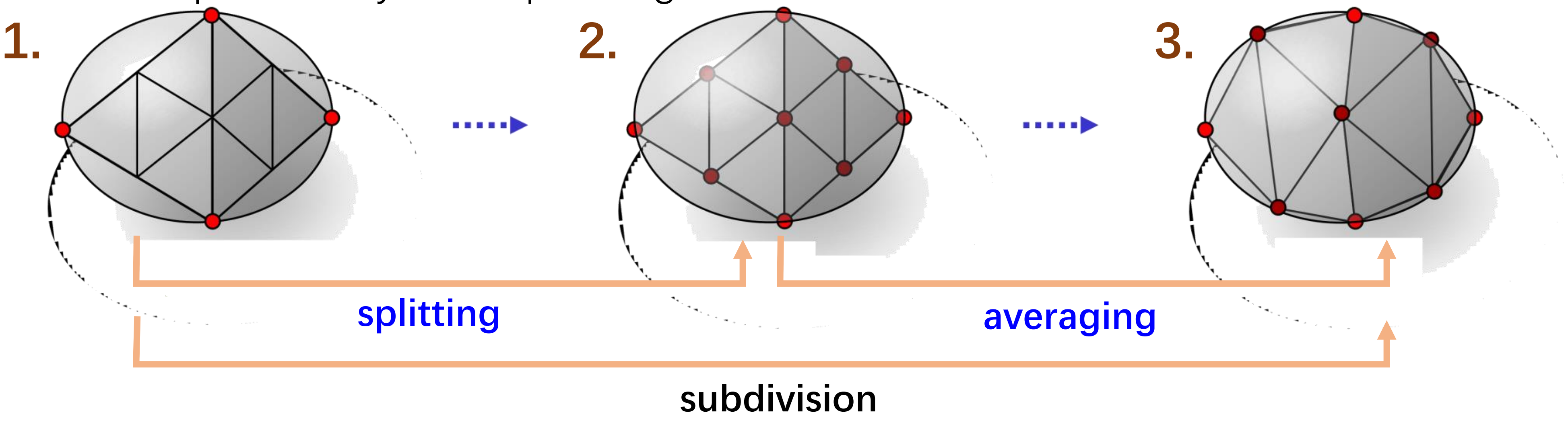
- Subdivide current polygon
- Insert linearly interpolated points (*splitting*)
- Move points: local weighted average (*averaging*)
  - To all points – approximating scheme
  - To new points only – interpolating scheme



Basic Scheme

Subdivision Curves & Surfaces: Three Steps

- Subdivide current mesh
- Insert linearly interpolated points (*splitting*)
- Move points: local weighted average (*averaging*)
  - To all points – approximating scheme
  - To new points only – interpolating scheme



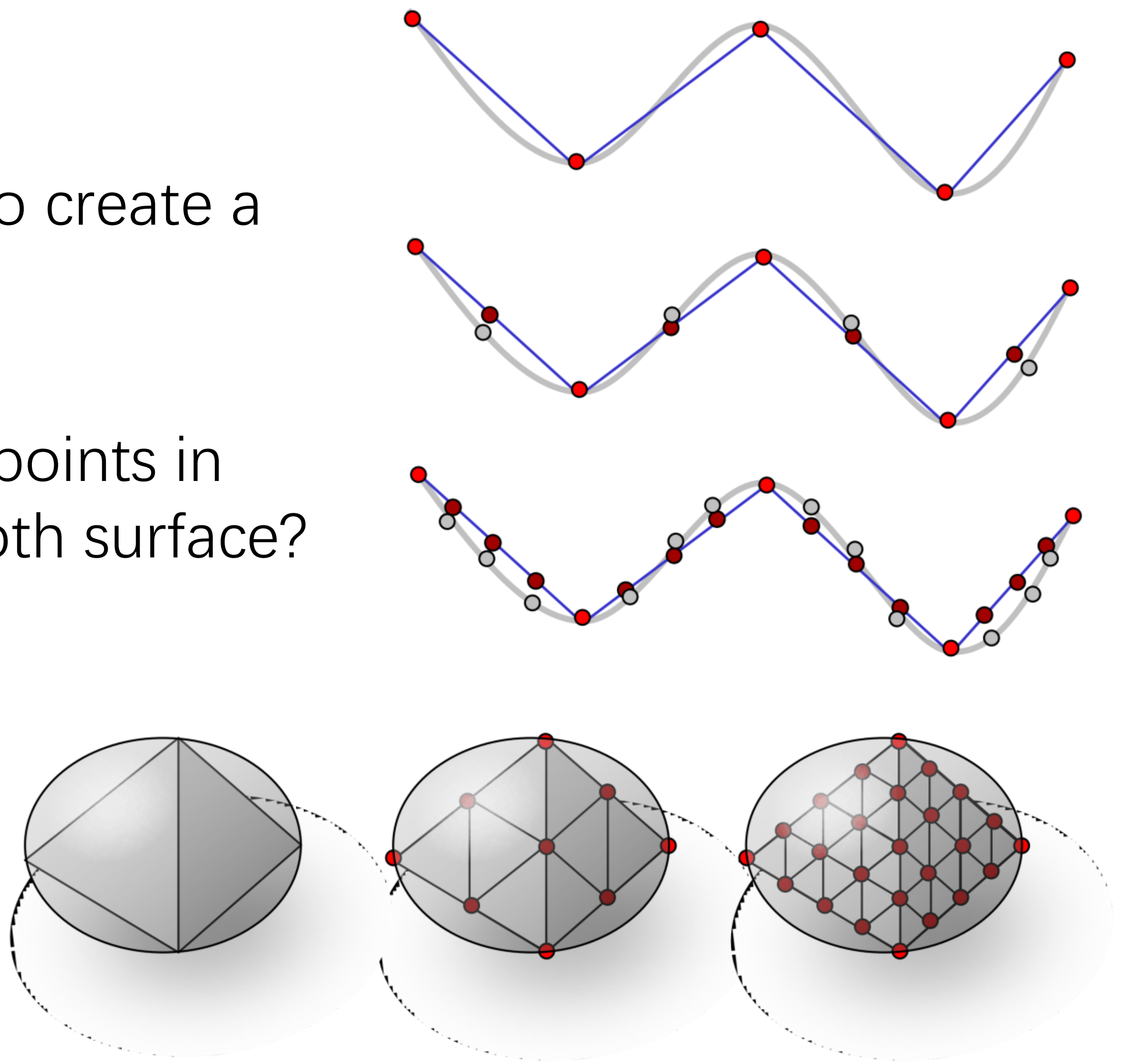
Subdivision Surfaces

The main question is:

- How should we place the new points to create a smooth surface?

  (interpolating scheme)
- Respectively: how should we alter the points in each subdivision step to create a smooth surface?

  (approximating scheme)



Subdivision Schemes

More precisely

- What are good *averaging masks*?
- The averaging mask determines the weights by which new point positions are computed

Interesting observation:

- Most averaging schemes do not converge
  (in particular interpolating schemes)
- We need to be very careful to design a good averaging mask
- How can we guarantee C^1, C^2 surfaces?

Subdivision Surfaces – History

de Rahm described a 2D (curve) subdivision scheme in 1947; rediscovered in 1974 by Chaikin

Concept extended to 3D (surface) schemes by two separate groups in 1978:

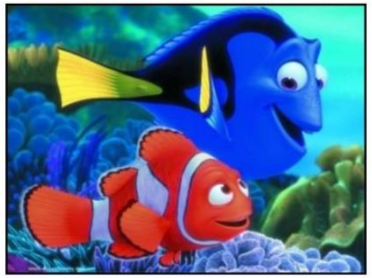
- Doo and Sabin found a biquadratic surface
- Catmull and Clark found a bicubic surface

Subsequent work in the 1980s (Loop 1987, Dyn [Butterfly subdivision] 1990) led to tools suitable for CAD/CAM and animation

Subdivision Surfaces and the Movies

Pixar first demonstrated subdivision surfaces in 1997 with Geri's Game

- Up until then they'd done everything in NURBS (Toy Story, a Bug's Life)
- From 1999 onwards, everything they did was with subdivision surfaces (Toy Story 2, Monsters Inc, Finding Nemo…)



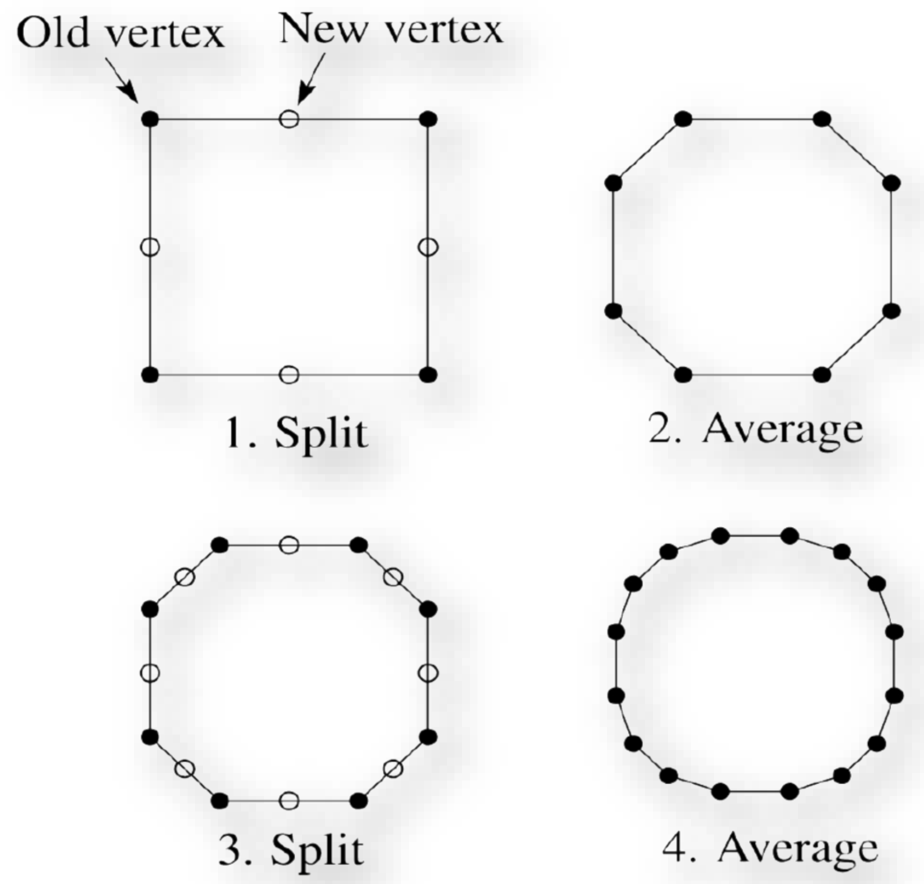
It's not clear what Dreamworks uses, but they have recent patents on subdivision techniques

Curves Revisited

Corner Cutting Splines [Chaikin 1974]:

1. Split each line segment in half
2. Average every point with its next neighbor (clock-wise)
3. Repeat

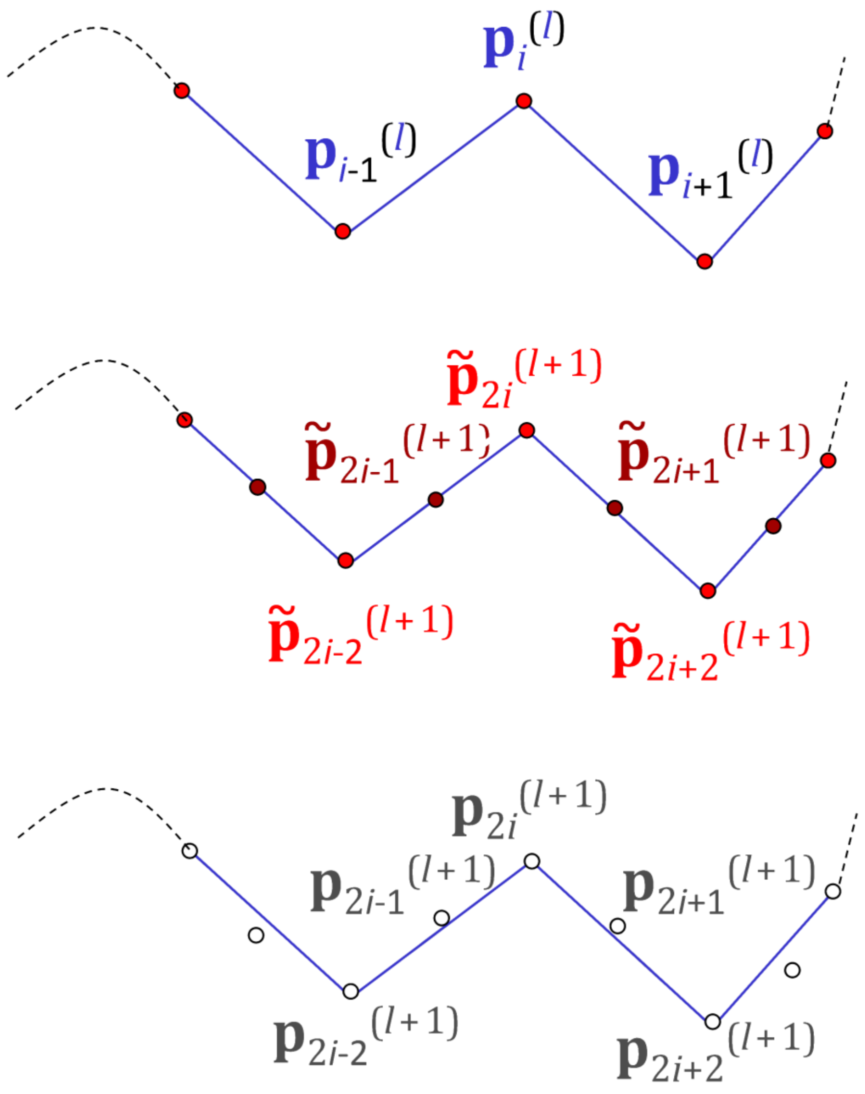
- Converges to quadratic B-Spline curve



Matrix Notation

Curve Subdivision in matrix notation:

- Control points at level l: $\boldsymbol{p}_i^{(l)}$
- "Splitted" points at level $l+1$: $\tilde{\boldsymbol{p}}_i^{(l+1)}$
- "Averaged" control points at level $l+1$: $\boldsymbol{p}_i^{(l+1)}$



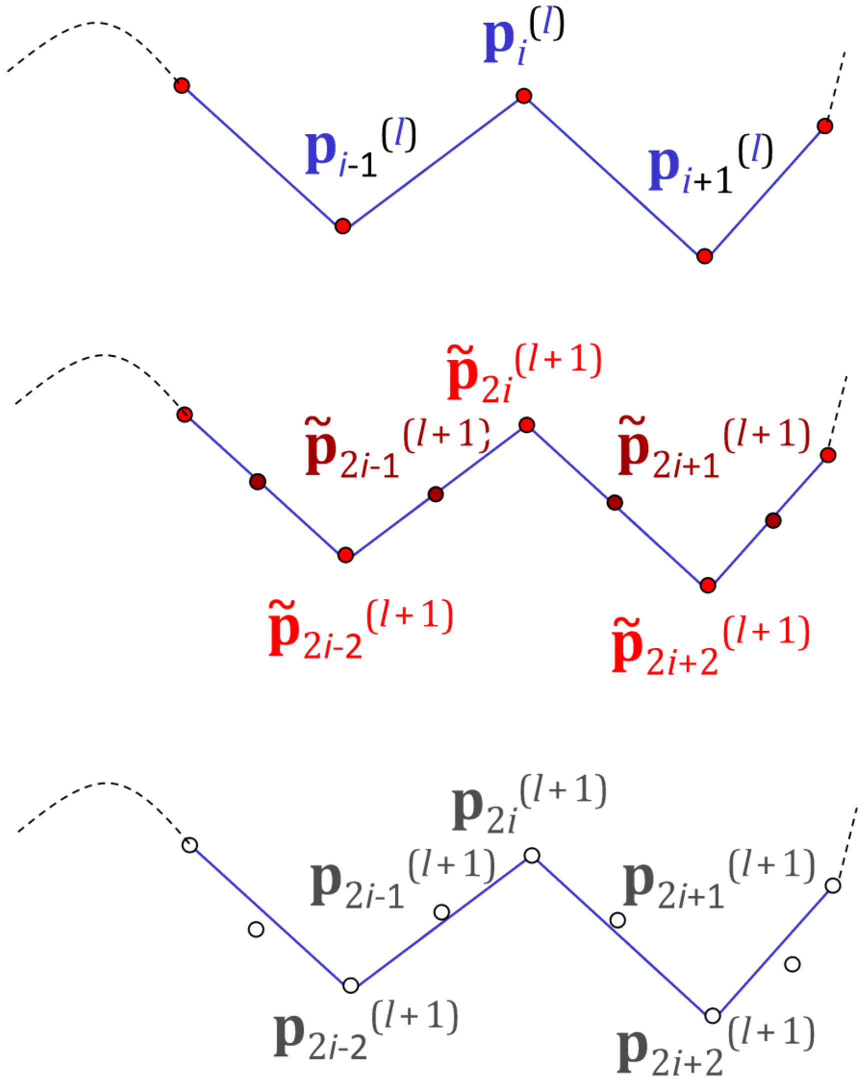
Matrix Notation

Splitting in matrix notation

$$2n\left\{\begin{pmatrix}\vdots\\ \tilde{x}_{2i}^{(l+1)}\\ \tilde{x}_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right. = 2n\left\{\underbrace{\begin{pmatrix}\ddots &&&& \\ & 1 &&& \\ & 1/2 & 1/2 && \\ && 1 && \\ && 1/2 & 1/2 & \\ &&&& \ddots\end{pmatrix}}_{n}\right. \left.\begin{pmatrix}\vdots\\ x_i^{(l)}\\ x_{i+1}^{(l)}\\ \vdots\end{pmatrix}\right\} n$$

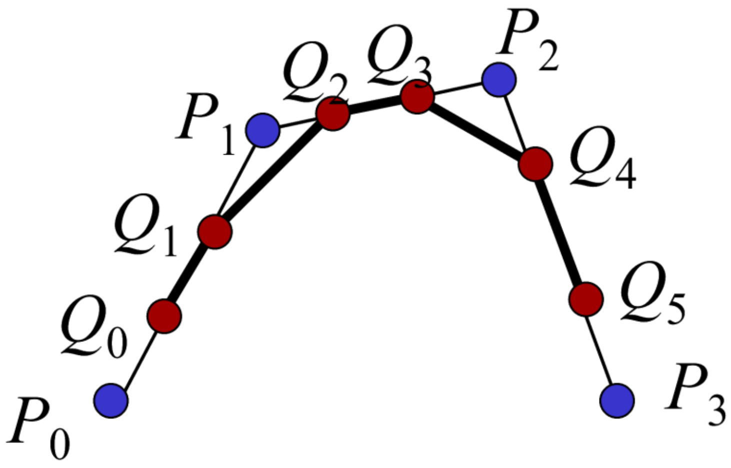
Averaging in matrix notation

$$2n\left\{\begin{pmatrix}\vdots\\ x_{2i}^{(l+1)}\\ x_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right. = 2n\left\{\underbrace{\begin{pmatrix}\ddots &&& \\ & 1/2 & 1/2 & \\ && 1/2 & 1/2 \\ &&& \ddots\end{pmatrix}}_{2n}\right. \left.\begin{pmatrix}\vdots\\ \tilde{x}_{2i}^{(l+1)}\\ \tilde{x}_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right\} 2n$$



a different view on the same algorithm…

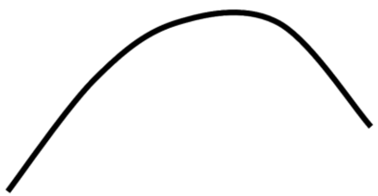
Chaikin's Corner Cutting



Apply Iterated Function System

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

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



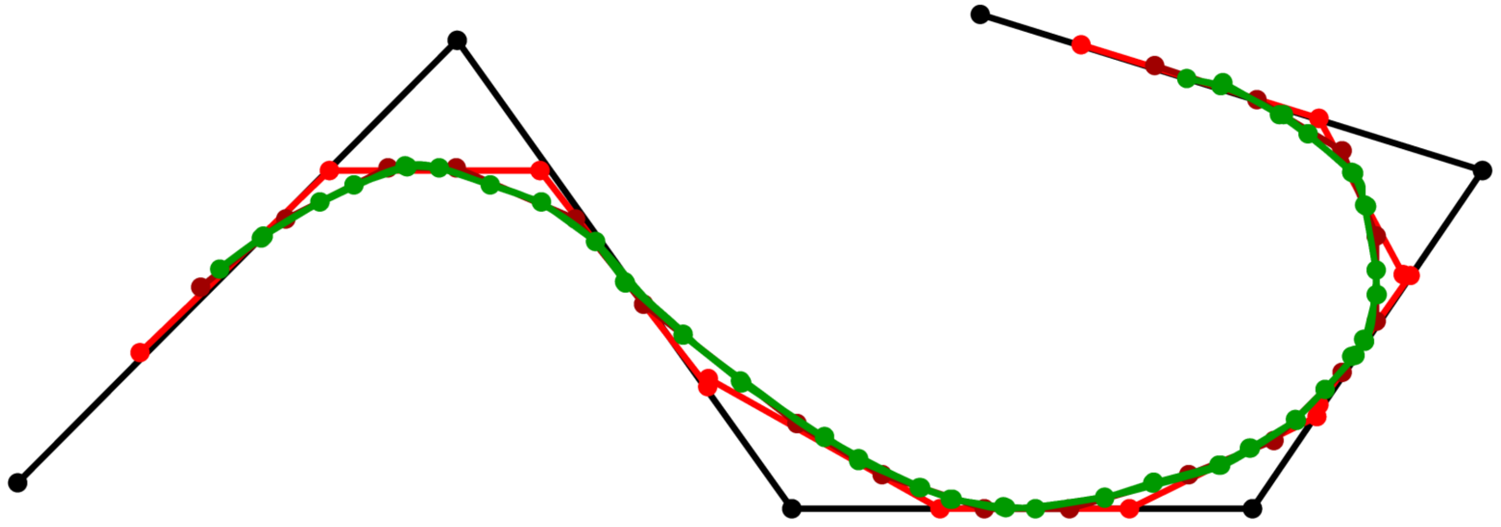
Limit Curve/Surface

- $Q_0 = \frac{3}{4}P_0 + \frac{1}{4}P_1$
- $Q_1 = \frac{1}{4}P_0 + \frac{3}{4}P_1$
- $Q_2 = \frac{3}{4}P_1 + \frac{1}{4}P_2$
- $Q_3 = \frac{1}{4}P_1 + \frac{3}{4}P_2$
- $Q_4 = \frac{3}{4}P_2 + \frac{1}{4}P_3$
- $Q_5 = \frac{1}{4}P_2 + \frac{3}{4}P_3$

Chaikin's Corner Cutting

Chaikin curve subdivision (2D)

- On each edge, insert new control points at $^1/_4$ and $^3/_4$ between old vertices; delete old points
- The *limit curve* is C^1 everywhere



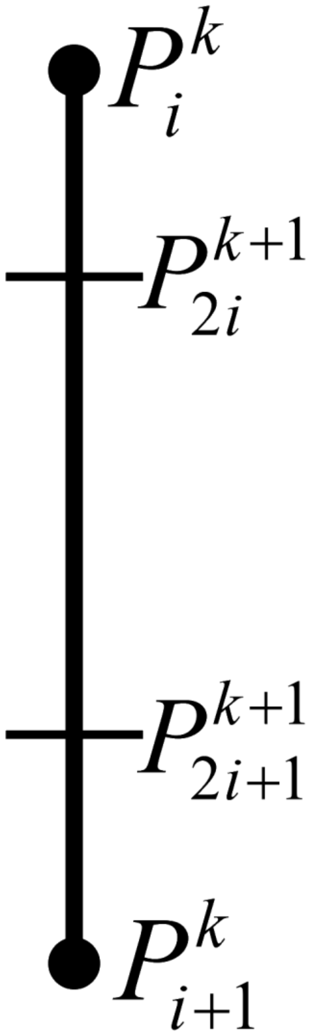
Chaikin's Corner Cutting

Chaikin can be written programmatically as

$$P_{2i}^{k+1} = ({}^{3}\!/_{4})P_i^k + ({}^{1}\!/_{4})P_{i+1}^k \qquad \leftarrow \textit{Even}$$

$$P_{2i+1}^{k+1} = ({}^{1}\!/_{4})P_i^k + ({}^{3}\!/_{4})P_{i+1}^k \qquad \leftarrow \textit{Odd}$$

- …where k is the 'generation'; each generation will have twice as many control points as before
- Notice the different treatment of generating odd and even points
- Borders (terminal points) are a special case



Chaikin's Corner Cutting

Chaikin can be written in matrix/vector notation as:

$$\begin{pmatrix} \vdots \\ P_{2i-2}^{k+1} \\ P_{2i-1}^{k+1} \\ P_{2i}^{k+1} \\ P_{2i+1}^{k+1} \\ P_{2i+2}^{k+1} \\ P_{2i+3}^{k+1} \\ \vdots \end{pmatrix} = \frac{1}{4} \begin{pmatrix} \ddots & & & & & & & \\ & 0 & 3 & 1 & 0 & 0 & 0 & \\ & 0 & 1 & 3 & 0 & 0 & 0 & \\ & 0 & 0 & 3 & 1 & 0 & 0 & \\ & 0 & 0 & 1 & 3 & 0 & 0 & \\ & 0 & 0 & 0 & 3 & 1 & 0 & \\ & 0 & 0 & 0 & 1 & 3 & 0 & \\ & & & & & & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ P_{i-2}^{k} \\ P_{i-1}^{k} \\ P_{i}^{k} \\ P_{i+1}^{k} \\ P_{i+2}^{k} \\ P_{i+3}^{k} \\ \vdots \end{pmatrix}$$

Chaikin's Corner Cutting

The standard notation compresses the scheme to a *kernel*:

- $h = ({}^1\!/_4)[\ldots, 0,0,1,3,3,1,0,0, \ldots]$

The kernel interlaces the odd and even rules

It also makes matrix analysis possible: eigen-analysis of the matrix form can be used to prove the continuity of the subdivision limit surface

The limit curve of Chaikin is a quadratic B-spline!

Cubic B-Spline Subdivision Scheme

Lane-Riesenfeld subdivision

Algorithm:

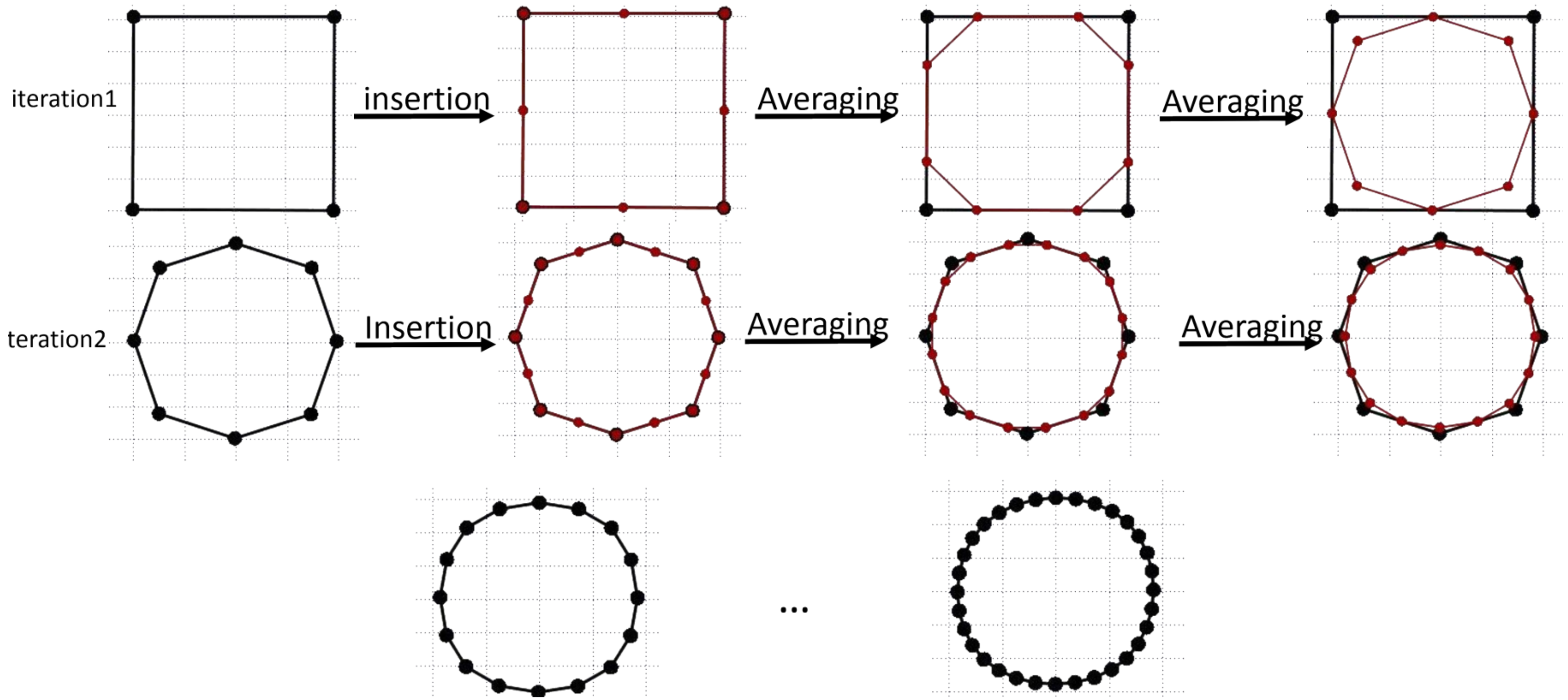
- Linearly subdivide the curve by inserting the midpoint on each edge
- Perform Averaging by replacing each edge by its midpoint d times

- Let's examine the case of $d = 2$

Lane-Riesenfeld subdivision

Examples:

- Closed curve



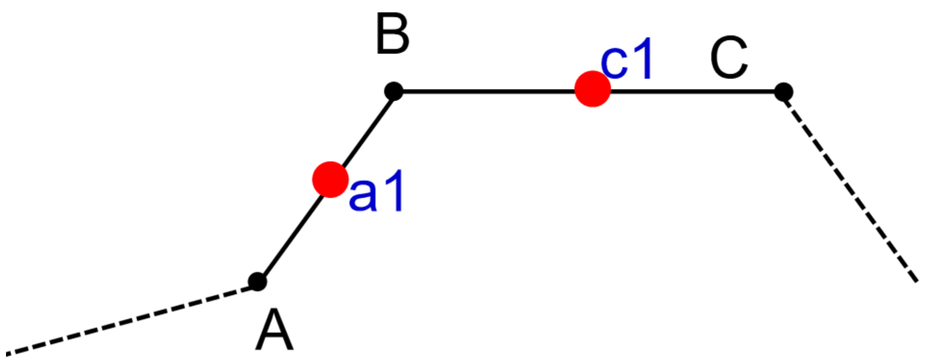
Lane-Riesenfeld subdivision

Close examination

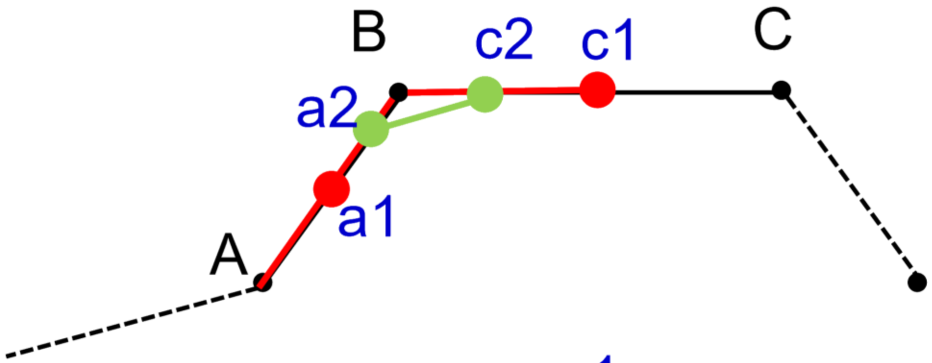
- Step by step

$$\begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} = \frac{1}{8}\begin{pmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{pmatrix}\begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

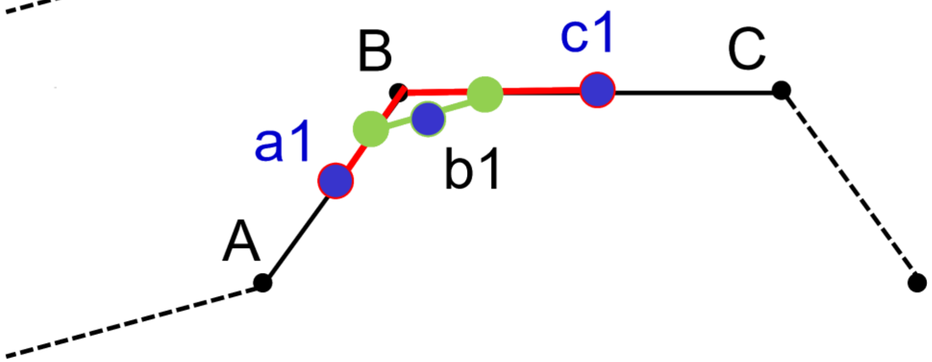
$$a_1 = \frac{A+B}{2}$$
$$c_1 = \frac{B+C}{2}$$



$$a_2 = \frac{B+a_1}{2}$$
$$c_2 = \frac{B+c_1}{2}$$



$$b_1 = \frac{a_2+c_2}{2} = \frac{a_1+2B+c_1}{4} = \frac{A+6B+C}{8}$$

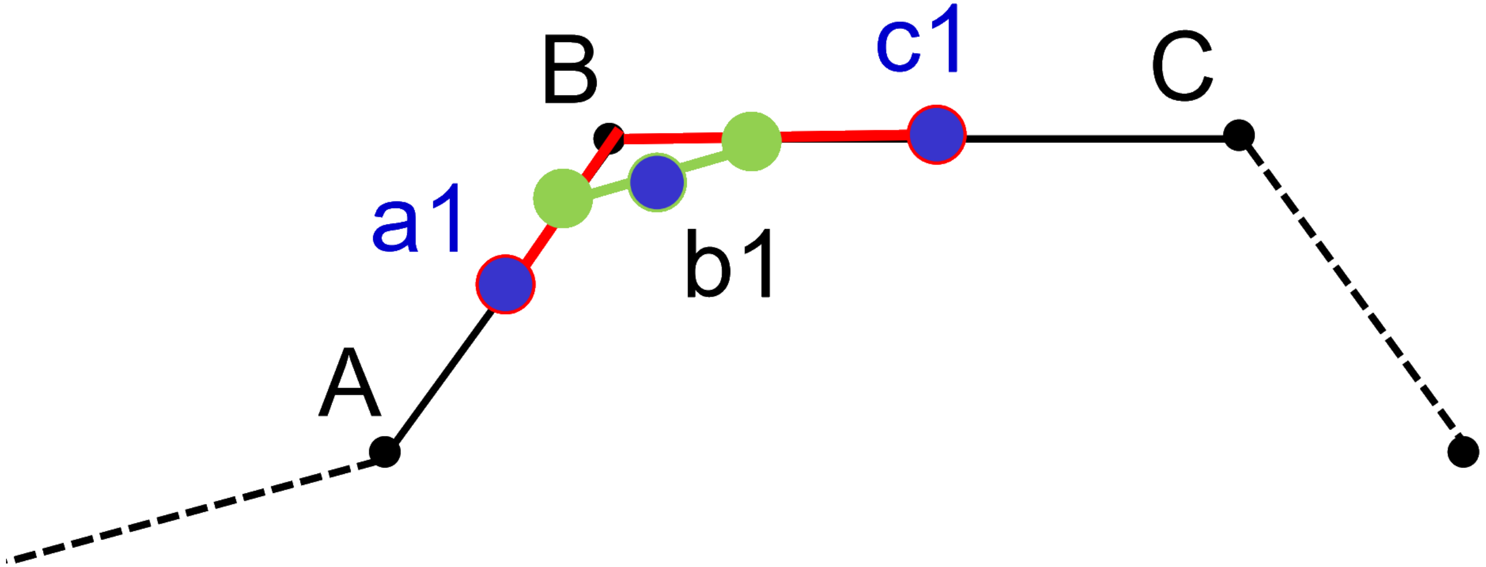


Lane-Riesenfeld subdivision

Close examination:

- In matrix form

$$\begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} = \frac{1}{8}\begin{pmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{pmatrix}\begin{pmatrix} A \\ B \\ C \end{pmatrix}$$



Separate Splitting Step

Using a separate splitting matrix

$$\begin{pmatrix} \vdots \\ \boldsymbol{p}_{2i}^{(l+1)} \\ \boldsymbol{p}_{2i+1}^{(l+1)} \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} \ddots & & & & & \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & \\ & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & \\ & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \\ & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ & & & & & \ddots \end{pmatrix}}_{2n \times 2n \text{ averaging}} \underbrace{\begin{pmatrix} \ddots & & & \\ & 1 & & \\ & \frac{1}{2} & \frac{1}{2} & \\ & & 1 & \\ & & \frac{1}{2} & \frac{1}{2} \\ & & & \ddots \end{pmatrix}}_{2n \times n \text{ spliting}} \begin{pmatrix} \vdots \\ \boldsymbol{p}_{i}^{(l)} \\ \boldsymbol{p}_{i+1}^{(l)} \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} \vdots \\ \boldsymbol{p}_{2i}^{(l+1)} \\ \boldsymbol{p}_{2i+1}^{(l+1)} \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} \ddots & & & & \\ & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\ & & \frac{1}{2} & \frac{1}{2} & \\ & & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ & & & \frac{1}{2} & \frac{1}{2} \\ & & & & \ddots \end{pmatrix}}_{\text{One step}} \begin{pmatrix} \vdots \\ \boldsymbol{p}_{i}^{(l)} \\ \boldsymbol{p}_{i+1}^{(l)} \\ \vdots \end{pmatrix}$$

$$\boldsymbol{p}_{2i}^{[l+1]} = \boxed{\frac{1}{4}\boldsymbol{p}_{i}^{[l]} + \frac{1}{2}\left(\frac{1}{2}\boldsymbol{p}_{i}^{[l]} + \frac{1}{2}\boldsymbol{p}_{i+1}^{[l]}\right) + \frac{1}{4}\boldsymbol{p}_{i+1}^{[l]}} = \frac{1}{2}\boldsymbol{p}_{i}^{[l]} + \frac{1}{2}\boldsymbol{p}_{i+1}^{[l]}$$

$$\boldsymbol{p}_{2i+1}^{[l+1]} = \boxed{\frac{1}{4}\left(\frac{1}{2}\boldsymbol{p}_{i}^{[l]} + \frac{1}{2}\boldsymbol{p}_{i+1}^{[l]}\right) + \frac{1}{2}\boldsymbol{p}_{i+1}^{[l]} + \frac{1}{4}\left(\frac{1}{2}\boldsymbol{p}_{i+1}^{[l]} + \frac{1}{2}\boldsymbol{p}_{i+2}^{[l]}\right)} = \frac{1}{8}\boldsymbol{p}_{i}^{[l]} + \frac{6}{8}\boldsymbol{p}_{i+1}^{[l]} + \frac{1}{8}\boldsymbol{p}_{i+2}^{[l]}$$

Separate Splitting Step

Using a separate splitting matrix

$$\begin{pmatrix}\vdots\\ \boldsymbol{p}_{2i}^{(l+1)}\\ \boldsymbol{p}_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix} = \underbrace{\begin{pmatrix}\ddots &&&&&\\ \frac{1}{4}&\frac{1}{2}&\frac{1}{4}&&&\\ &\frac{1}{4}&\frac{1}{2}&\frac{1}{4}&&\\ &&\frac{1}{4}&\frac{1}{2}&\frac{1}{4}&\\ &&&\frac{1}{4}&\frac{1}{2}&\frac{1}{4}\\ &&&&&\ddots\end{pmatrix}}_{2n\times 2n\text{ averaging}} \underbrace{\begin{pmatrix}\ddots&&&\\ &1&&\\ &\frac{1}{2}&\frac{1}{2}&\\ &&1&\\ &&\frac{1}{2}&\frac{1}{2}\\ &&&\ddots\end{pmatrix}}_{2n\times n\text{ spliting}} \begin{pmatrix}\vdots\\ \boldsymbol{p}_{i}^{(l)}\\ \boldsymbol{p}_{i+1}^{(l)}\\ \vdots\end{pmatrix}$$

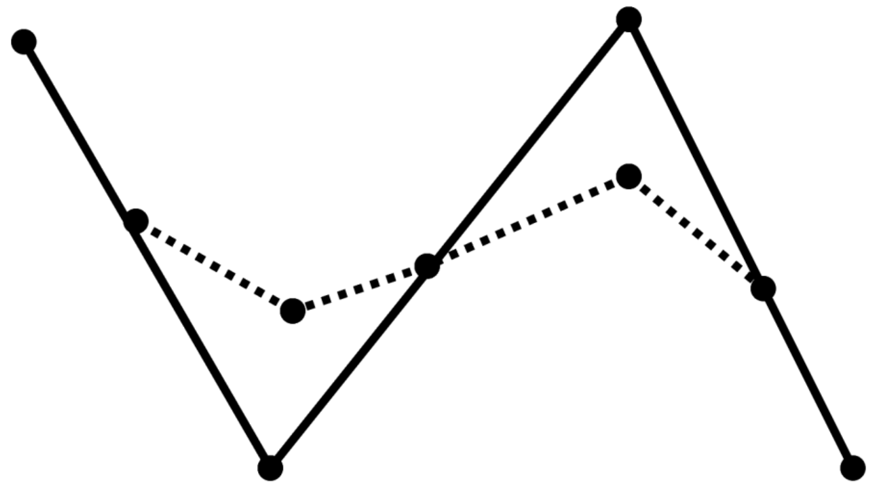
Cubic Subdivision

Consider the Kernel

- $h = \left(\frac{1}{8}\right)[\ldots, 0,0,1,4,6,4,1,0,0, \ldots]$

You would read this as

- $P_{2i}^{k+1} = ({}^{1}/_{8})\left(P_{i-1}^{k} + 6P_{i}^{k} + P_{i+1}^{k}\right)$
- $P_{2i+1}^{k+1} = ({}^{1}/_{8})\left(4P_{i}^{k} + 4P_{i+1}^{k}\right)$



The limit curve is provably C^2 continuous

General Formula:

B-spline curve subdivision:

- Splitting step as usual (insert midpoints on lines)
- Averaging mask is stationary (constant everywhere):

$$\frac{1}{2^{d-1}}\left(\binom{d-1}{0},\binom{d-1}{1},\dots,\binom{d-1}{d-1}\right)$$

for B-splines of degree d

Approximating the curve

- Infinite subdivision will create a dense point set that converges to the curve

Spectral Convergence Analysis

of the cubic B-Spline Subdivision Scheme

The Spectral Limit Trick

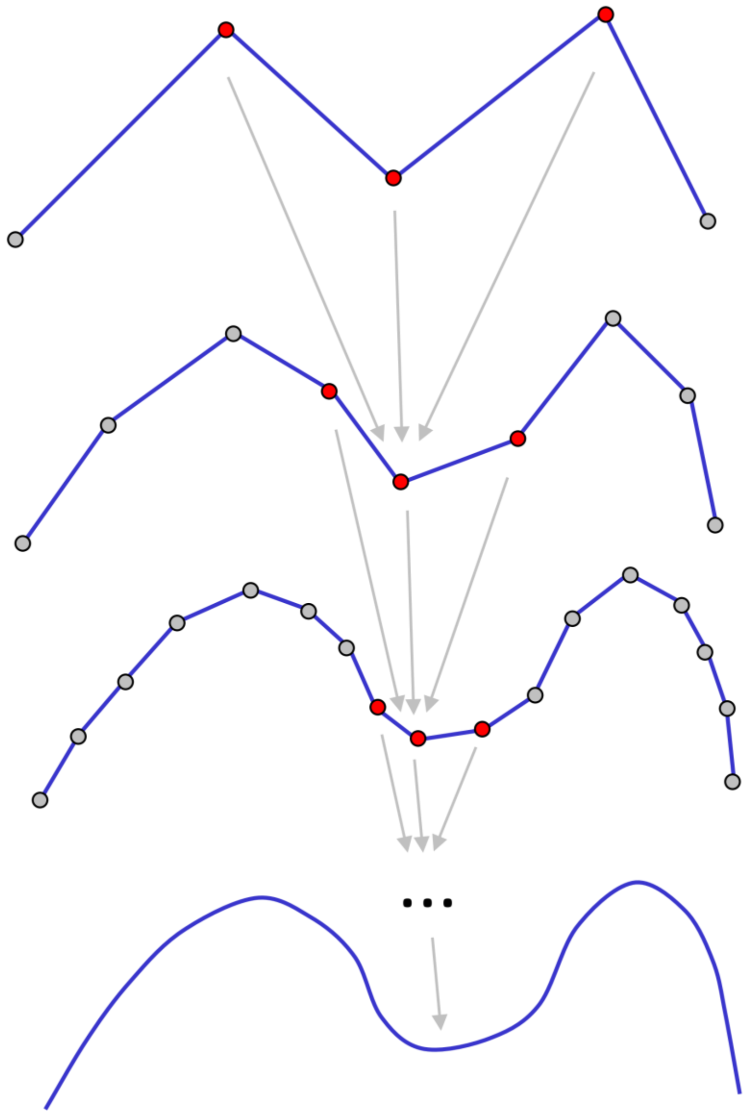
Problem:

- We need to subdivide several times to obtain a good approximation
- This might yield more control points than necessary
  (think of adaptive rendering with low level of detail)
- Can we directly compute the limit position for a control points?

Computing the Limit

Observations:

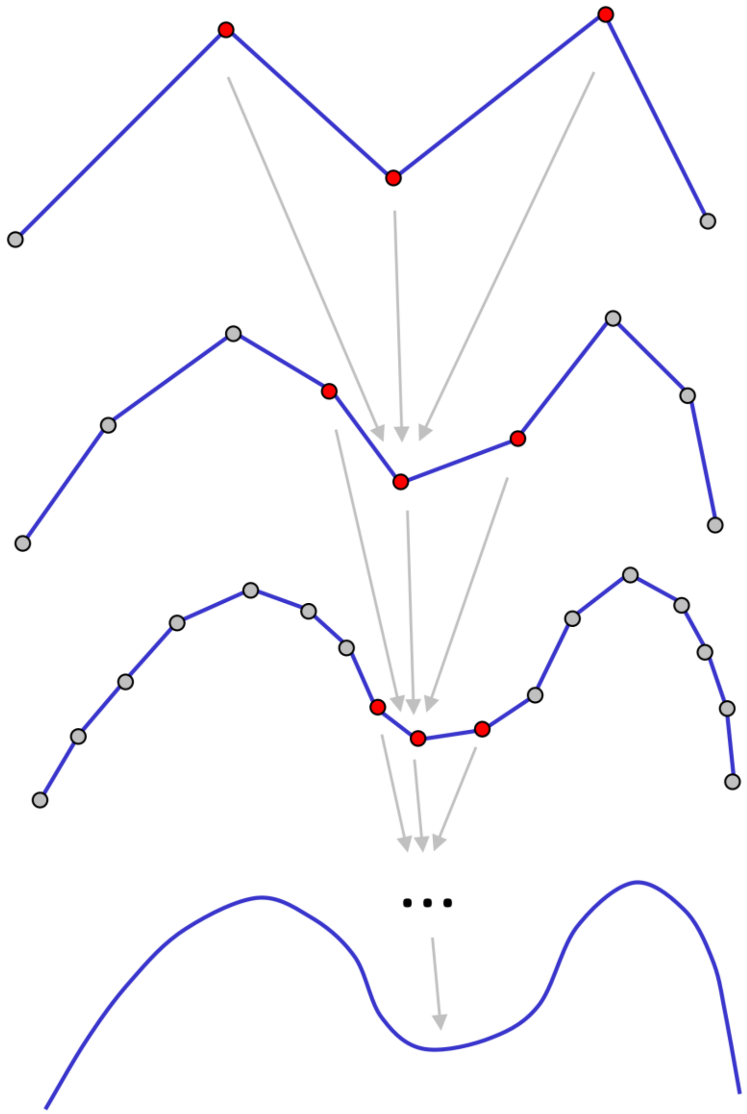
- Every curve point is influenced only by a fixed number of control points
- Even stronger : Every point $p^{[l+1]}$ is only influenced by a small neighborhood of points in $p^{[l]}$
- To each neighborhood, the same subdivision matrix is applied (splitting & averaging)



The Local Subdivision Matrix

Invariant Neighborhood

- Example: Cubic B-splines
  - A single point lies in one of two adjacent spline segments
  - So at most 5 control points are influencing each point on the curve
  - A closer look at the subdivision rule reveals that limit properties can actually be computed from 3 points (two direct neighbors)



Local Subdivision Matrix

Local subdivision matrix:

- Transforms a neighborhood of points

Example: cubic B-spline

- Only the two direct neighbors influence the point in the next level
- The local subdivision matrix is

x_- = left neighbor
x = point ($x/y/z$-coordinate)
x_+ = right neighbor

$$\begin{pmatrix} x_-^{[l+1]} \\ x^{[l+1]} \\ x_+^{[l+1]} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}}_{= M_{subdiv}} \begin{pmatrix} x_-^{[l]} \\ x^{[l]} \\ x_+^{[l]} \end{pmatrix}$$

To the Limit…

This means:

- At any recursion depth of the subdivision, we can send a point to the limit by evaluating:

$$\begin{pmatrix} x_-^{[\infty]} \\ x^{[\infty]} \\ x_+^{[\infty]} \end{pmatrix} = \lim_{k\to\infty} \boldsymbol{M}_{subdiv}^{k} \begin{pmatrix} x_-^{[l]} \\ x^{[l]} \\ x_+^{[l]} \end{pmatrix} = \lim_{k\to\infty} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}^{k} \begin{pmatrix} x_-^{[l]} \\ x^{[l]} \\ x_+^{[l]} \end{pmatrix}$$

To the Limit⋯

Spectral power:

- Assuming the matrix $\boldsymbol{M}_{subdiv}$ is diagonizable, we get:

$$\begin{pmatrix} x_-^{[\infty]} \\ x^{[\infty]} \\ x_+^{[\infty]} \end{pmatrix} = \lim_{k\to\infty} \boldsymbol{U}\boldsymbol{D}^{\boldsymbol{k}}\boldsymbol{U}^{-\mathbf{1}} \begin{pmatrix} x_-^{[l]} \\ x^{[l]} \\ x_+^{[l]} \end{pmatrix} = U\left(\lim_{k\to\infty} D^k\right)U^{-1} \begin{pmatrix} x_-^{[l]} \\ x^{[l]} \\ x_+^{[l]} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & -1 & -2 \\ 1 & 0 & 1 \\ 1 & 1 & -2 \end{pmatrix} \lim_{k\to\infty} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix}^k \begin{pmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \end{pmatrix} \begin{pmatrix} x_-^{[l]} \\ x^{[l]} \\ x_+^{[l]} \end{pmatrix}$$

To the Limit…

Spectral power:

- For cubic B-splines:
- $$\begin{pmatrix} x_-^{[\infty]} \\ x^{[\infty]} \\ x_+^{[\infty]} \end{pmatrix} = \begin{pmatrix} 1 & -1 & -2 \\ 1 & 0 & 1 \\ 1 & 1 & -2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \end{pmatrix} \begin{pmatrix} x_-^{[1]} \\ x^{[1]} \\ x_+^{[1]} \end{pmatrix} = \begin{pmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} x_-^{[1]} \\ x^{[1]} \\ x_+^{[1]} \end{pmatrix}$$
- and hence

$$x^{[\infty]} = \left[\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right] \begin{pmatrix} x_-^{[1]} \\ x^{[1]} \\ x_+^{[1]} \end{pmatrix}$$

To the Limit, in General

- **In general:**
  - The dominant eigenvalue / eigenvector of the subdivision scheme determines the limit mask

Necessary Condition

Necessary condition for convergence:

- 1 must be the largest eigenvalue (in absolute value)
- Otherwise the subdivision either explodes (>1) or shrinks to the origin (<1)

$$\begin{pmatrix} x_{-n}^{[l+k]} \\ \vdots \\ x_0^{[l+k]} \\ \vdots \\ x_{+n}^{[l+k]} \end{pmatrix} = \boldsymbol{M}_{subdiv}^{k} \begin{pmatrix} x_{-n}^{[l]} \\ \vdots \\ x_0^{[l]} \\ \vdots \\ x_{+n}^{[l]} \end{pmatrix} = \boldsymbol{U}\boldsymbol{D}^{k}\boldsymbol{U}^{-1} \begin{pmatrix} x_{-n}^{[l]} \\ \vdots \\ x_0^{[l]} \\ \vdots \\ x_{+n}^{[l]} \end{pmatrix}$$

Affine Invariance

Affine Invariance

- The limit curve should be independent of the choice of a coordinate system
- We get this, if the intermediate subdivision points are affine invariant
- For this, the rows of the (local) subdivision matrix must sum to one:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Affine Invariance

Affine Invariance

- For this, the rows of the (local) subdivision matrix must sum to one:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

- This means: The one-vector **1** must be an eigenvector with eigenvalue 1:
  - $\boldsymbol{M}_{subdiv}\mathbf{1} = \mathbf{1}$
  - This must also be the largest eigenvalue / vector pair
  - One can show: it must be the only eigenvector with eigenvalue 1, otherwise the scheme does not converge

Summary

For a reasonable subdivision scheme, we need at least:

- **1** must be an eigenvector with eigenvalue 1.
- This must be the largest eigenvalue.
- The second eigenvalue should be smaller than 1
- All other eigenvalues should be smaller than the second one

(This is assuming a diagonizable subdivision matrix.)

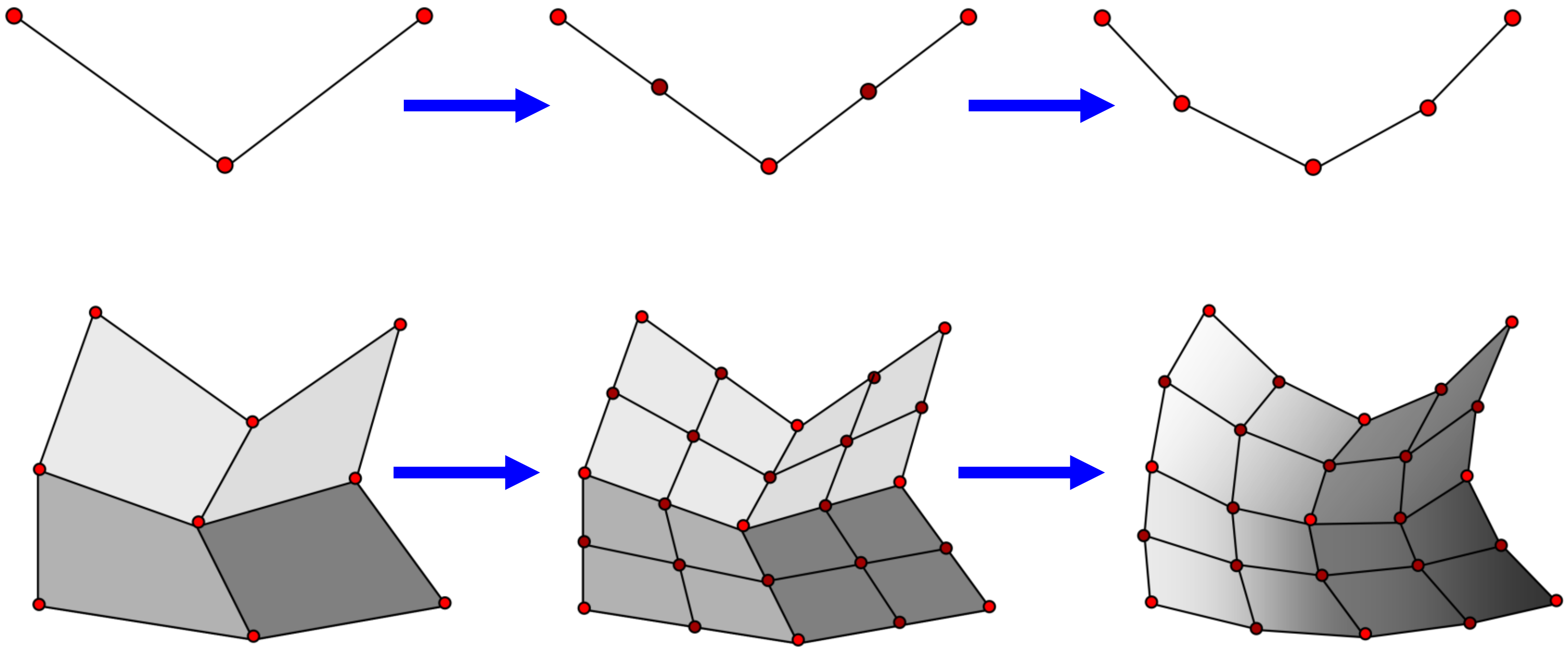
More details: Zorin, Schroder – Subdivision for Modeling and Animation, Siggraph 2000 course

B-Spline Subdivision Surfaces

B-Spline Subdivision Surfaces

B-Spline Subdivision Surfaces

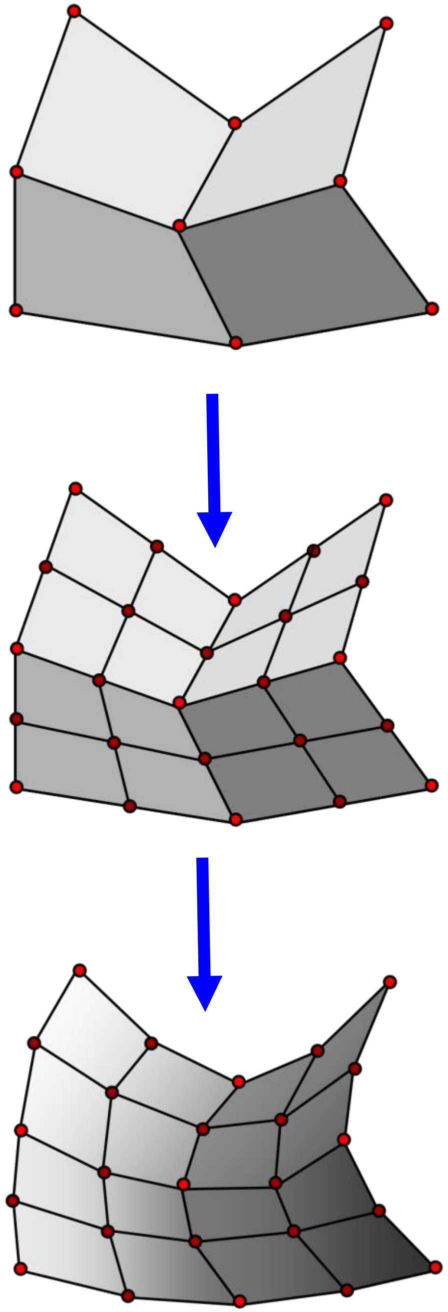
- We can apply the tensor product construction to obtain subdivision surfaces



B-Spline Subdivision Surfaces

Tensor Product B-Spline Subdivision Surfaces

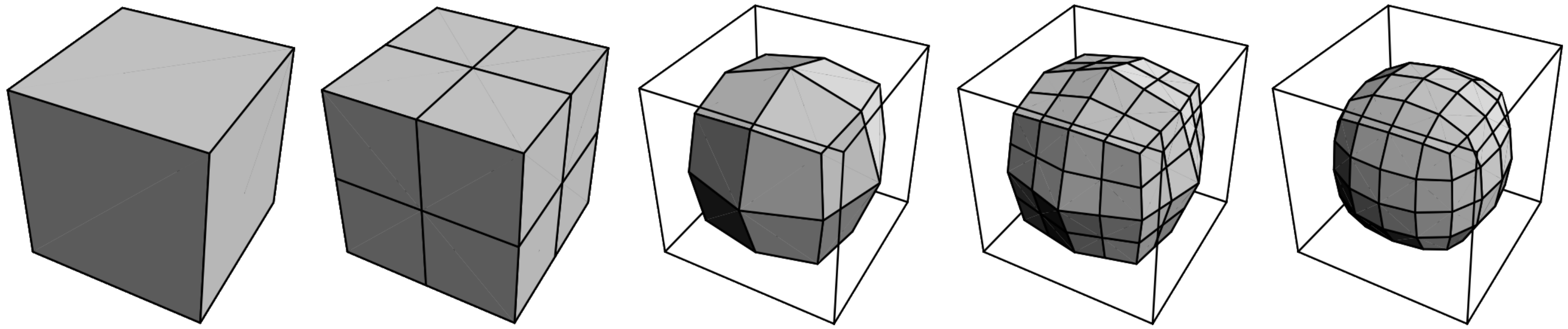
- Start with a regular quad mesh
  (will be relaxed later)
- In each subdivision step:
  - Divide each quad in four (quadtree subdivision)
  - Place linearly interpolated vertices
  - Apply 2-dimensional averaging mask

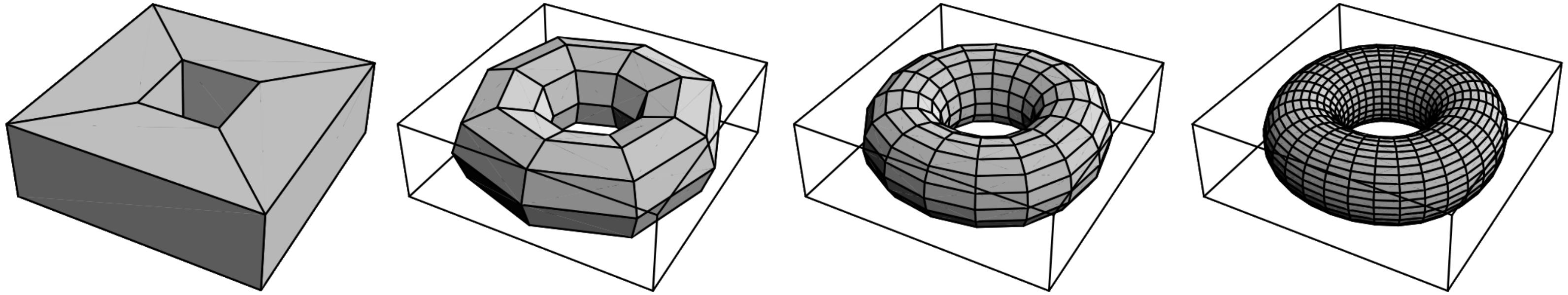


B-Spline Subdivision Surfaces

Bilinear Subdivision Surfaces + quad averaging:

- Quad averaging : reposition each vertex at the centroid of its adjacent quads





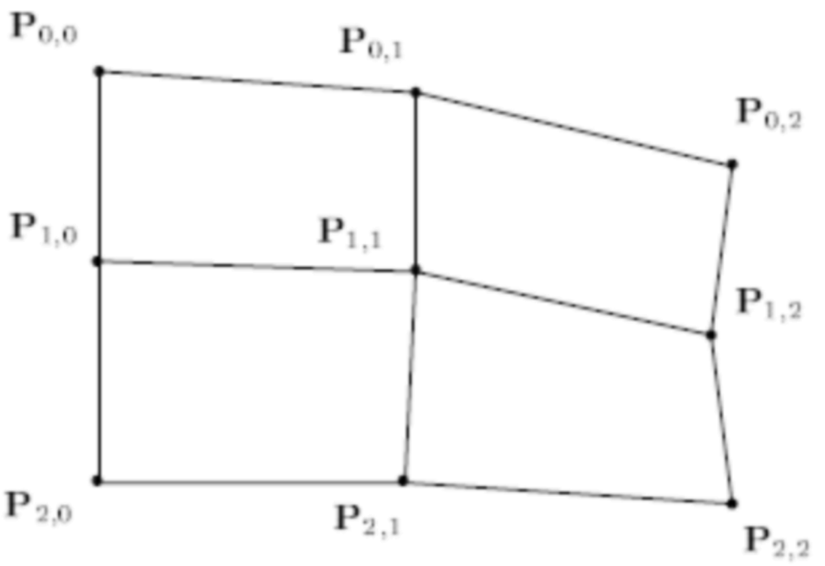
B-Spline Subdivision Surfaces

Biquadratic case:

- Recall the matrix B-spline patch representation

$$P(u,v) = \begin{bmatrix} 1 & u & u^2 \end{bmatrix} M P M^T \begin{bmatrix} 1 \\ v \\ v^2 \end{bmatrix}$$

$$M = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix}, \qquad P = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} \\ P_{1,0} & P_{1,1} & P_{1,2} \\ P_{2,0} & P_{2,1} & P_{2,2} \end{bmatrix}$$



B-Spline Subdivision Surfaces

Biquadratic case:

- By restricting to only one quadrant of the 2×2 patch, i.e. $u, v \in [0, \frac{1}{2}]$. We consider the new surface patch P' defined by re-parameterization $u' = \frac{u}{2}$, $v' = \frac{v}{2}$

$$P'(u,v) = P\left(\frac{u}{2}, \frac{v}{2}\right) = \begin{bmatrix} 1 & u/2 & u^2/4 \end{bmatrix} MPM^T \begin{bmatrix} 1 \\ v/2 \\ v^2/4 \end{bmatrix}$$

$$= \cdots = [1 \quad u \quad u^2] MP'M^T \begin{bmatrix} 1 \\ v \\ v^2 \end{bmatrix}$$

B-Spline Subdivision Surfaces

Biquadratic case:

- By restricting to only one quadrant of the 2×2 patch, i.e. $u, v \in [0, \frac{1}{2}]$. We consider the new surface patch P' defined by re-parameterization $u' = \frac{u}{2}$, $v' = \frac{v}{2}$

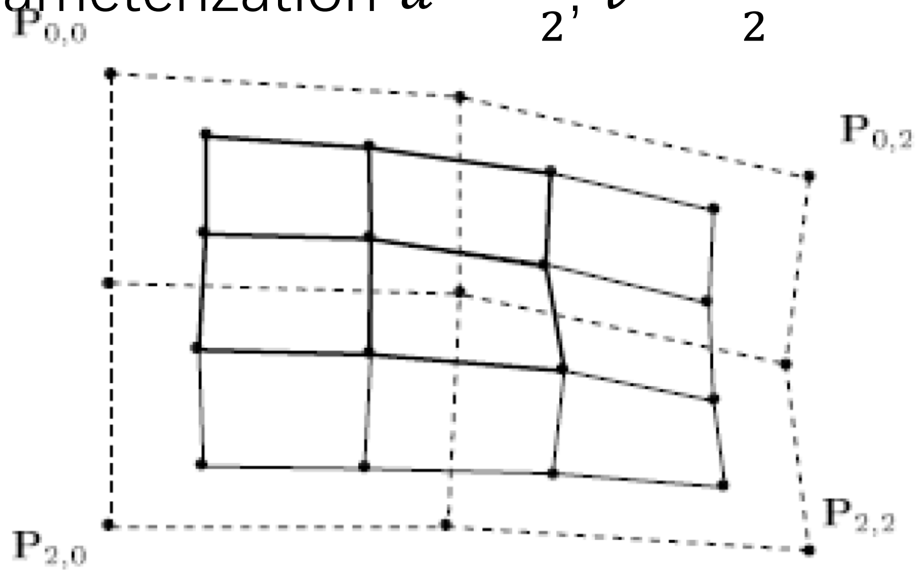
$$P' = SPS^T$$

$$S = M^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} M$$

B-Spline Subdivision Surfaces

Biquadratic case:

- By restricting to only one quadrant of the 2×2 patch, i.e. $u, v \in [0, \frac{1}{2}]$. We consider the new surface patch P' defined by re-parameterization $u' = \frac{u}{2}$, $v' = \frac{v}{2}$



$$P'_{00} = \frac{1}{16}(9P_{00} + 3P_{10} + 3P_{01} + P_{11})$$

$$P'_{01} = \frac{1}{16}(3P_{00} + P_{10} + 9P_{01} + 3P_{11})$$

$$P'_{02} = \frac{1}{16}(9P_{01} + 3P_{11} + 3P_{02} + 2P_{12})$$

$$P'_{11} = \frac{1}{16}(3P_{00} + 9P_{10} + P_{01} + 3P_{11})$$

$$P'_{11} = \frac{1}{16}(P_{00} + 3P_{10} + 3P_{01} + 9P_{11})$$

$$P'_{12} = \frac{1}{16}(3P_{01} + 9P_{11} + P_{02} + 3P_{12})$$

$$P'_{20} = \frac{1}{16}(9P_{10} + 3P_{20} + 3P_{11} + P_{21})$$

$$P'_{21} = \frac{1}{16}(3P_{10} + P_{20} + 9P_{11} + 3P_{21})$$

$$P'_{22} = \frac{1}{16}(9P_{11} + 3P_{21} + 3P_{12} + P_{22})$$

B-Spline Subdivision Surfaces

Bicubic case:

- Recall the matrix B-spline patch representation

$$P(u,v) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} MPM^T \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix}$$

$$M = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

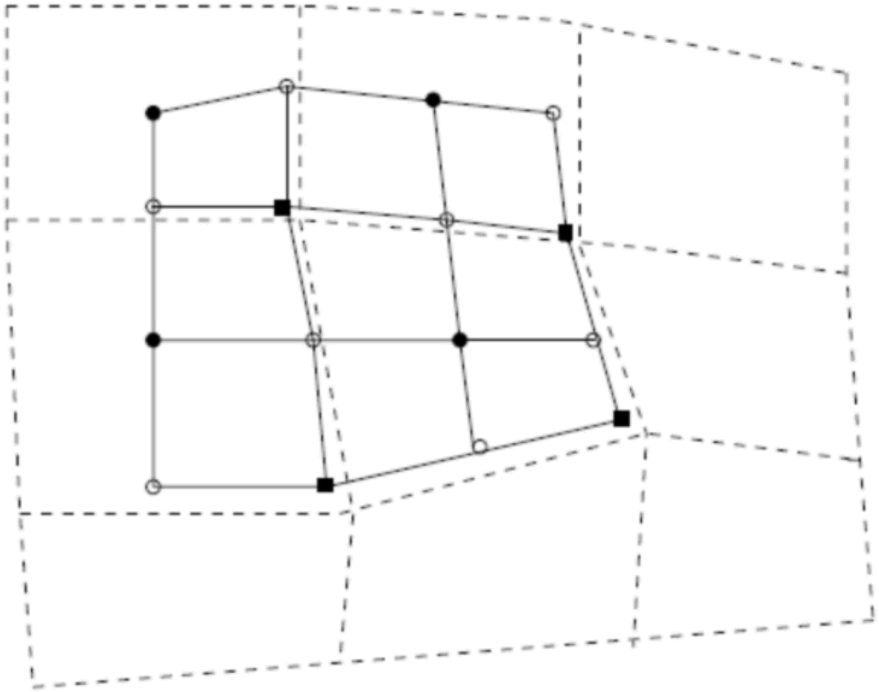
B-Spline Subdivision Surfaces

Bicubic case:

- By restricting to only one quadrant of the 3×3 patch, i.e. $u, v \in [0, \frac{1}{2}]$. We consider the new surface patch P' defined by re-parameterization $u' = \frac{u}{2}, v' = \frac{v}{2}$
- We obtain similarly (by matrix manipulation)

$$P' = SPS^T$$

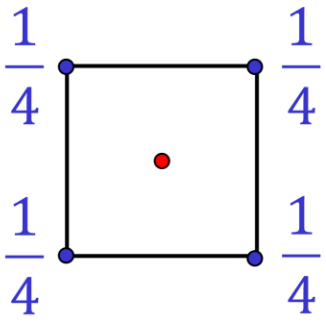
$$S = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \end{bmatrix}$$



Subdivision and Averaging Masks

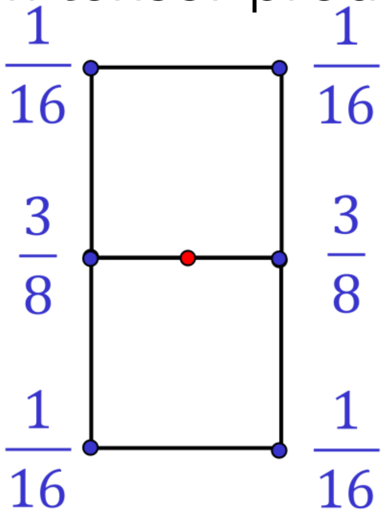
What is the subdivision mask?

- Can be derived from tensor product construction:



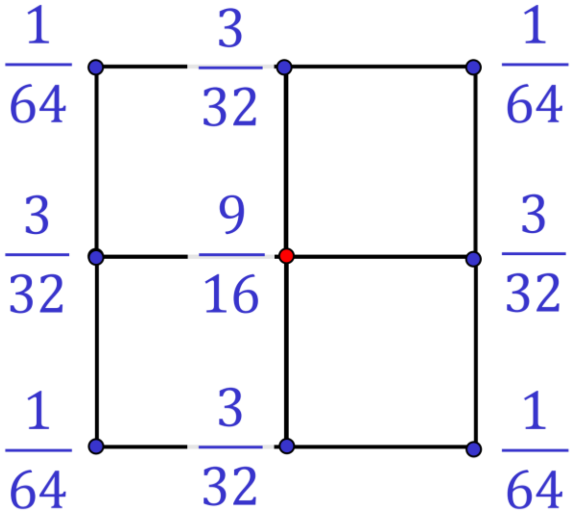
face midpoint
(odd/odd)

$$\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix}\cdot\left[\frac{1}{2},\frac{1}{2}\right]$$



edge midpoint
(even/odd)

$$\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix}\cdot\left[\frac{1}{8},\frac{3}{4},\frac{1}{8}\right]$$



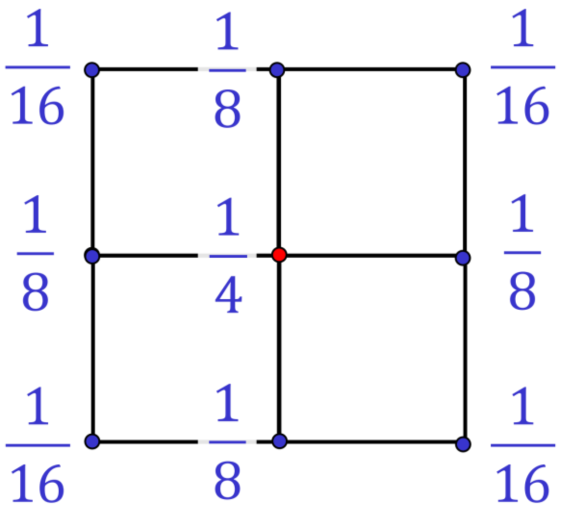
original vertex
(even/even)

$$\begin{pmatrix}\frac{1}{8}\\ \frac{3}{4}\\ \frac{1}{8}\end{pmatrix}\cdot\left[\frac{1}{8},\frac{3}{4},\frac{1}{8}\right]$$

Subdivision and Averaging Masks

What is the averaging mask?

- Can be derived from tensor product construction, too



Any (split) vertex

$$\begin{pmatrix}\frac{1}{4}\\ \frac{1}{2}\\ \frac{1}{4}\end{pmatrix}\cdot\left[\frac{1}{4},\frac{1}{2},\frac{1}{4}\right]$$

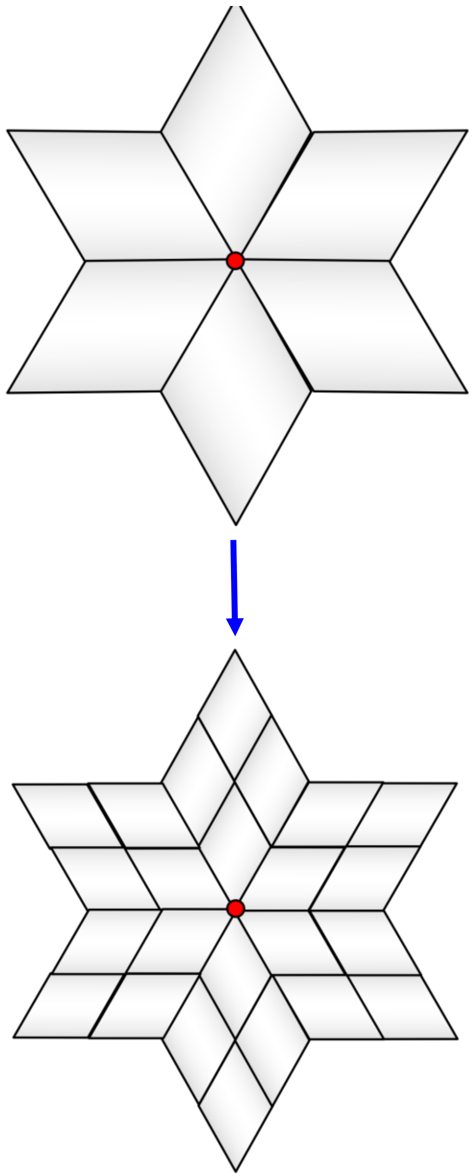
Remaining Problems

Remaining Problems:

- The derived rules work only in the interior or a regular quad mesh
- We did not really gain any flexibility over the standard B-spline construction
- We still need to figure out, how to …
  - …handle quad meshes of arbitrary topology
  - …handle boundary regions
    - Placing boundaries in the interior of objects will allow us to model sharp C^0 creases
    - So we also have some continuity control (despite the uniform B-Spline scheme)

Here is the answer…

Answer: Catmull-Clark subdivision scheme at extraordinary vertices



Observation:

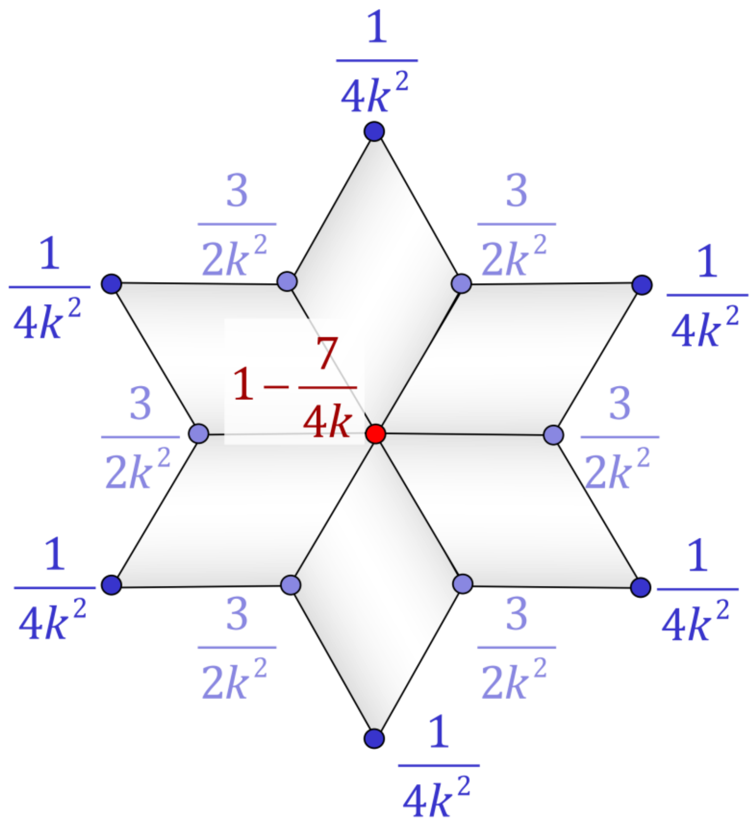
- The recursive subdivision rule always creates regular grids
- Problems can only occur at “extraordinary” vertices
  - These are vertices where the base has degree > 4
  - Extraordinary vertices are maintained by quadtree-like-subdivision
  - All new vertices are ordinary

Here is the answer…

Answer: Catmull-Clark subdivision scheme at extraordinary vertices

Subdivision mask at extraordinary vertex:

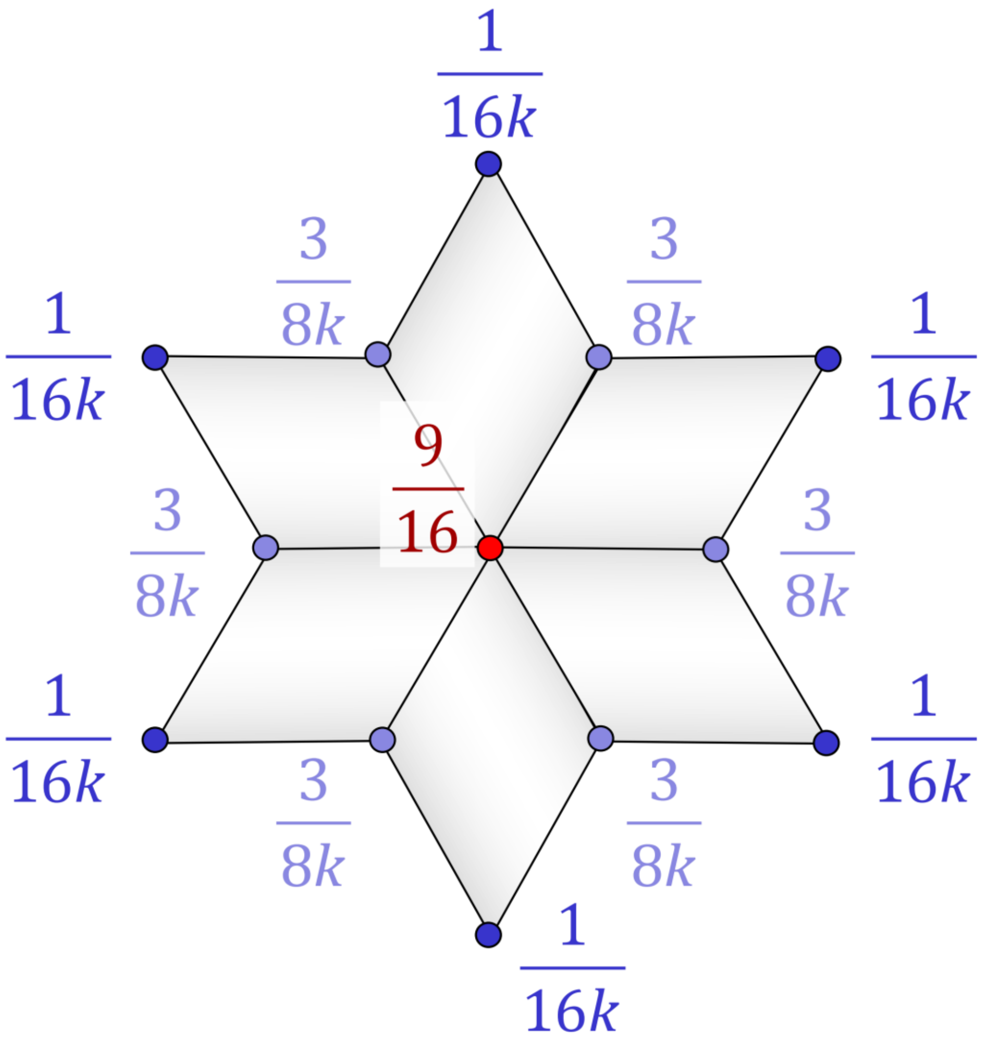
- Vertex degree k (number of incident faces)
- The surface is C^1 at extraordinary vertices



Here is the answer…

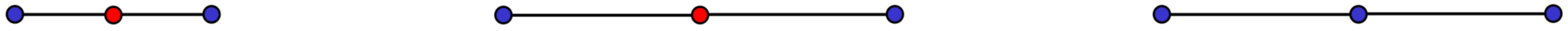
Averaging mask:

- Use after bilinear splitting



Boundary Rules

Subdivision mask at boundaries / sharp creases:



$\frac{1}{2}$ $\frac{1}{2}$

(odd)

$\frac{1}{8}$ $\frac{3}{4}$ $\frac{1}{8}$

(even)

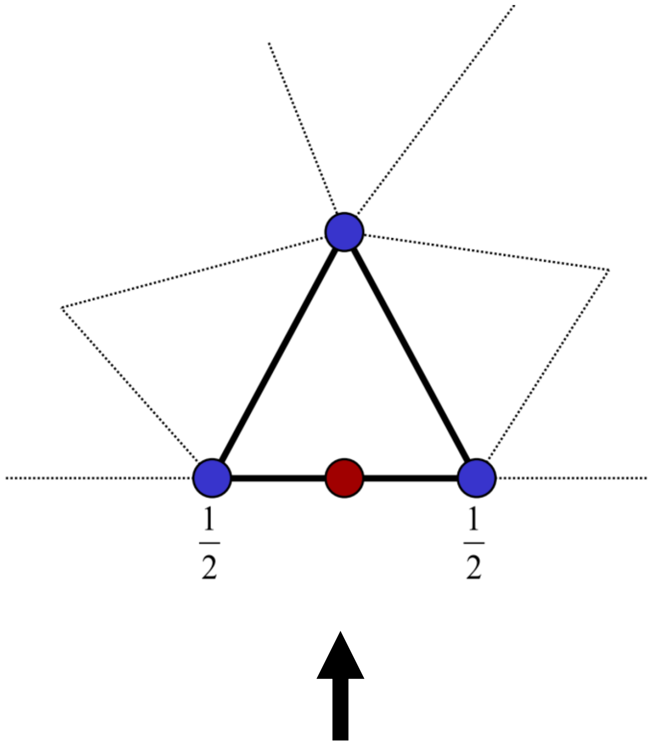
$\frac{1}{4}$ $\frac{1}{2}$ $\frac{1}{4}$

(averaging mask)

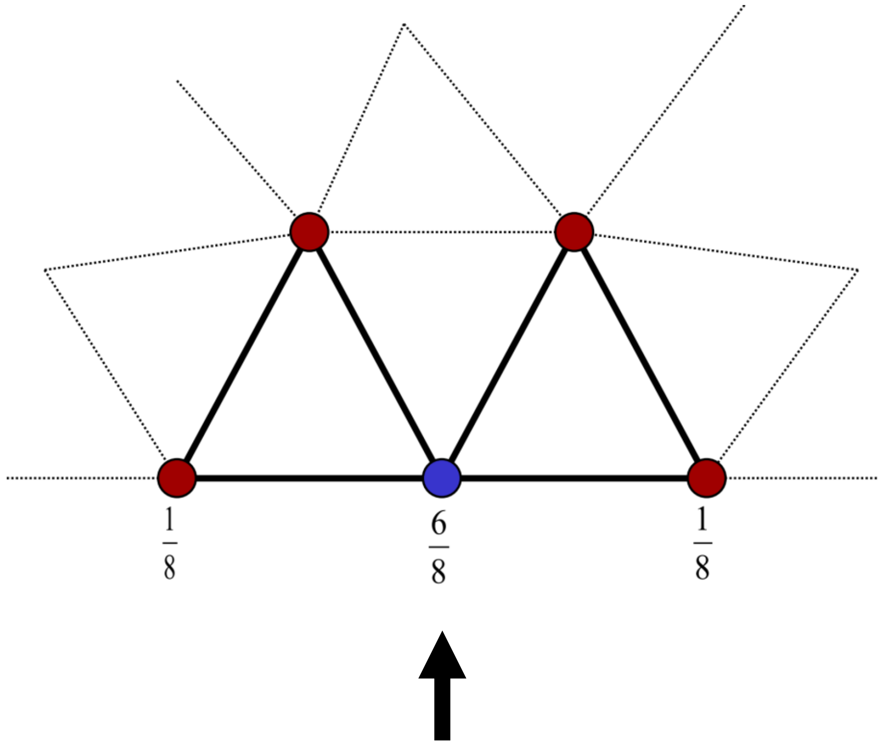
- Just use the normal spline curve rules
- This gives visually good results
- However, the surface is not strictly C^1 at the boundary
- There is a modified weighting scheme that creates half-sided C^1-continuous surfaces at the boundary curves

Boundary Rules

Subdivision Mask for Boundary Conditions



Edge Rule (odd)



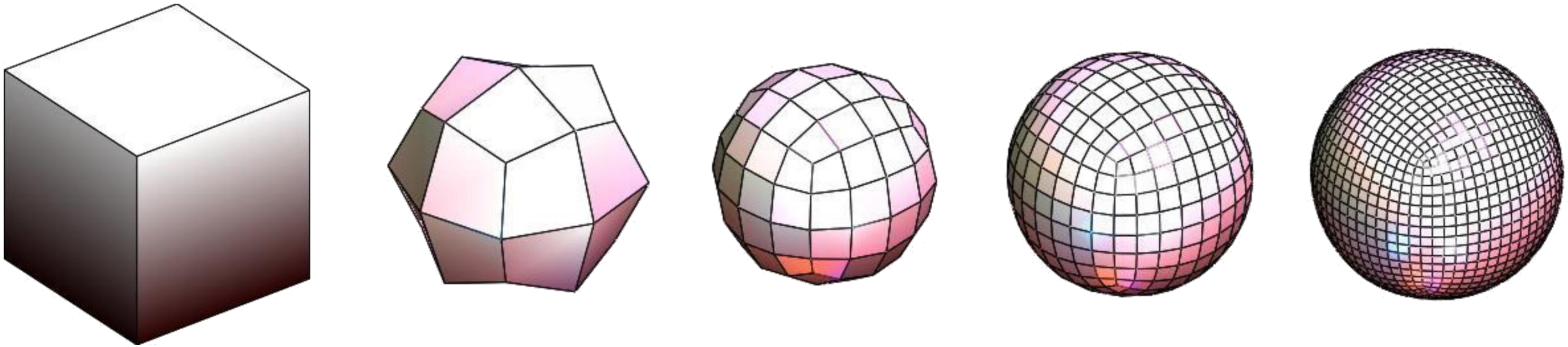
Vertex Rule (even)

Catmull-Clark in short

Face, edge, vertex points:

1. Introduce a face point for each face of the original mesh. The point is simply the average of all the points that bound the face.
2. An edge point is created for each interior edge of the polygonal surface. The point is the average of the midpoint of the edge and the two face points on both sides of the edge
3. A vertex point is generated for each interior vertex P of the original mesh. The point is the average of Q, $2R$, and $\frac{(n-3)S}{n}$, where Q is the average of the face points on all the faces adjacent to P, R is the average of the midpoints of all the edges incident on P, and S is simply P itself

Catmull-Clark scheme



Other Subdivision Schemes

Loop, Butterfly, …

Subdivision Zoo

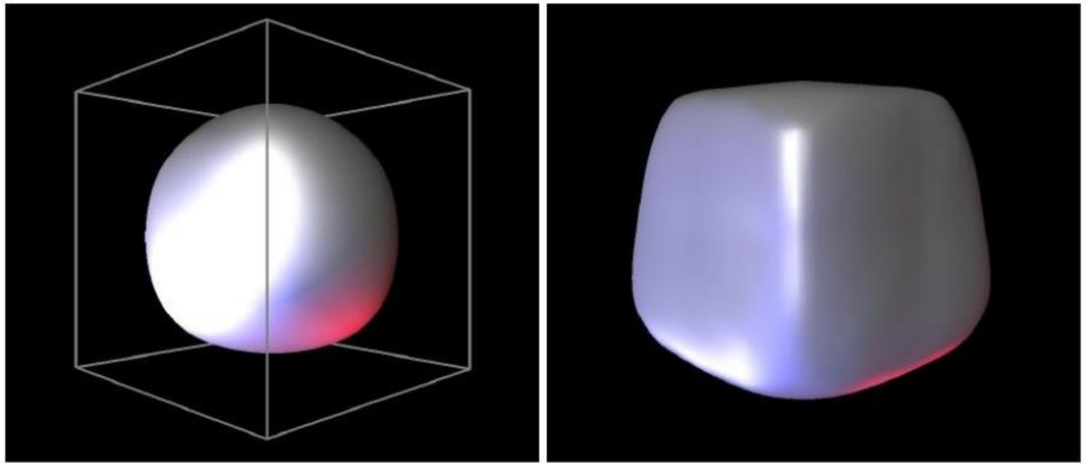
A large number of subdivision scheme exists. The most popular are:

- Catmull-Clark subdivision
  (*quad-mesh*, *approximating*, C^2 surfaces, C^1 at extraordinary vertices)
- Loop subdivision
  (*triangular, approximating*, C^2 *surfaces*, C^1 *at extraordinary vertices*)
- Butterfly subdivision
  (*triangular*, *interpolation*, C^1 *surfaces*, C^1 *at extraordinary vertices*)

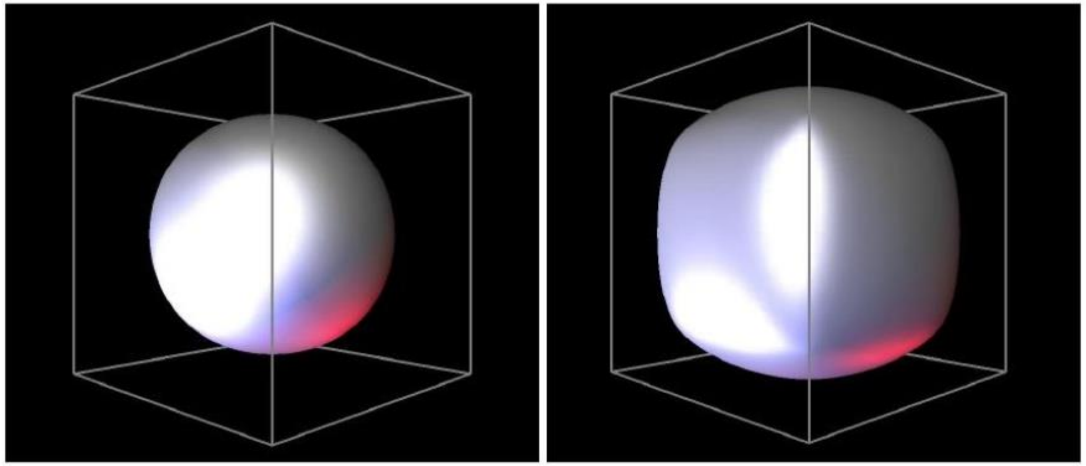
Examples of other schemes:

- $\sqrt{3}$-subdivision (level of detail increases more slowly)
- Circular subdivision (used e.g. for surfaces of revolution)

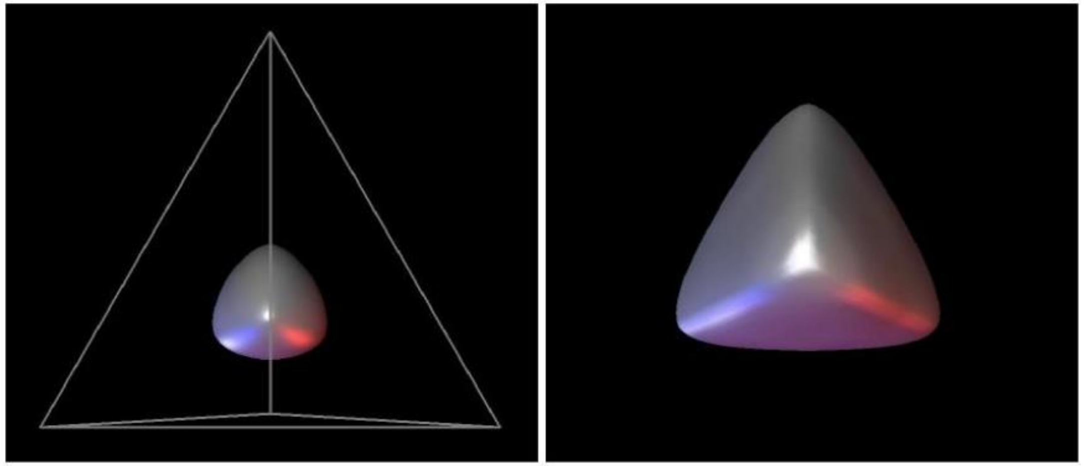
Comparisons



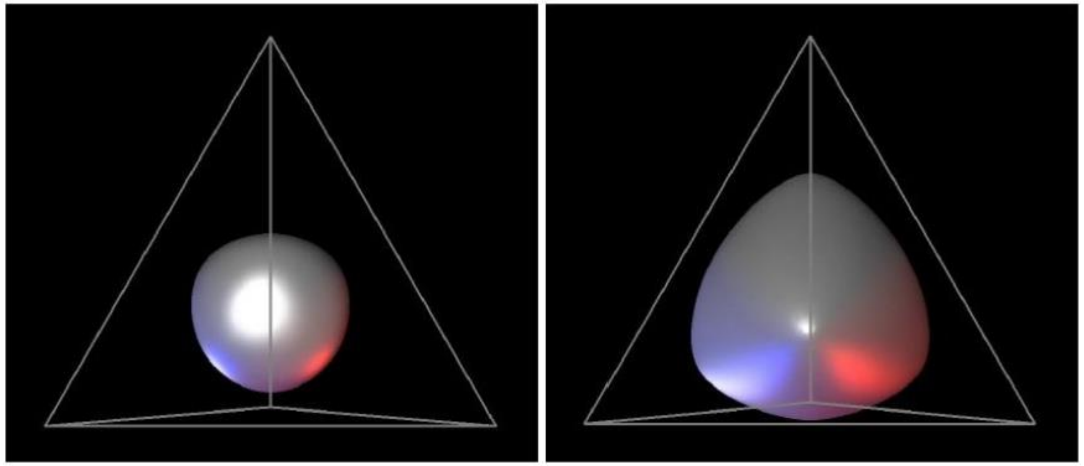
Loop *Butterfly*



Catmull-Clark *Doo-Sabin*



Loop *Butterfly*

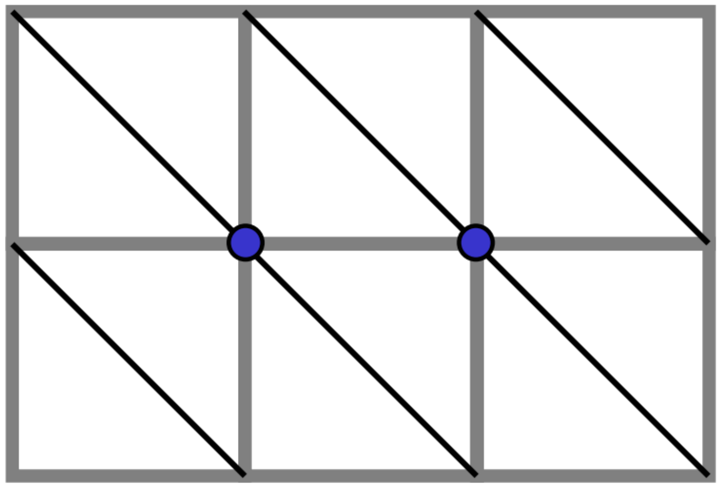


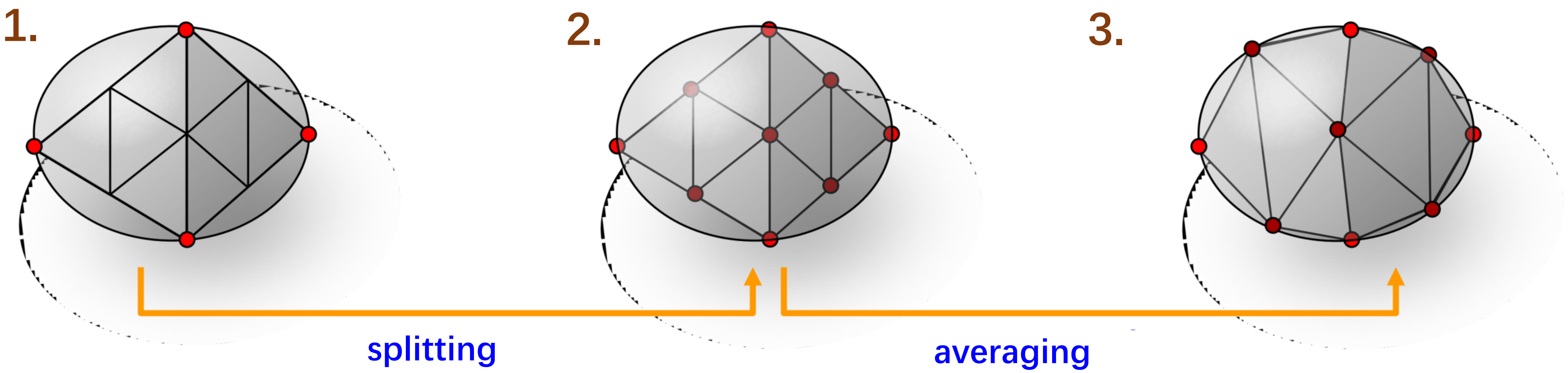
Catmull-Clark *Doo-Sabin*

Triangular Subdivision

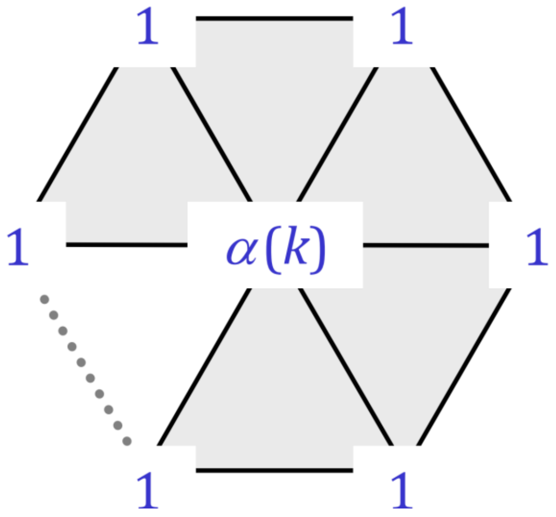
Triangular Subdivision:

- Uses 1:4 triangular splits
  - Extraordinary vertices: valence $\neq 6$
- Again:
  - Splitting with linear interpolation
  - Then apply averaging mask

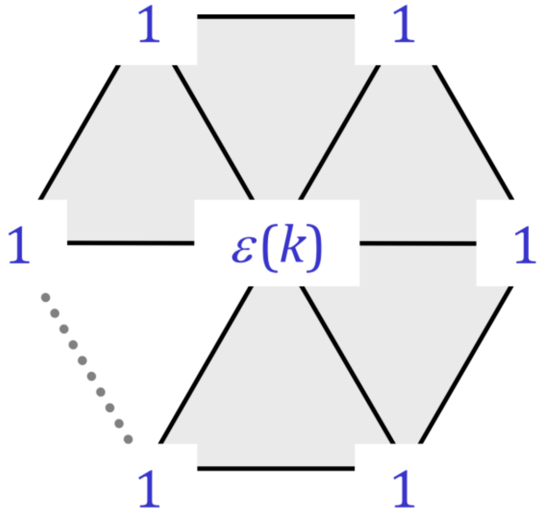




Loop Subdivision



averaging mask



evaluation (limit) mask

$$\frac{1}{4} \;—\; \frac{1}{2} \;—\; \frac{1}{4}$$

boundary/sharp crease mask

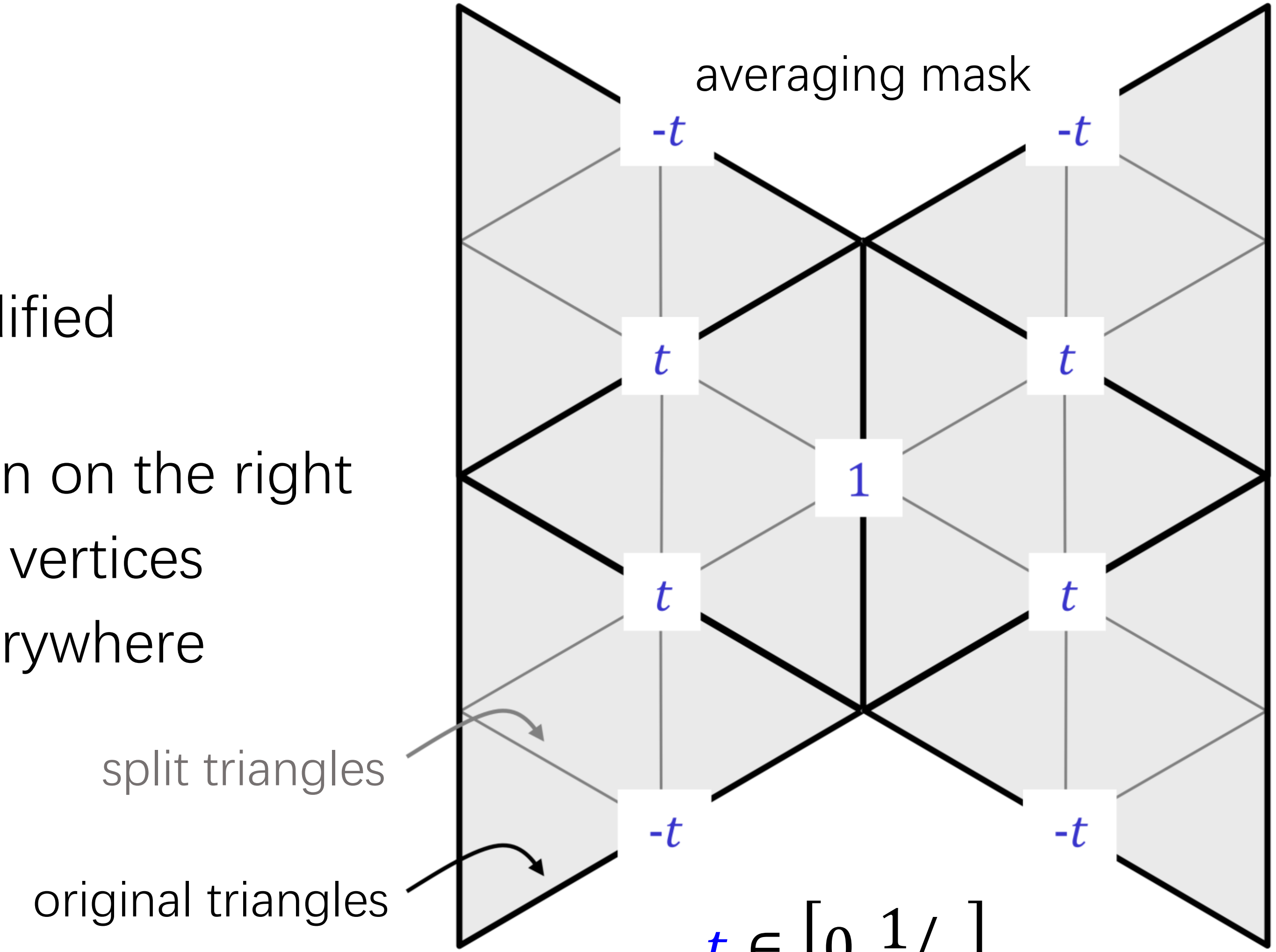
$$\alpha(k) = {k\left(1-\beta(k)\right)}\Big/{\beta(k)} \qquad \varepsilon(k) = {3k}\Big/{\left(4\beta(k)\right)}$$

$$\beta(k) = \frac{5}{4} - \frac{(3+2\cos(2\pi/k))^2}{32}$$

Butterfly Scheme

Butterfly scheme:

- Original points remain unmodified (interpolating scheme)
- New points averaged as shown on the right
- C^1, except from extraordinary vertices
- Can be modified to be C^1 everywhere



$t \in [0, {}^1/_8]$

$0 \mathrel{\widehat{=}}$ *polyhedral*, ${}^1/_8 \mathrel{\widehat{=}}$ *smooth*