Geometric Modeling

Summer Semester 2010

Subdivision Surfaces

B-Spline Subdivision · Spectral Analysis · Example Schemes

UNIVERSITÄT DES SAARLANDES

CLUSTER OF EXCELLENCE

max planck institut informatik

Overview...

Topics:

- Rational Spline Curves
- Spline Surfaces
- Triangle Meshes & Multi-Resolution Representations
- Subdivision Surfaces
  - Introduction
  - B-Spline Subdivision Curves
  - Spectral Analysis
  - B-Spline Subdivision Surfaces
  - Other Subdivision Schemes
  - Connection to Wavelets
  - Stochastic Subdivision

Subdivision Surfaces

Introduction

Subdivision Surfaces

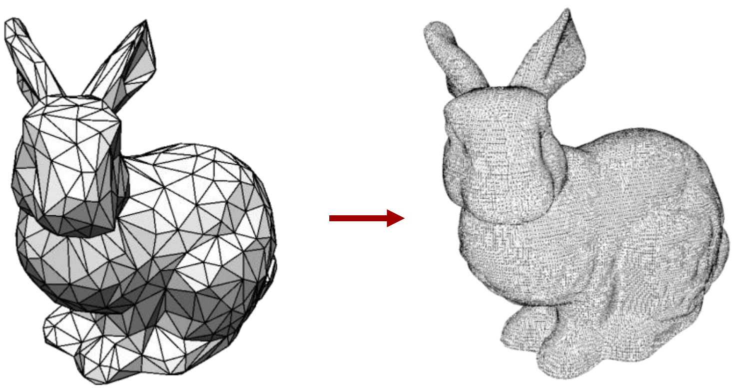
Problems with Spline Patches:

- 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

Simple Idea:



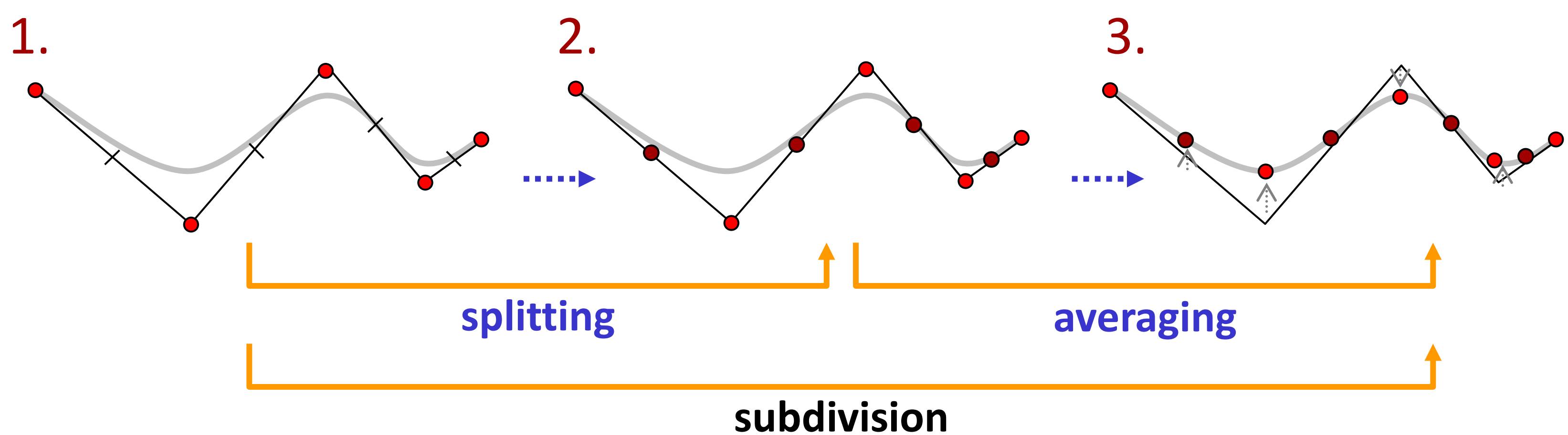
- Use a triangle mesh / quad mesh itself as a parametrization domain ("base mesh")
- Use 1:4 splits to refine the base mesh (subdivision connectivity meshes)
- Find an interpolation rule to create a smooth surface

This basic idea leads to subdivision surfaces.

Basic Scheme

Subdivision Curves & Surface: Three Steps

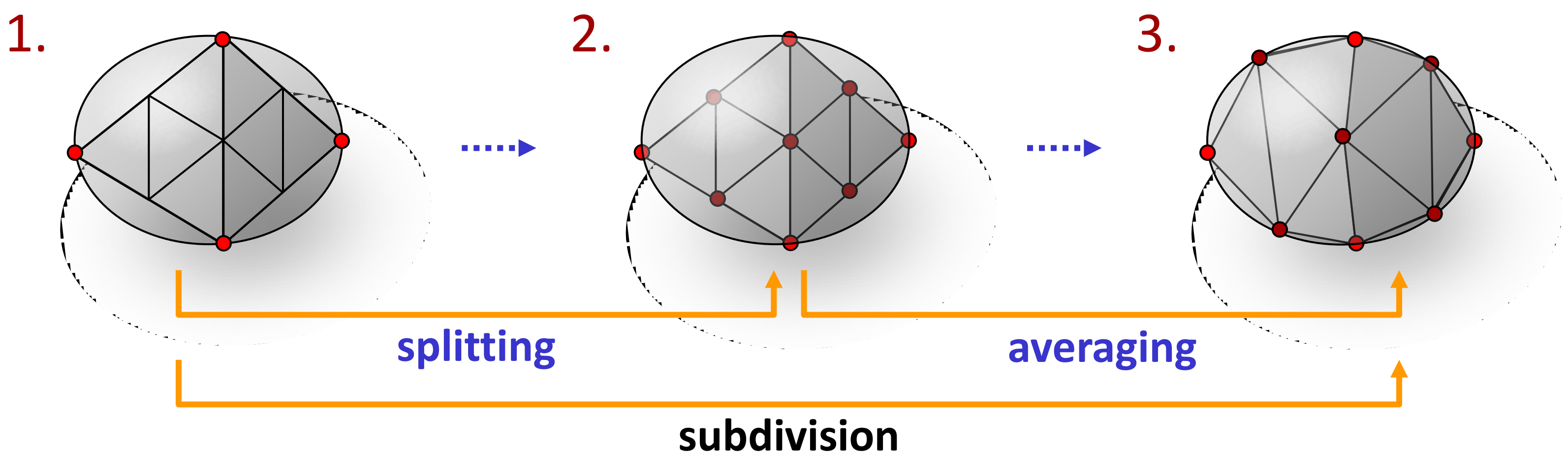
1. Subdivide current mesh
2. Insert linearly interpolated points (*splitting*)
3. Move points: Local weighted average (*averaging*)
   - To all points – approximating scheme
   - To new points only – interpolating scheme



Basic Scheme

Subdivision Curves & Surface: Three Steps

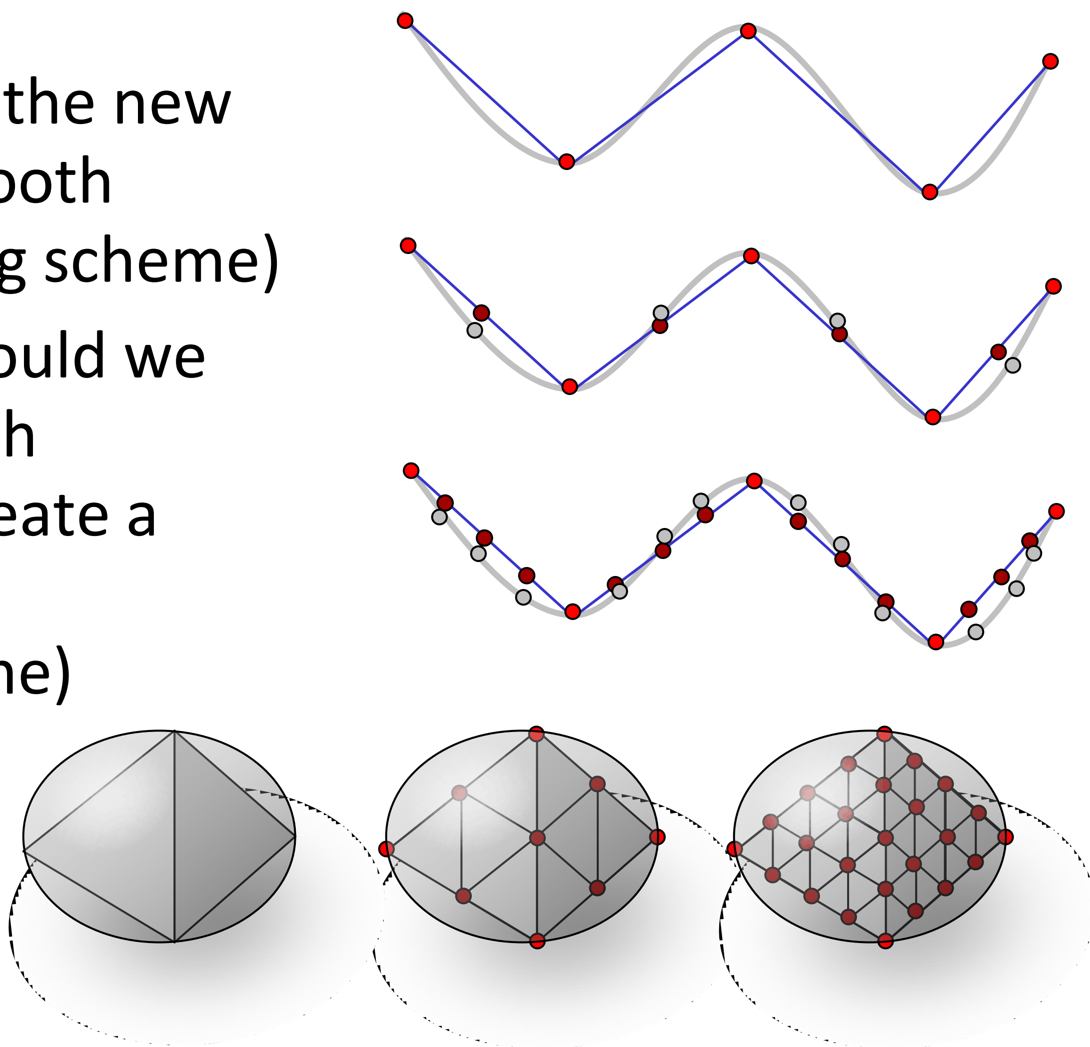
1. Subdivide current mesh
2. Insert linearly interpolated points (*splitting*)
3. Move points: Local weighted average (*averaging*)
   - To all points – approximating scheme
   - To new points only – interpolating scheme



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

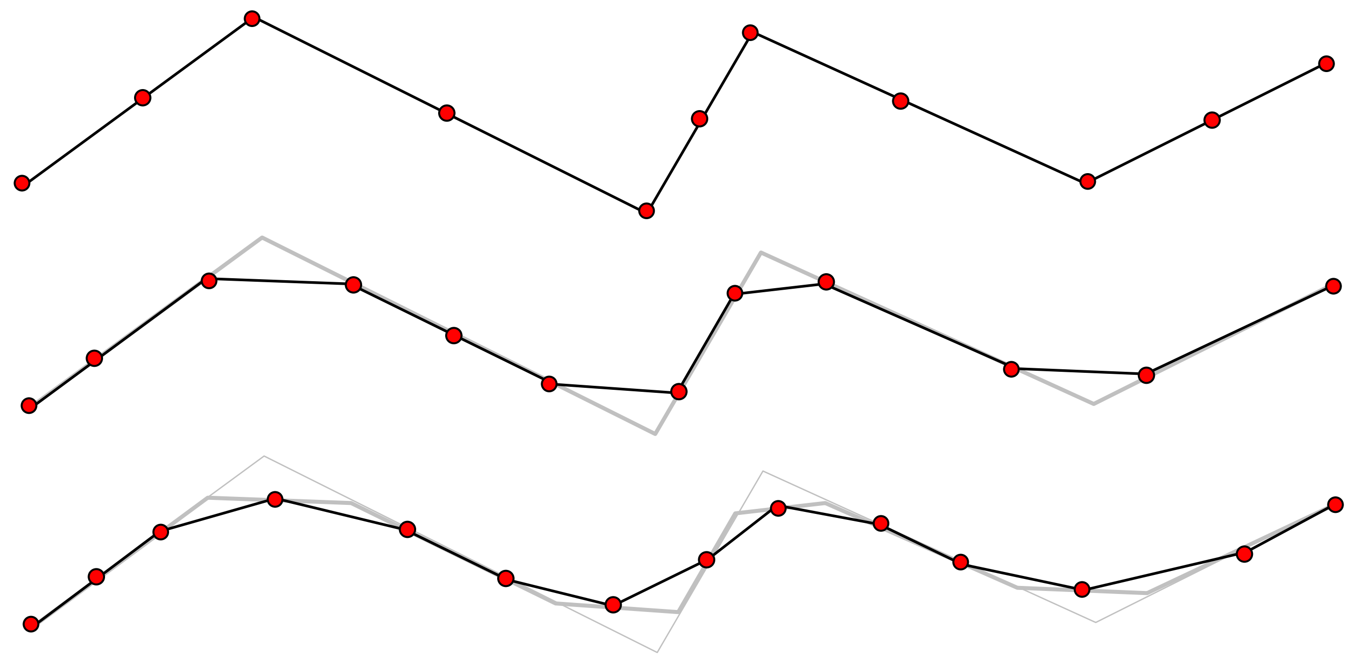
More precisely:

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

Interesting observation:

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

Example: Corner Cutting



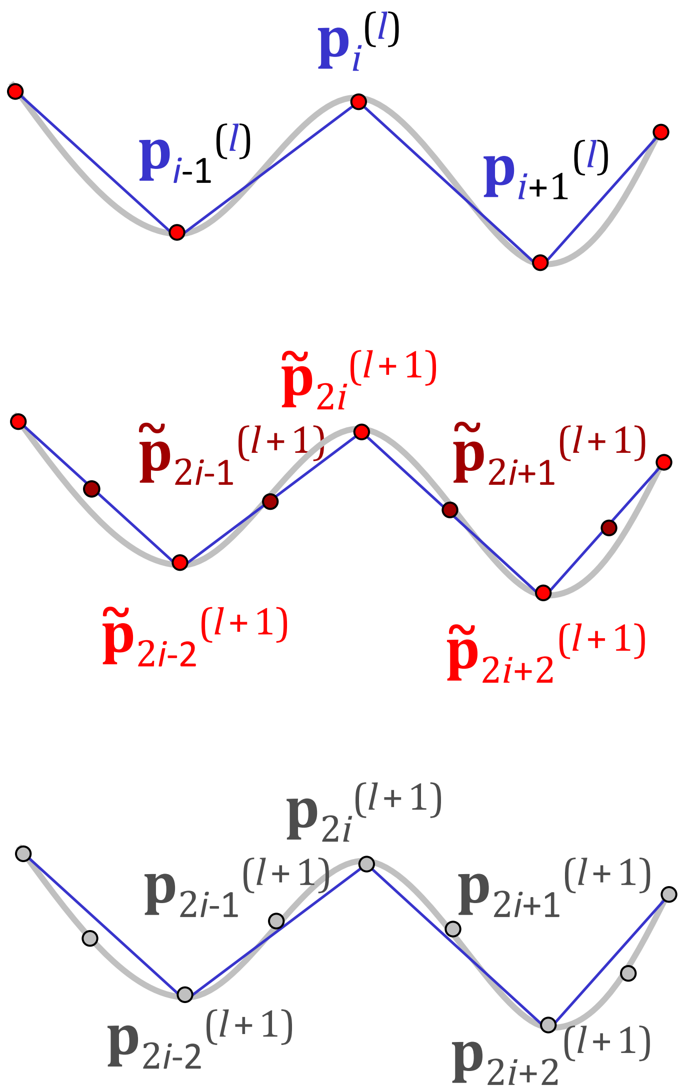
Corner Cutting Splines [Chaikin 1974]:

- Simple idea: Replace every point by the average of two neighbors (after splitting)
- Converges to quadratic B-Spline curve

Matrix Notation

Curve subdivision in matrix notation:

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



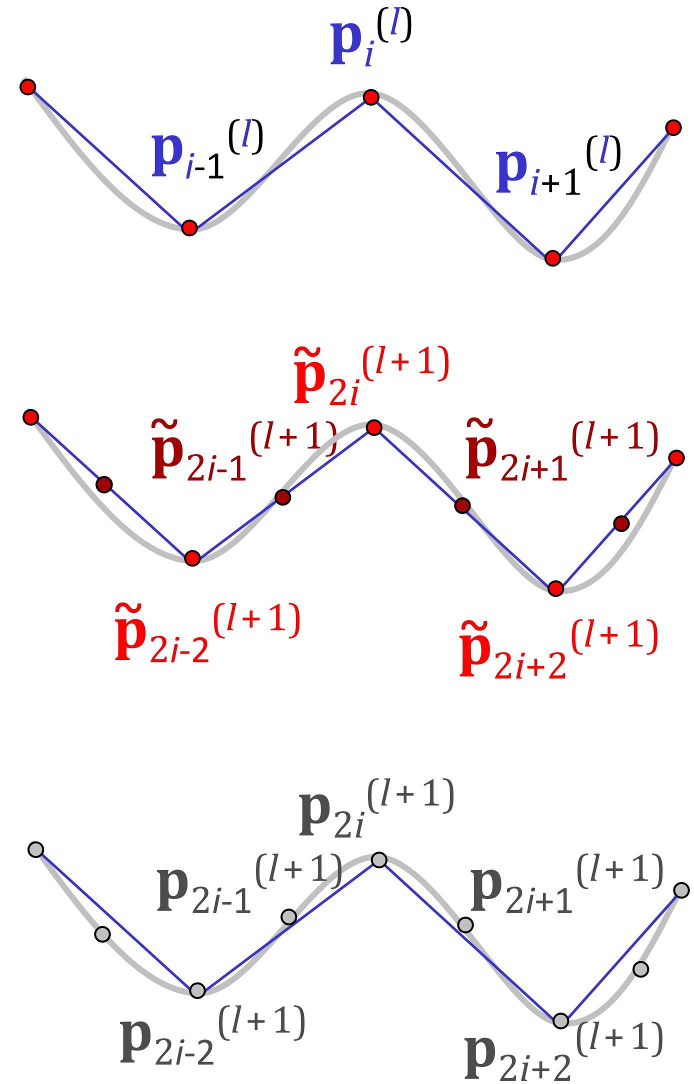
Matrix Notation

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:

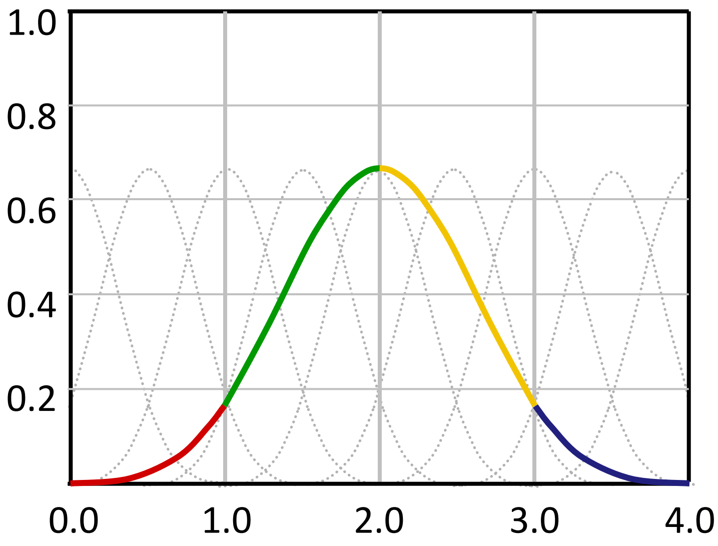
$$2n\left\{\begin{pmatrix}\vdots\\ x_{2i}^{(l+1)}\\ x_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right. = 2n\left\{\underbrace{\begin{pmatrix}\ddots & & & \\ & 1 & 1 & \\ & & 1 & 1 \\ & & & \ddots\end{pmatrix}}_{2n}\right. \left.\begin{pmatrix}\vdots\\ \tilde{x}_{2i}^{(l+1)}\\ \tilde{x}_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right\}2n$$



B-Spline Subdivision Schemes

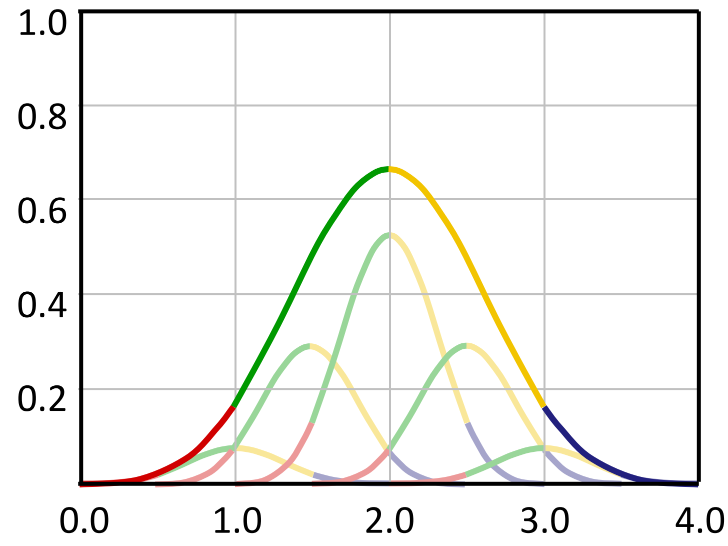
(Regular Subdivision)

B-Spline Subdivision



uniform cubic B-spline basis function $b(t)$

gray: *dilated* functions $b(2t+i)$



expressed as sum of dilated functions $b(t) = \Sigma_i \, c_i \, b(2t + i)$

B-Spline Subdivision

Subdivision Formula:

- Uniform B-spline functions $b(t+i)$ span the space $V^{(0)}$ of piecewise polynomials of degree d, piecewise in intervals $t \in [i, i+1)$
- *Dilated* uniform B-spline functions $b(2t+i)$ span the space $V^{(1)}$ of piecewise polynomials of degree d, piecewise in intervals $t \in [i/2, i/2+0.5)$
- Obviously $V^{(0)} \subset V^{(1)}$
- This means: We can express the larger functions as linear combinations of smaller ones.

Subdivision masks

Question: What are the coefficients for the linear combination?

Standard answer:

- Solve a linear system (underdetermined)

Direct derivation (arbitrary degree):

- We look at the definition of higher order basis functions by repeated convolution.
- By analyzing the effect of each further convolution, we get a subdivision formula by induction.

Repeated Convolution

Defining B-splines by repeated convolution (recap):

- We start with 0th degree basis functions
- Increase smoothness by convolution

Degree-zero B-spline:

$$b^{(0)}(t) = \begin{cases} 1, \text{if } t \in [0 \ldots 1) \\ 0, \text{otherwise} \end{cases}$$

General-degree B-spline:

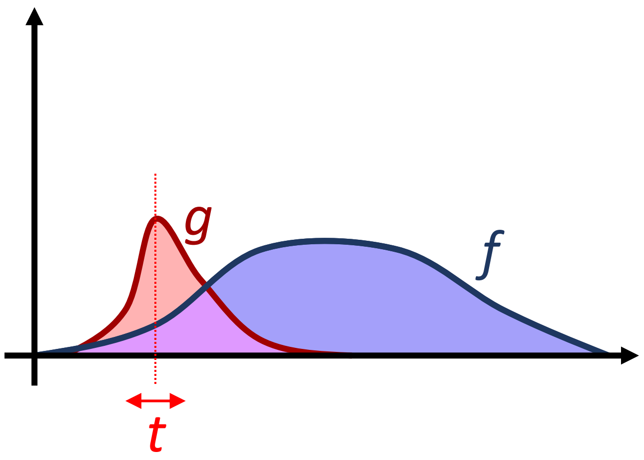
$$b^{(i+1)}(t) = b^{(i)}(t) \otimes b^{(0)}(t) = \int_{-\infty}^{\infty} b^{(i)}(x) b^{(0)}(x-t)\,dx$$

Repeated Convolution

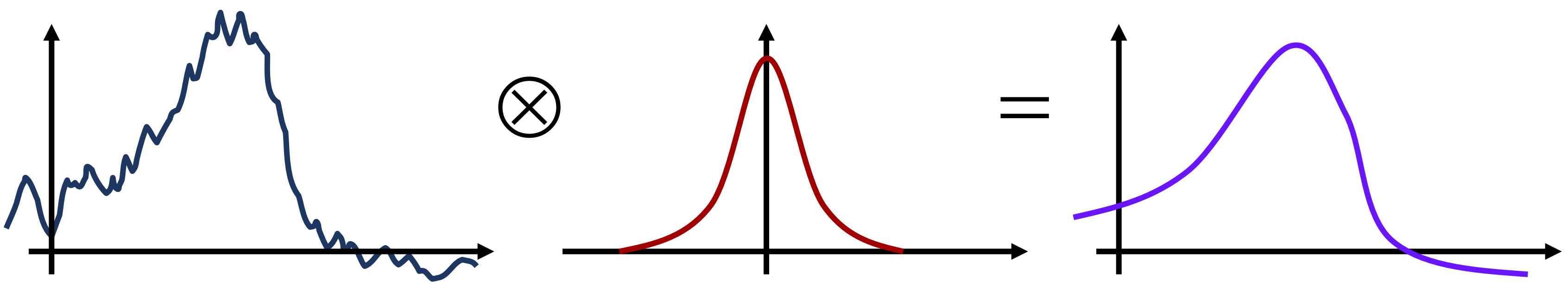
Convolution:

- Weighted average of functions
- Definition:

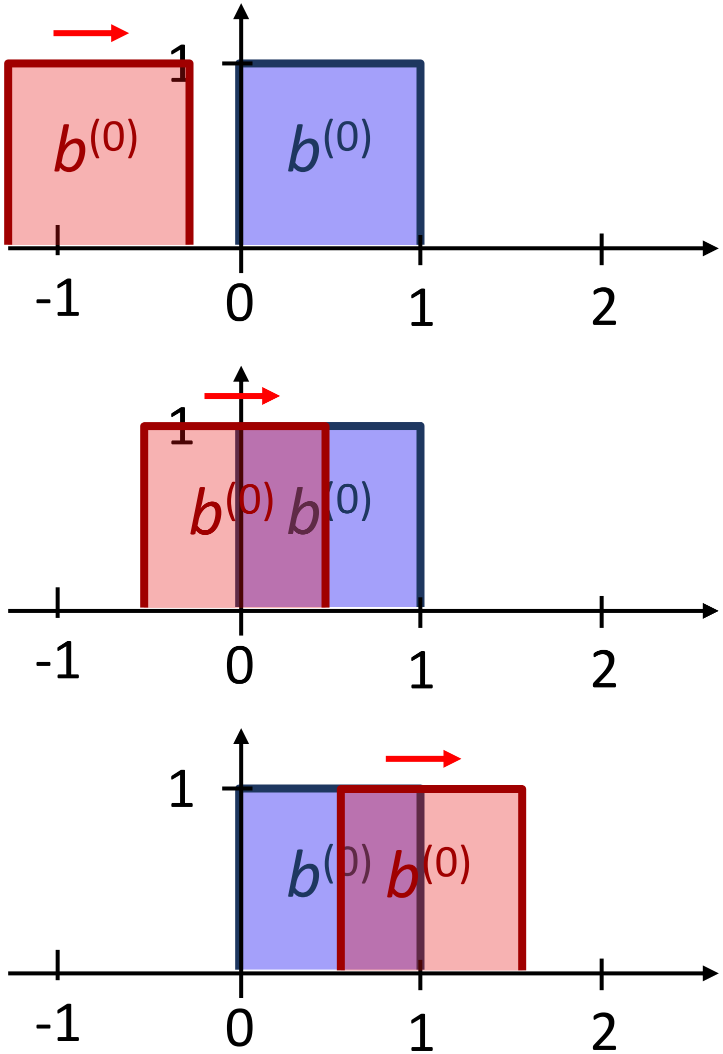
$$f(t)\otimes g(t)=\int_{-\infty}^{\infty} f(x)g(x-t)dx$$

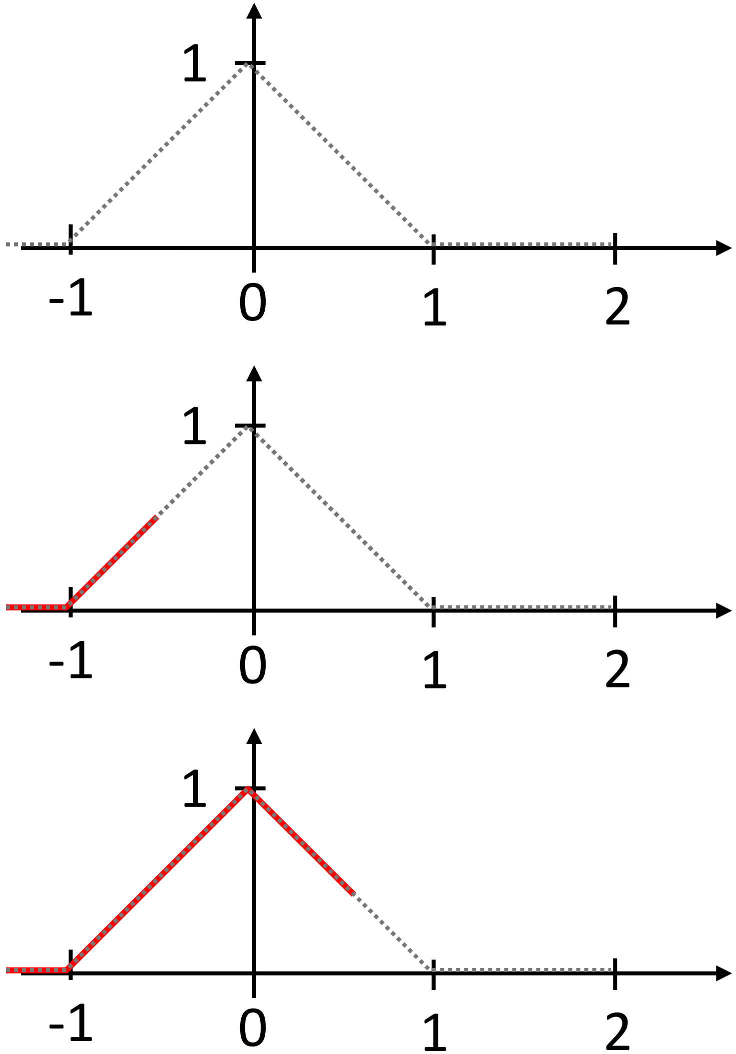


Example:

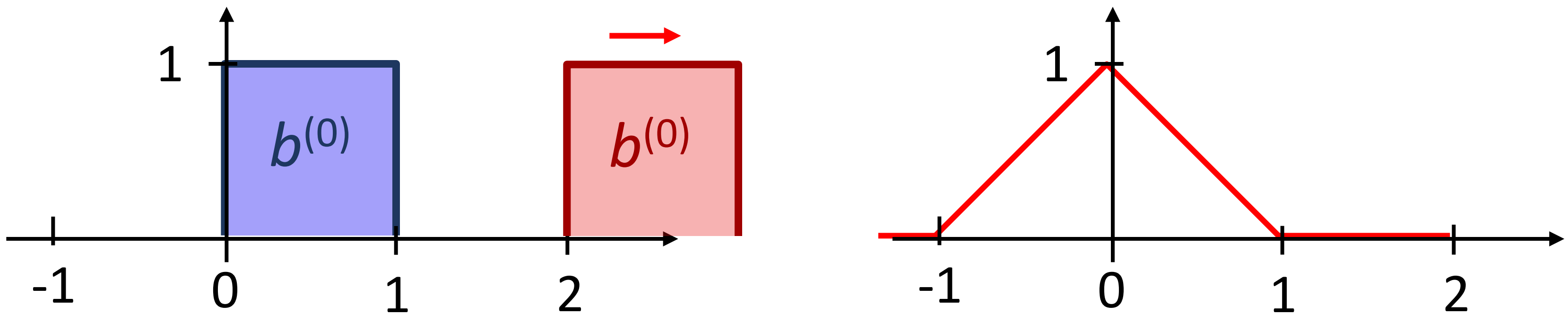


Illustration





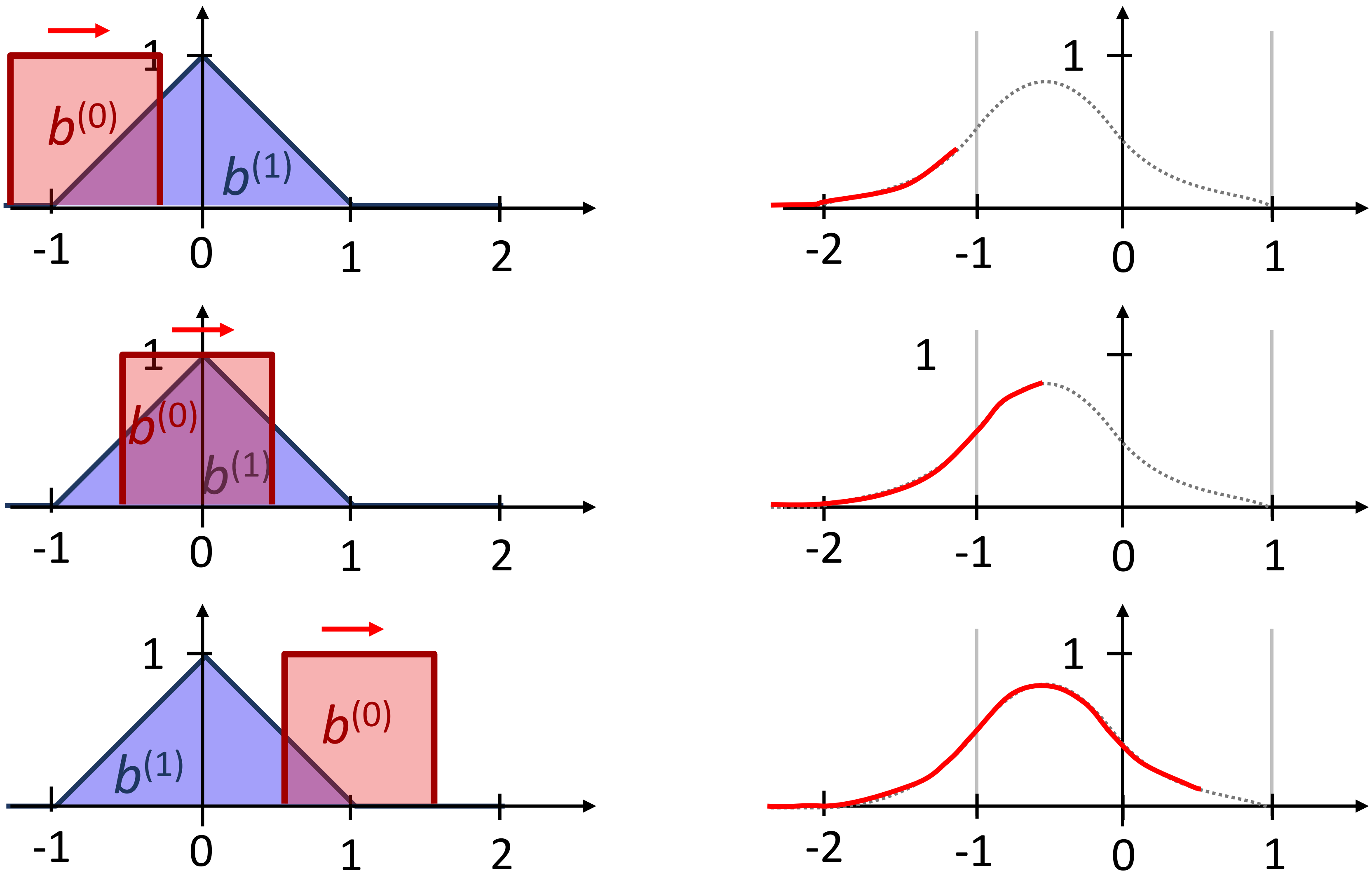
Illustration



Result:

- Piecewise linear B-spline basis function
- Each convolution with b_0 increases the continuity by 1.

Illustration



Properties of Convolution

Linearity:

$$f(t)\otimes\big(g(t)+h(t)\big)=f(t)\otimes g(t)+f(t)\otimes h(t)$$

Time Shift:

$$f(t+x)\otimes g(t+y)=\big(f\otimes g\big)(t+x+y)$$

Time Scaling:

$$f(2t)\otimes g(2t)=\frac{1}{2}\big(f\otimes g\big)(2t)$$

Subdivision Formula

Subdivision formula for linear B-splines:

$$
\begin{aligned}
b^{(1)}(t) &= b^{(0)}(t) \otimes b^{(0)}(t) \\
&= \left[b^{(0)}(2t) + b^{(0)}(2t-1)\right] \otimes \left[b^{(0)}(2t) + b^{(0)}(2t-1)\right] \\
&= b^{(0)}(2t) \otimes b^{(0)}(2t) + b^{(0)}(2t-1) \otimes b^{(0)}(2t) \\
&\quad + b^{(0)}(2t) \otimes b^{(0)}(2t-1) + b^{(0)}(2t-1) \otimes b^{(0)}(2t-1) \\
&= \frac{1}{2} b^{(1)}(2t) + b^{(1)}(2t-1) + \frac{1}{2} b^{(1)}(2t-2) \\
&= \frac{1}{2}\left(\sum_{i=0}^{2} \binom{2}{i} b^{(1)}(2t-i) \right)
\end{aligned}
$$

Linearity:

$$f(t) \otimes \left(g(t) + h(t)\right) = f(t) \otimes g(t) + f(t) \otimes h(t)$$

Time Shift:

$$f(t+x) \otimes g(t+y) = (f \otimes g)(t+x+y)$$

Time Scaling:

$$f(2t) \otimes g(2t) = \frac{1}{2}(f \otimes g)(2t)$$

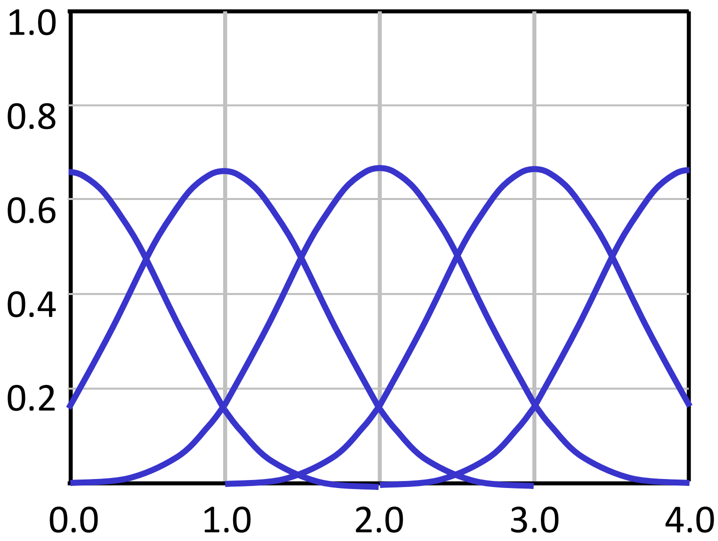
Subdivision Formula

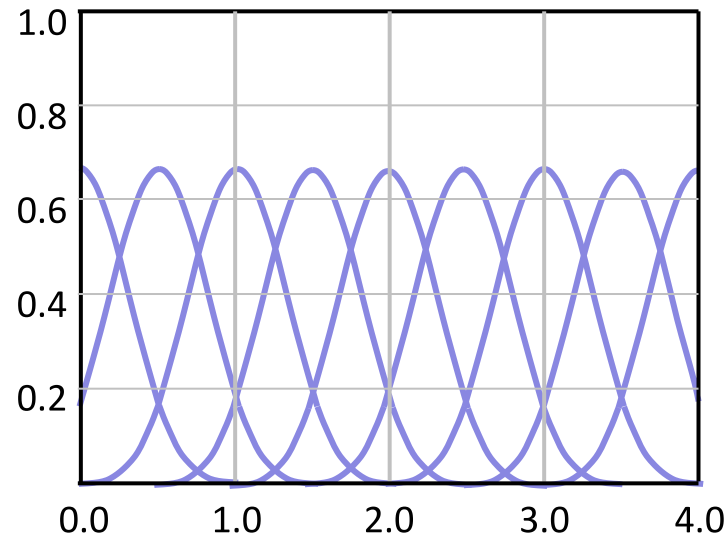
Analogously (induction) for general degrees:

$$b^{(d)}(t) = \underbrace{b^{(0)}(t) \otimes b^{(0)}(t) \otimes \ldots \otimes b^{(0)}(t)}_{d+1 \text{ times}}$$

$$= \frac{1}{2^d}\left(\sum_{i=0}^{d+1}\binom{d+1}{i} b^{(d)}(2t-i)\right)$$

This means....





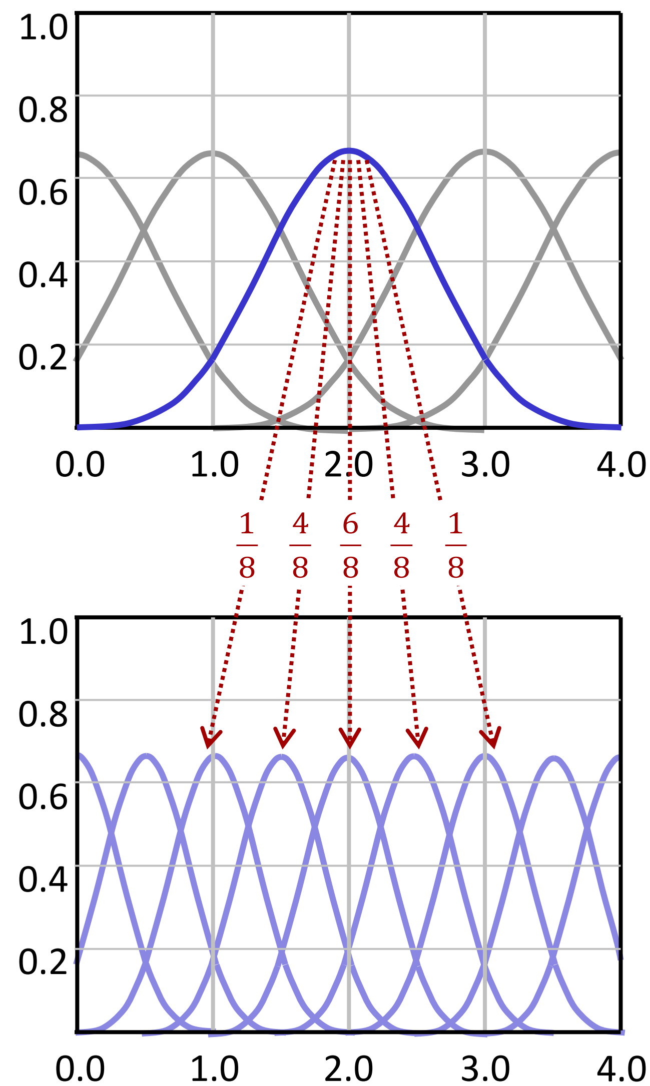
Going from basis $\mathbf{b}(t+i)$ to basis $\boldsymbol{b}(2t+i)$ we get:

$$b^{(d)}(t+j)=\frac{1}{2^d}\left(\sum_{i=0}^{d+1}\binom{d+1}{i}b^{(d)}\big(2(t+j)-i\big)\right)$$

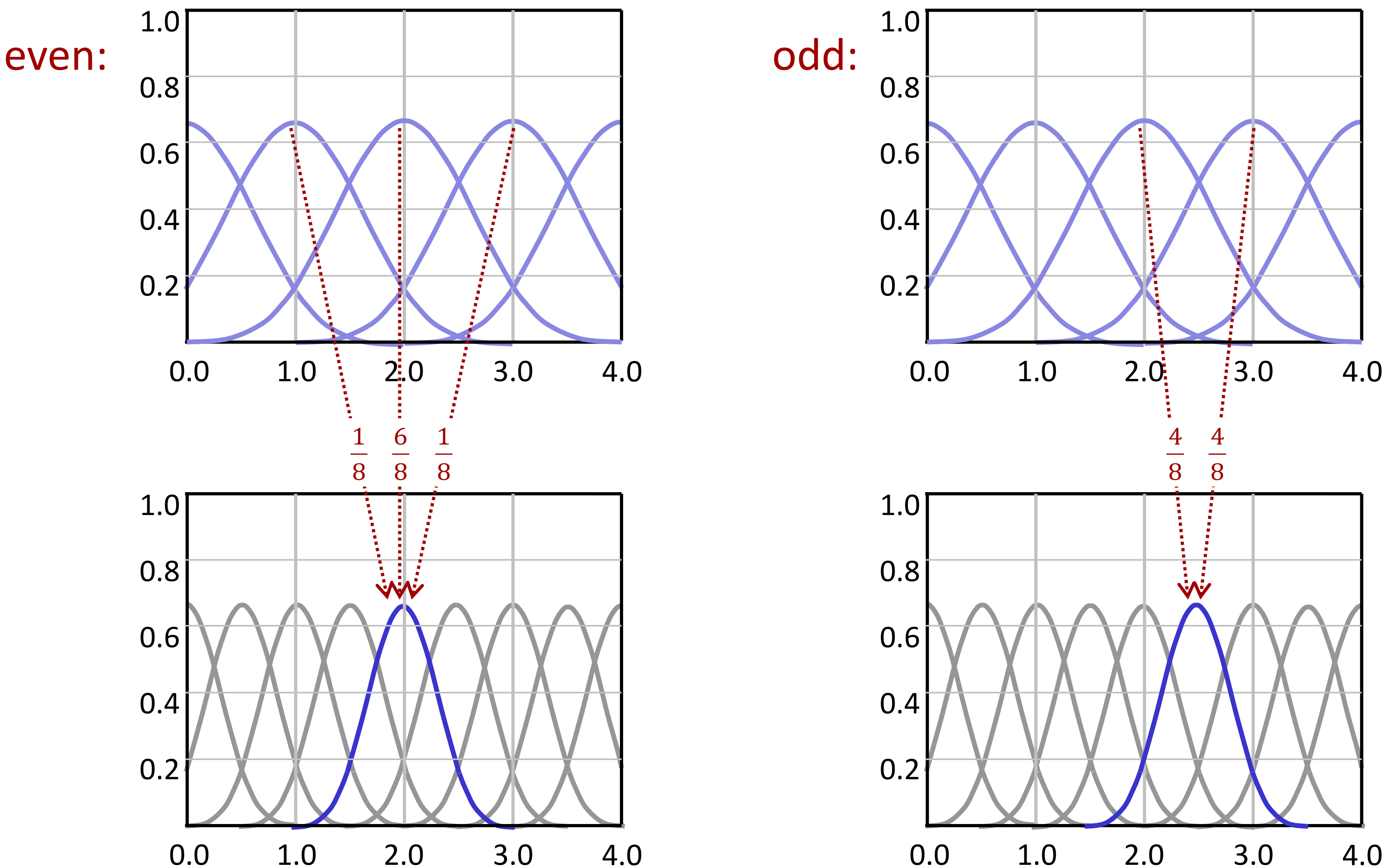
Example

Cubic case:

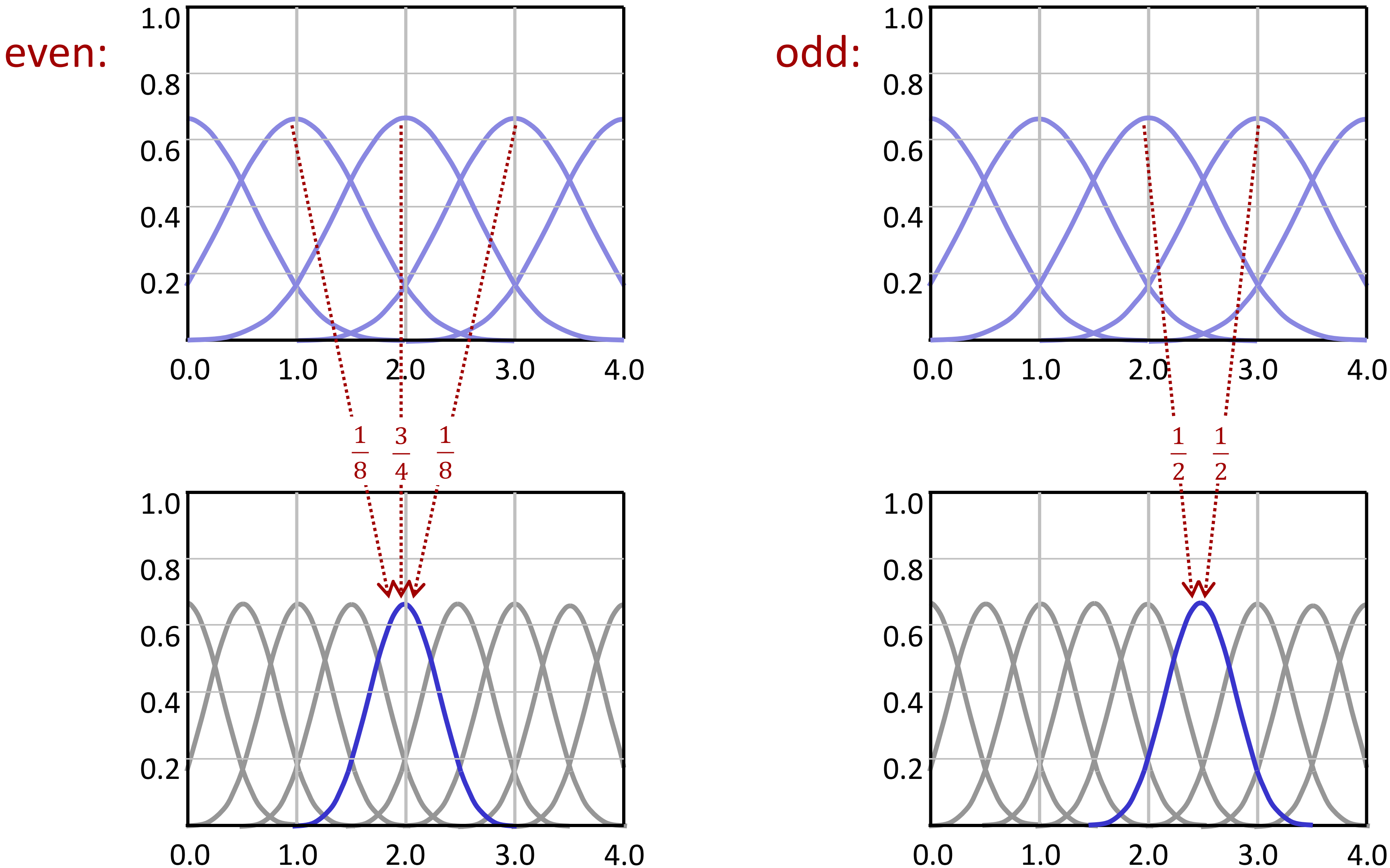
$$
\begin{aligned}
b^{(3)}(t) = &\frac{1}{8} b^{(3)}(2t) \\
&+ \frac{4}{8} b^{(3)}(2t-1) \\
&+ \frac{6}{8} b^{(3)}(2t-2) \\
&+ \frac{4}{8} b^{(3)}(2t-3) \\
&+ \frac{1}{8} b^{(3)}(2t-4)
\end{aligned}
$$



In Terms of Control Points:



In Terms of Control Points:



Subdivision Matrix

In terms of control points:

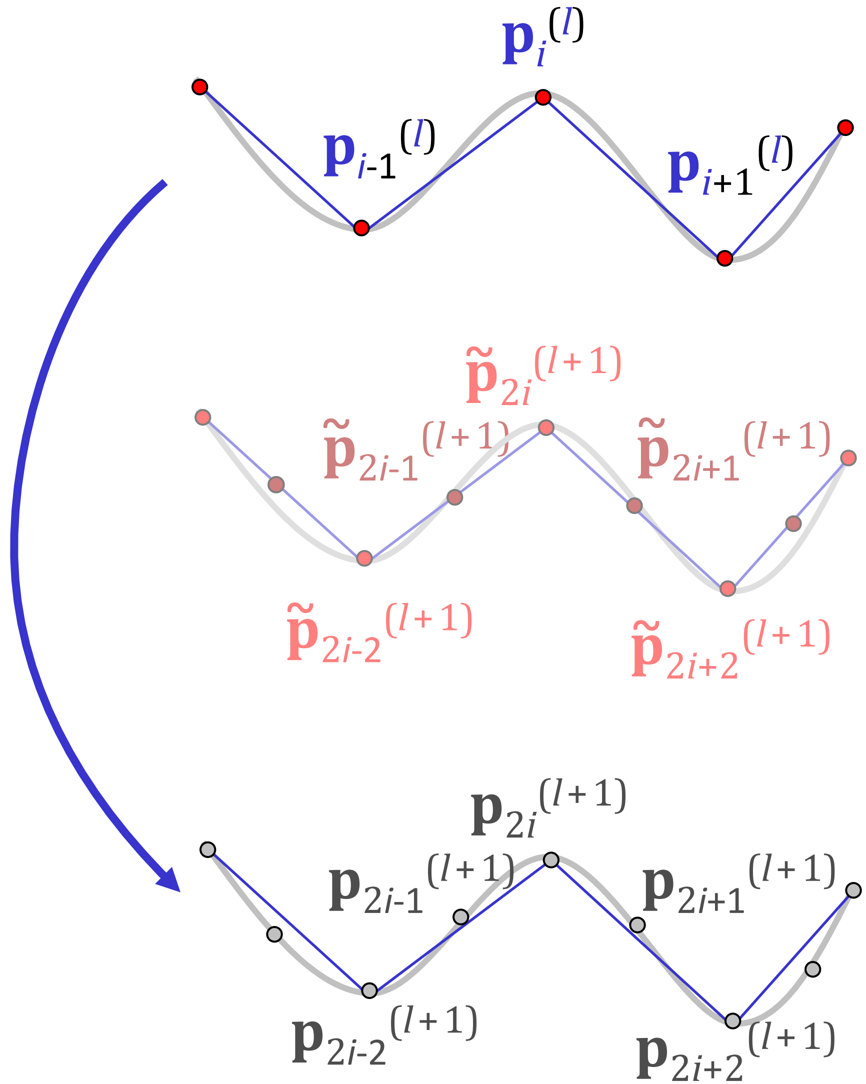
$$\begin{pmatrix} \vdots \\ p_{2i}^{[l+1]} \\ p_{2i+1}^{[l+1]} \\ \vdots \end{pmatrix} = \begin{pmatrix} \ddots & & & & & \\ & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & & \\ & & \frac{1}{2} & \frac{1}{2} & & \\ & & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\ & & & \frac{1}{2} & \frac{1}{2} & \\ & & & & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ p_{i}^{[l]} \\ p_{i+1}^{[l]} \\ \vdots \end{pmatrix}$$

Splitting and Averaging

So far:

- Splitting & averaging in one step

$$
\begin{pmatrix} \vdots \\ p_{2i}^{[l+1]} \\ p_{2i+1}^{[l+1]} \\ \vdots \end{pmatrix}
=
\begin{pmatrix}
\ddots & & & & \\
& \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & & \\
& & \frac{1}{2} & \frac{1}{2} & & \\
& & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\
& & & \frac{1}{2} & \frac{1}{2} & \\
& & & & & \ddots
\end{pmatrix}
\begin{pmatrix} \vdots \\ p_{i}^{[l]} \\ p_{i+1}^{[l]} \\ \vdots \end{pmatrix}
$$



Separate Splitting Step

Using a separate splitting matrix:

$$
\begin{pmatrix} \vdots \\ p_{2i}^{[l+1]} \\ p_{2i+1}^{[l+1]} \\ \vdots \end{pmatrix}
=
\underbrace{\begin{pmatrix}
\ddots & & & & & & \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & \\
 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & \\
 & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & \\
 & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \\
 & & & & & & \ddots
\end{pmatrix}}_{2n\times 2n \text{ averaging}}
\underbrace{\begin{pmatrix}
\ddots & & & & \\
 & 1 & & & \\
 & \frac{1}{2} & \frac{1}{2} & & \\
 & & 1 & & \\
 & & \frac{1}{2} & \frac{1}{2} & \\
 & & & & \ddots
\end{pmatrix}}_{n\times 2n \text{ splitting}}
\begin{pmatrix} \vdots \\ p_{i}^{[l]} \\ p_{i+1}^{[l]} \\ \vdots \end{pmatrix}
$$

Separate Splitting Step

Using a separate splitting matrix:

$$
\begin{pmatrix} \vdots \\ p_{2i}^{[l+1]} \\ p_{2i+1}^{[l+1]} \\ \vdots \end{pmatrix}
=
\underbrace{\begin{pmatrix}
\ddots &&&&&& \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} &&&& \\
& \frac{1}{4} & \frac{1}{2} & \frac{1}{4} &&& \\
&& \frac{1}{4} & \frac{1}{2} & \frac{1}{4} && \\
&&& \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \\
&&&&&& \ddots
\end{pmatrix}}_{2n\times 2n \text{ averaging}}
\underbrace{\begin{pmatrix}
\ddots &&&& \\
& 1 &&& \\
& \frac{1}{2} & \frac{1}{2} && \\
&& 1 && \\
&& \frac{1}{2} & \frac{1}{2} & \\
&&&& \ddots
\end{pmatrix}}_{n\times 2n \text{ splitting}}
\begin{pmatrix} \vdots \\ p_{i}^{[l]} \\ p_{i+1}^{[l]} \\ \vdots \end{pmatrix}
$$

$$
\begin{pmatrix} \vdots \\ p_{2i}^{[l+1]} \\ p_{2i+1}^{[l+1]} \\ \vdots \end{pmatrix}
=
\underbrace{\begin{pmatrix}
\ddots &&&&& \\
& \frac{1}{8} & \frac{3}{4} & \frac{1}{8} && \\
&& \frac{1}{2} & \frac{1}{2} && \\
&& \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \\
&&& \frac{1}{2} & \frac{1}{2} & \\
&&&&& \ddots
\end{pmatrix}}_{\text{one step}}
\begin{pmatrix} \vdots \\ p_{i}^{[l]} \\ p_{i+1}^{[l]} \\ \vdots \end{pmatrix}
$$

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

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

General Formula:

B-spline curve subdivision:

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

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

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

The Spectral Limit Trick

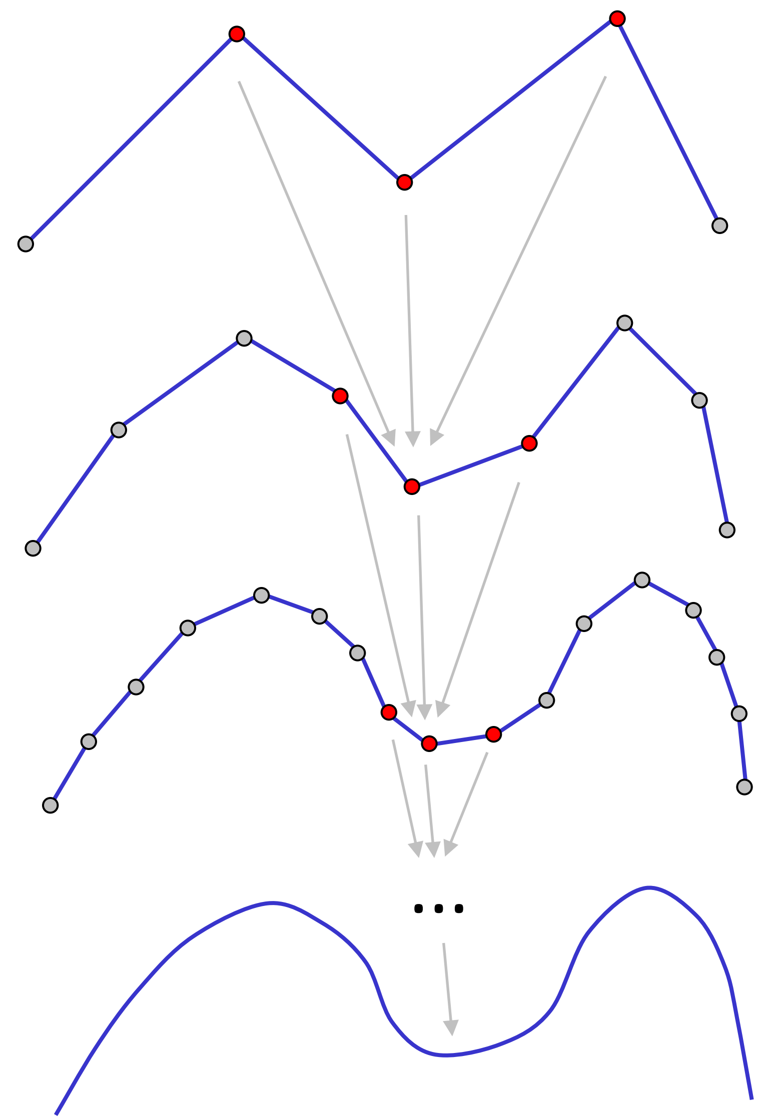
Problem:

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

Computing the Limit

Observations:

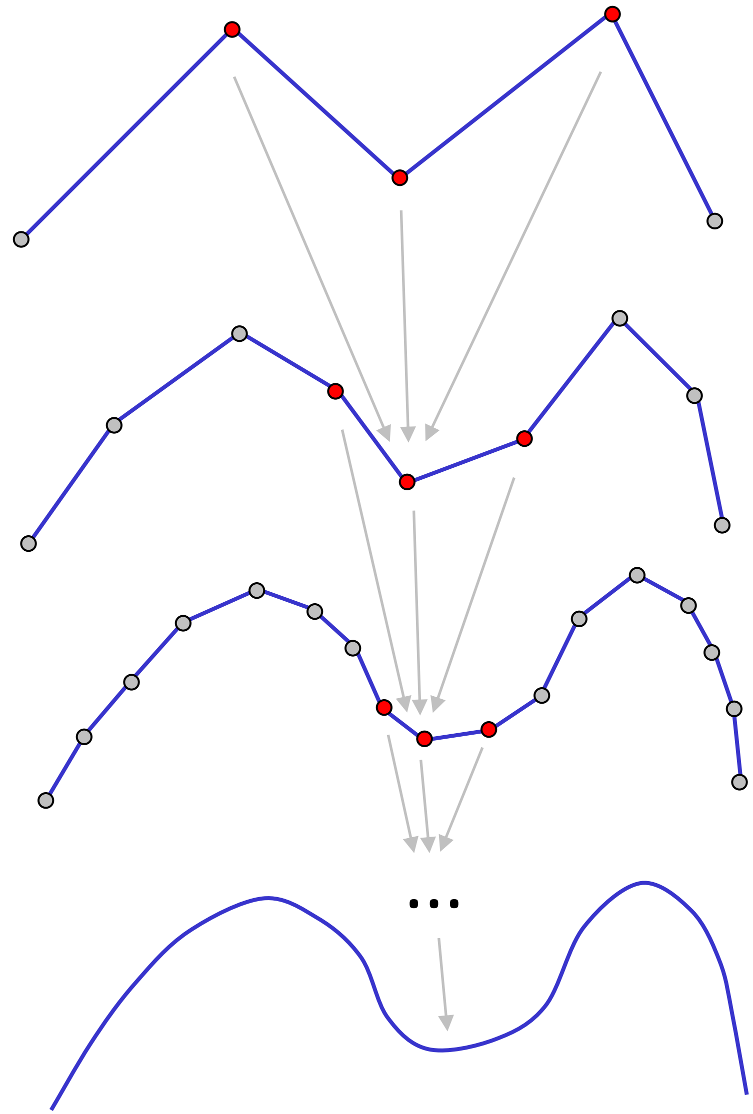
- Every curve point is influence only by a fixed number of control points
- Even stronger: Every point $\mathbf{p}^{[l+1]}$ is only influence by a small neighborhood of points in $\mathbf{p}^{[l]}$.
- To each neighborhood, the same subdivision matrix is applied (splitting & averaging)



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:

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

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

To the Limit...

This means:

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

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

To the Limit...

Spectral power:

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

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

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

To the Limit...

Spectral power:

- For cubic B-splines:

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

- and hence:

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

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)

$$\begin{pmatrix} x_{-n}^{[l+k]} \\ \vdots \\ x_{0}^{[l+k]} \\ \vdots \\ x_{+n}^{[l+k]} \end{pmatrix} = \mathbf{M}_{subdiv}^{k} \begin{pmatrix} x_{-n}^{[l]} \\ \vdots \\ x_{0}^{[l]} \\ \vdots \\ x_{+n}^{[l]} \end{pmatrix} = \mathbf{U}\mathbf{D}^{k}\mathbf{U}^{-1} \begin{pmatrix} x_{-n}^{[l]} \\ \vdots \\ x_{0}^{[l]} \\ \vdots \\ x_{+n}^{[l]} \end{pmatrix}$$

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

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{6}{8} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Affine Invariance

Affine Invariance:

- The rows of the (local) subdivision matrix must sum to one: $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{6}{8} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$
- This means: The one-vector **1** must be an eigenvector with eigenvalue 1:
  - $\mathbf{M}_{subdiv}\mathbf{1} = \mathbf{1}$
  - 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

Derivatives

One can show:

- The tangent vectors are related to the second largest eigenvalue
- Intuition:
  - Difference to neighboring points is zero for first eigenvalue/eigenvector in the limit
  - Second largest shrinks faster, spreads the tangent direction
- For a C^1 continuous curve, the second largest eigenvalue must be unique (only one eigenvector). This is a necessary condition.

Summary

For a reasonable subdivision scheme, we need at least:

- **1** must be a 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.)

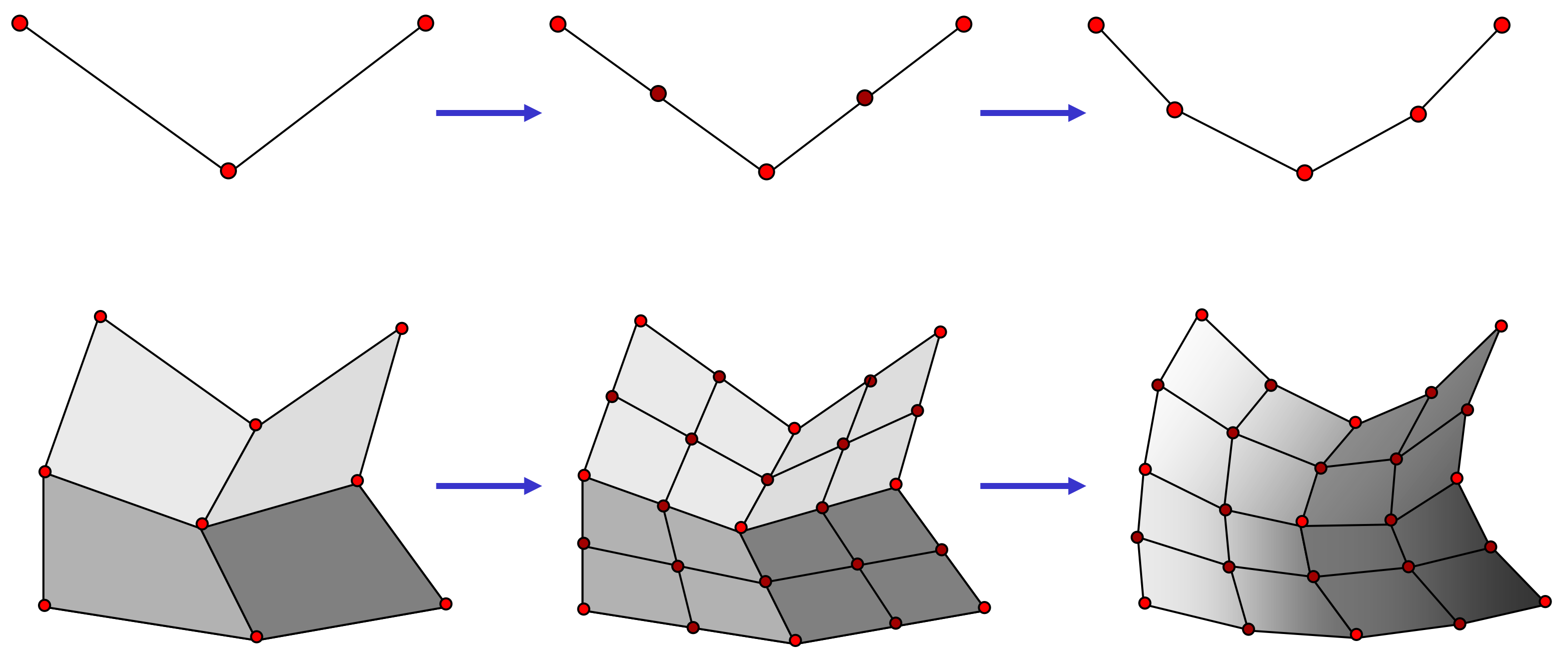
More details: Zorin, Schröder – Subdivision for Modeling and Animation, Siggraph 2000 course, available online (use Google).

B-Spline Subdivision Surfaces

B-Spline Subdivision Surfaces

B-Spline Subdivision Surfaces

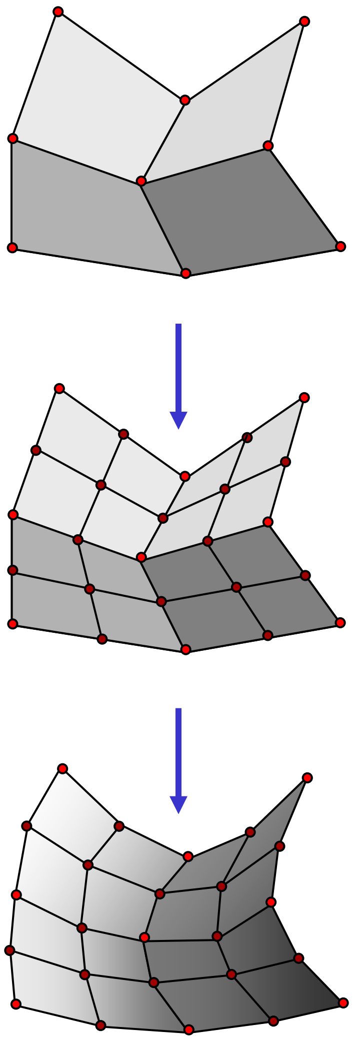
- We can apply the tensor product construction to obtain subdivision surfaces:



B-Spline Subdivision Surfaces

Tensor Product B-Spline Subdivision Surfaces:

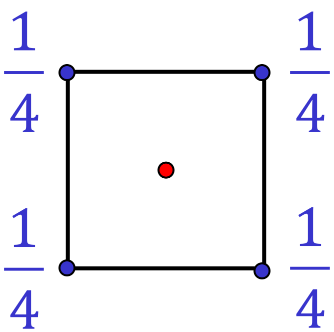
- Start with a regular quad mesh (will be relaxed later)
- In each subdivision step:
  - Divide each quad in four (quadtree subdivision)
  - Place linearly interpolated vertices
  - Apply 2-dimensional averaging mask



Subdivision and Averaging Masks

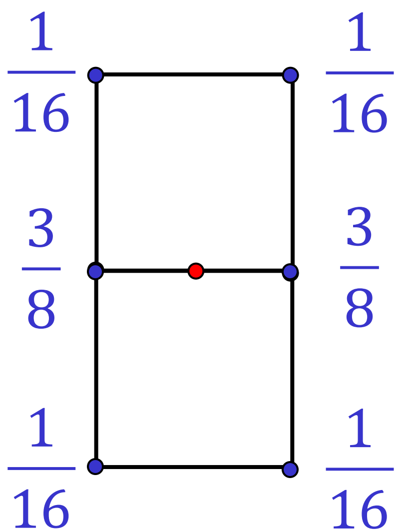
What is the subdivision mask?

- Can be derived from tensor product construction:



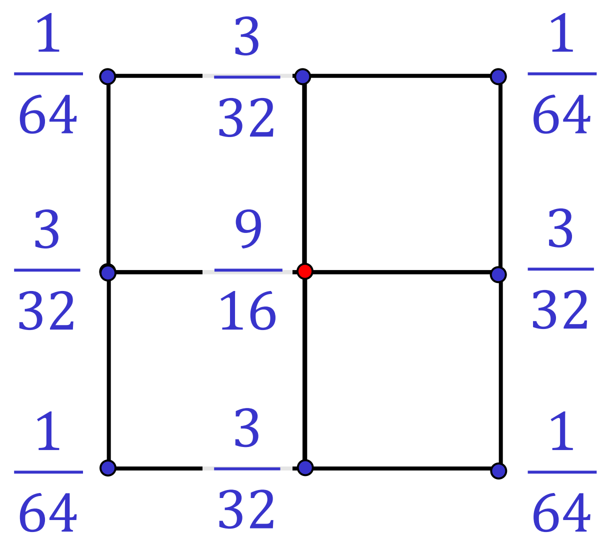
face midpoint
(odd/odd)

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



edge midpoint
(even/odd)

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



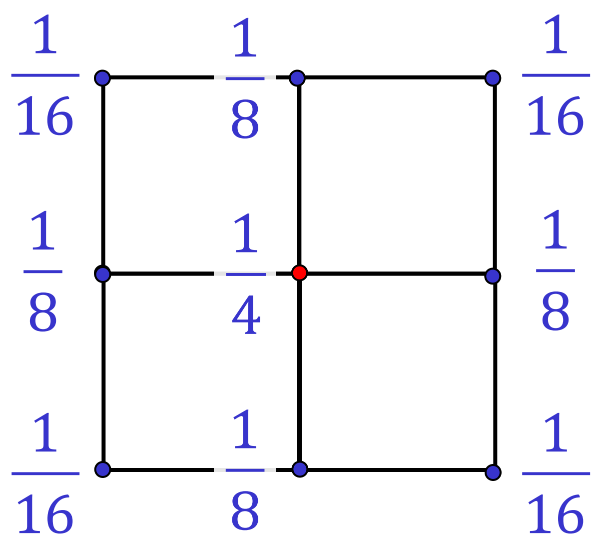
original vertex
(even/even)

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

Subdivision and Averaging Masks

What is the averaging mask?

- Can be derived from tensor product construction, too:



any (split) vertex

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

Generalizations

Generalizations:

- General degree B-spline tensor product surface subdivision rules can be derived in the same way
- Just use the 1-dimensional subdivision / averaging masks and form a tensor product mask
- Guaranteed to converge to a C^{d-1}-smooth surface
- Ok, but something is missing...

Remaining Problems

Remaining Problems:

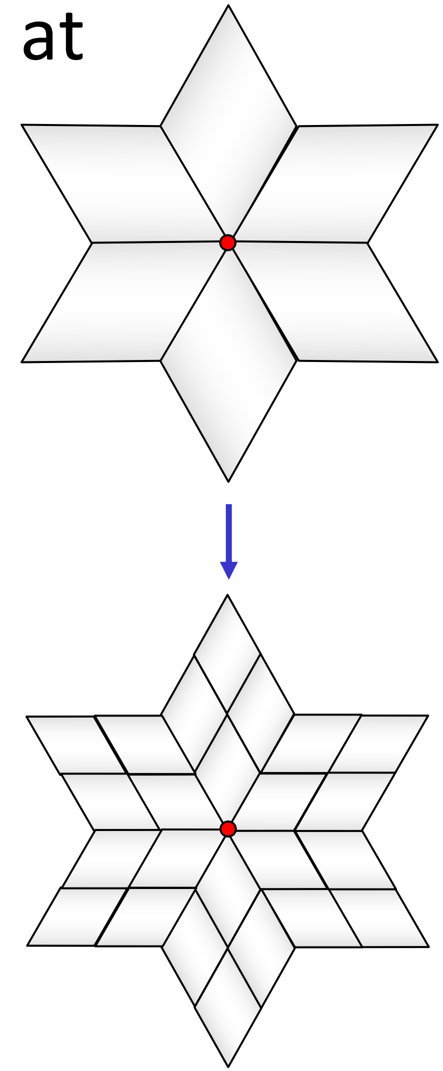
- The derived rules work only in the interior or a regular quad mesh
- We did not really gain any flexibility over the standard B-spline construction
- We still need to figure out, how to...:
  - ...handle quad meshes of arbitrary topology.
  - ...handle boundary regions.
    - Placing boundaries in 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 mesh has degree > 4
  - Extraordinary vertices are maintained by quadtree-like-subdivision
  - All new vertices are ordinary

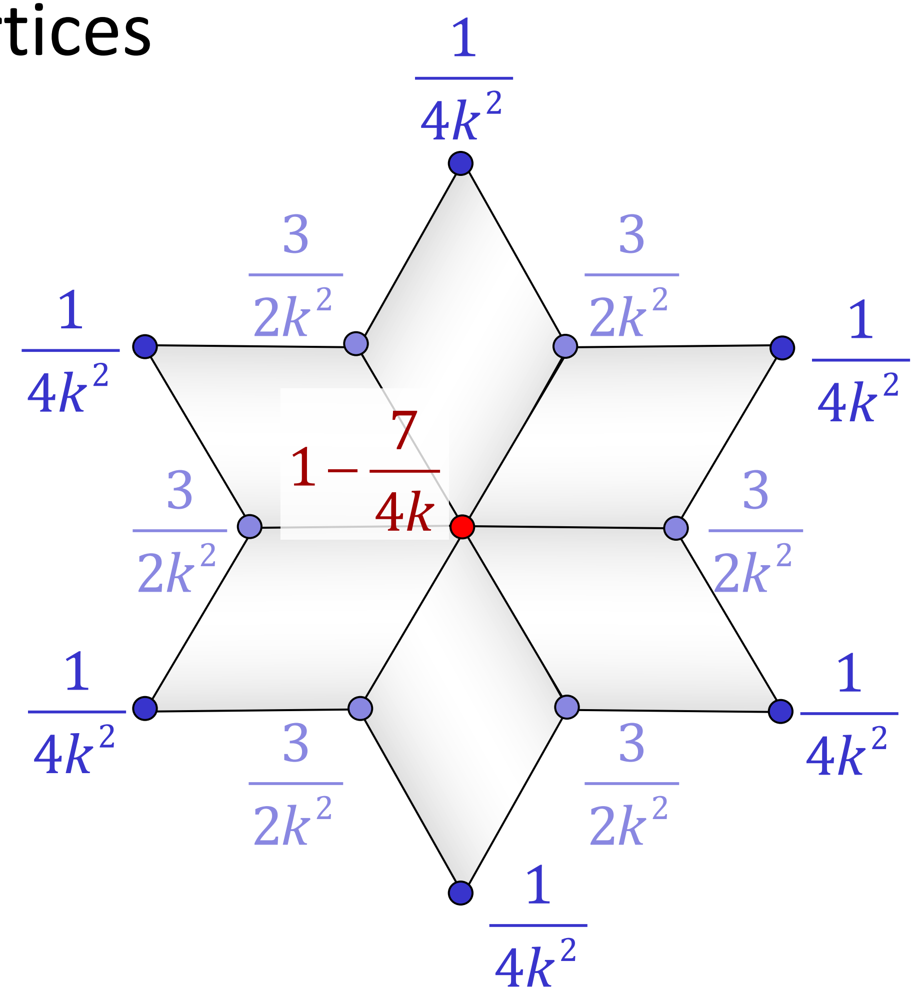


Here is the answer...

Answer: Catmull-Clark subdivision scheme at extraordinary vertices

Subdivision mask at extraordinary vertex:

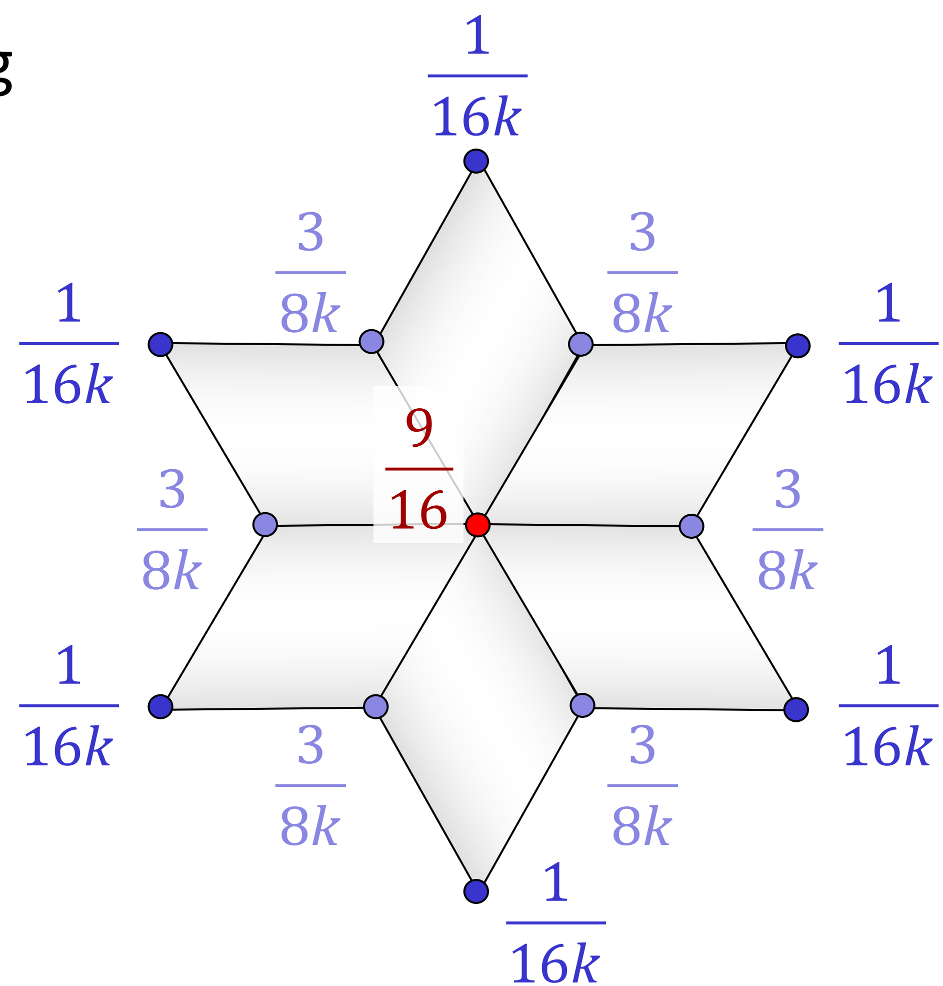
- Vertex degree k
- The surface is C^1 at extraordinary vertices



Here is the answer...

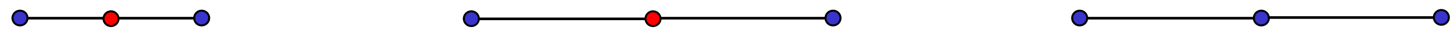
Averaging mask:

- 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

Other Subdivision Schemes

Loop, Butterfly, ...

Subdivision Zoo

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

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

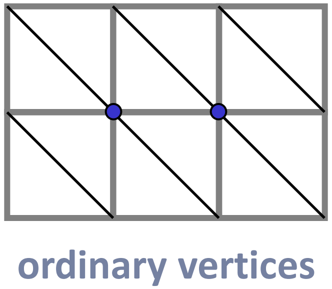
Examples of other schemes:

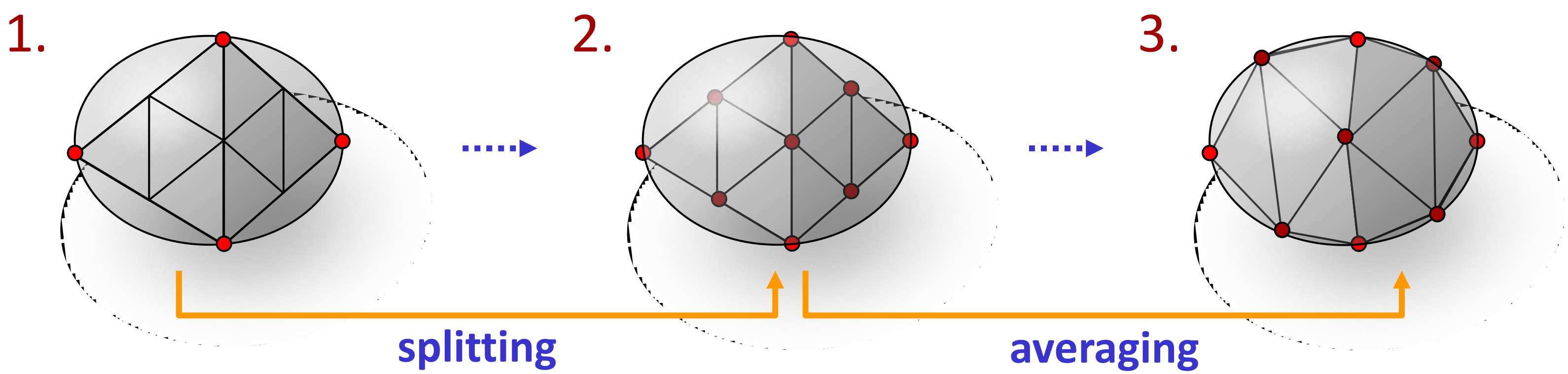
- $\sqrt{3}$ – subdivision (level of detail increases more slowly)
- Circular subdivision (used e.g. for surfaces of revolution)

Triangular Subdivision

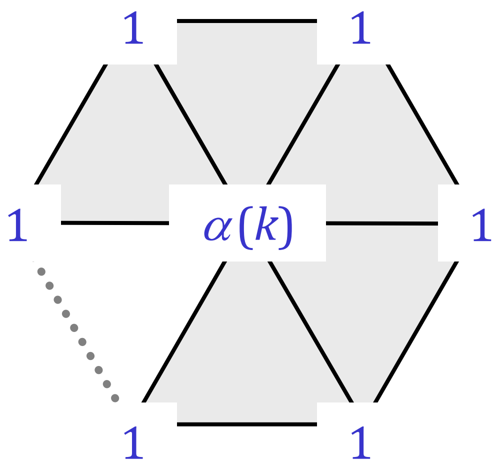
Triangular Subdivision:

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





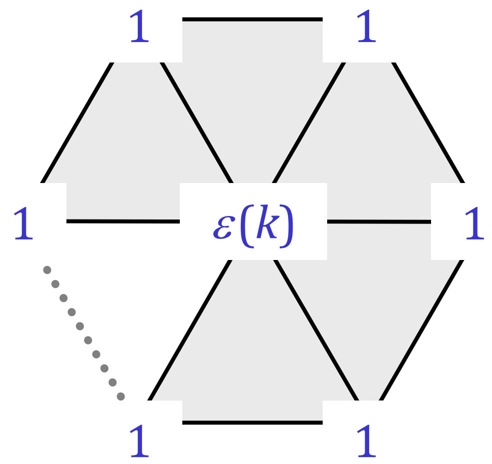
Loop Subdivision



averaging mask

$\alpha(k) = k(1 - \beta(k))/\beta(k)$

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



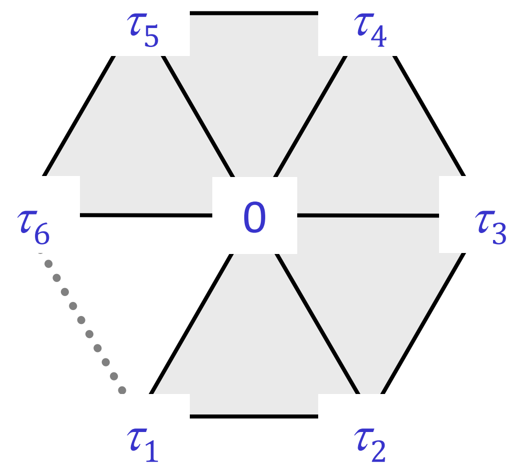
evaluation (limit) mask

$\varepsilon(k) = 3k/(4\beta(k))$

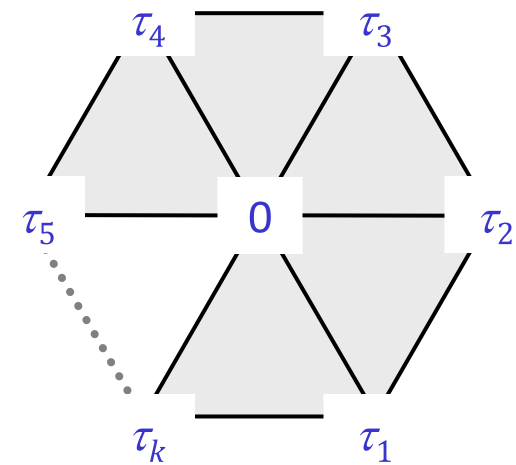
$$\frac{1}{4} \;\text{———}\; \frac{1}{2} \;\text{———}\; \frac{1}{4}$$

boundary / sharp crease mask

Loop Subdivision



tangent mask 1



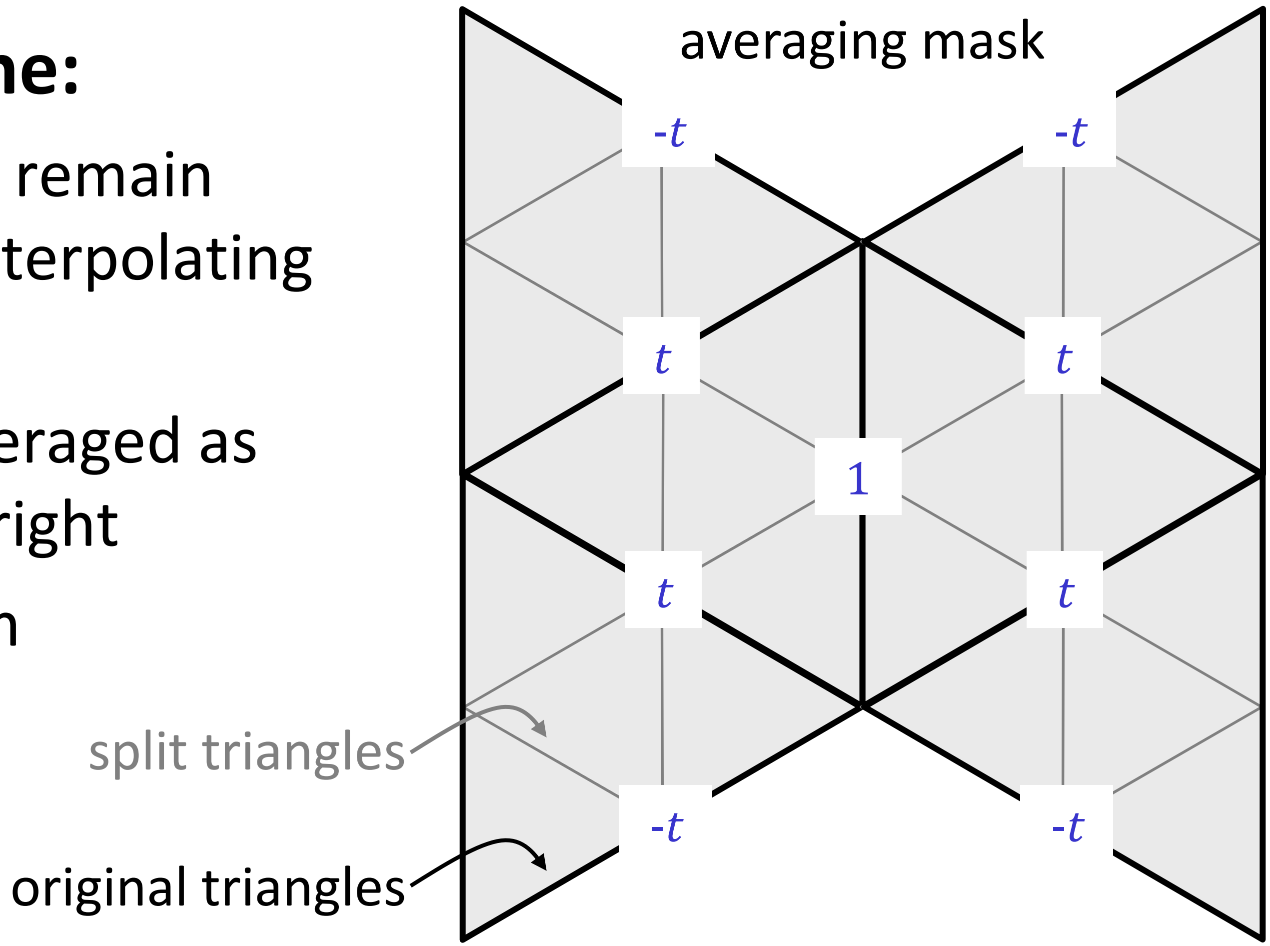
tangent mask 2

$$\tau_i = \cos(2\pi i/k)$$

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



$t \in [0...1/8]$

$0 \triangleq$ *polyhedral*, $1/8 \triangleq$ *smooth*

Connection to Wavelet

Nested Function Spaces and Wavelet Bases

Wavelets

Wavelets:

- Wavelet bases are an interesting tool that is very useful in computer graphics (and many other fields)
- Main idea:
  - Define a basis for a function space
  - Separate large scale from fine scale information
  - “Level-of-detail” basis
  - This allows for compression, scale-dependent processing and often leads to numerical more efficient and stable algorithms (e.g. wavelet preconditioning)

Nested Function Space

Starting Point:

- We consider a set of nested function spaces:
  $V_0 \subset V_1 \subset \ldots \subset V_i \subset \ldots$
- This means: with increasing index, a larger number of functions can be represented.
- We can find a basis B_i for V_i and B_{i+1} for V_{i+1} such that $B_i \subset B_{i+1}$.

Special Spaces

We consider a special case:

- The basis functions in Bi are dilates of one and the same "mother" basis function $\phi(t)$:

  $\phi_i^j = \phi(2^j t - i)$

  $B_j = \{\phi_i^j\}_i$
- The $\phi_i^j(t)$ are called scaling functions.
- With increaing j, the scaling functions should provide a more detailed representation of the *same* function.
- In order to achieve this, we need an additional property...

Subdivision Property

Nested Spaces:

- We want to have $\mathrm{span}(B_j) \subset \mathrm{span}(B_{j+1})$.
- For this, we need to be able to reproduce the ϕ_i^j by linear combinations of functions ϕ_i^{j+1}.
- This is the same condition we need for our geometric subdivision schemes.
- This means: Each subdivision scheme creates a corresponding nested scaling space (and vice versa).

Wavelet Bases

What we get so far:

- The span of the set $B_0 \cup B_1 \cup \ldots \cup B_n$ can describe functions at various level of detail.
- Problem:
  - This is not a basis. The set is still redundant.
  - The finest resolution set would be sufficient.
- Solution:
  - Remove redundant functions
  - Store only differences

Wavelet Bases

Constructing a wavelet basis:

- Start with $V_0 = \text{span}(B_0)$. Add all $\{\phi_i^0\}_i$ to the wavelet basis.
- Then add *wavelet basis functions* $\{\psi_i^1\}_i$ that form a basis of $V_1 \setminus V_0$ to this set.
- Continue with the next level of wavelets $\{\psi_i^1\}_i$ that form a basis of $V_2 \setminus V_1$.
- And so on.

Same structure:

- We again want a dilation property: $\psi_i^j = \psi(2^j t - i)$
- $\psi(t)$ is called the "mother wavelet"

Wavelet Basis

We get:

- Scaling functions for the coarsest level function space.
- Wavelets for the missing details at level 1,2,3,...,*n*.

$$
V_n \left\{ \begin{array}{ccc}
V_0 & \rightarrow & \{\phi_i^0\} \\
V_1 \setminus V_0 & \rightarrow & \{\psi_i^2\}_i \\
V_2 \setminus V_1 & \rightarrow & \{\psi_i^3\}_i \\
 & & \vdots \\
V_n \setminus V_{n\text{-}1} & \rightarrow & \{\psi_i^n\}_i
\end{array} \right.
$$

Wavelet Basis Construction

How to choose the wavelet basis?

- Assume we are given the scaling functions.
- We (usually) still have a lot of freedom for choosing the wavelet basis.
- Desirable properties:
  - Small support (efficiency)
  - Orthogonality (good for compression)
  - Symmetry (useful for signal processing)
  - High order of smoothness (depends on scaling function)
- Not all can be obtained at the same time
- Various compromises have been proposed in literature

Example: Haar basis

Haar Basis:

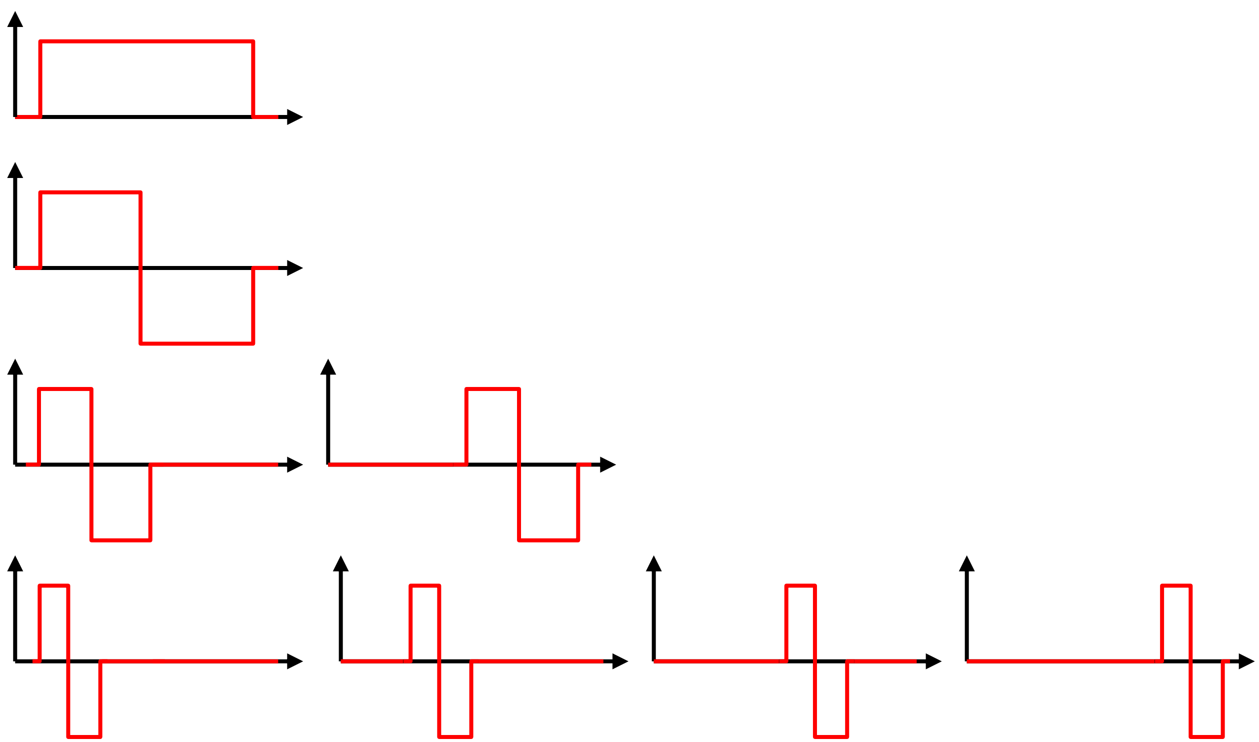
- Corresponds to zero-order subdivision curves (piecewise constant)

$$V_0 \rightarrow \{\phi_i^0\}$$

$$V_1 \setminus V_0 \rightarrow \{\psi_i^2\}_i$$

$$V_2 \setminus V_1 \rightarrow \{\psi_i^3\}_i$$

$$V_3 \setminus V_2 \rightarrow \{\psi_i^n\}_i$$



Function compression

Function compression recipe:

- Transform to a wavelet basis
- Make sure the basis functions have unit norm (integral square norm)
- Keep the only the largest wavelet coefficients and the top level scaling function coefficent(s)
- If you do this with a smooth wavelet basis (such as higher order B-spline subdivision based wavelets), combined with arithmetic encoding of the coefficients, you basically get JPEG 2000.

Image Compression Example

Haar basis wavelet compression:

- This is 10 lines of code...

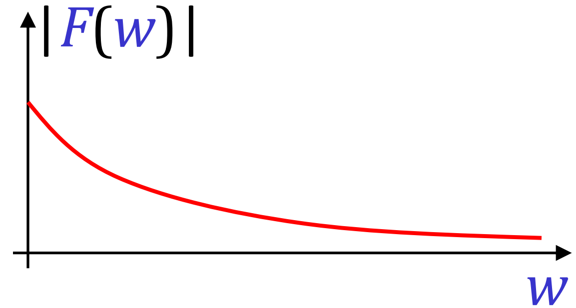
Stochastic Subdivision

Nice Fractal Landscapes & the Similar

Fractal Brownian Motion

Modeling rough surfaces:

- Can be modeled as "Fractal Brownian Motion" (FBM).
- The signal f is a noise signal with known power spectrum.
- In the Fourier domain, we have:
  $|F(w)| \sim w^{-h}$,
  i.e.: the energy in the spectrum decays with frequency.



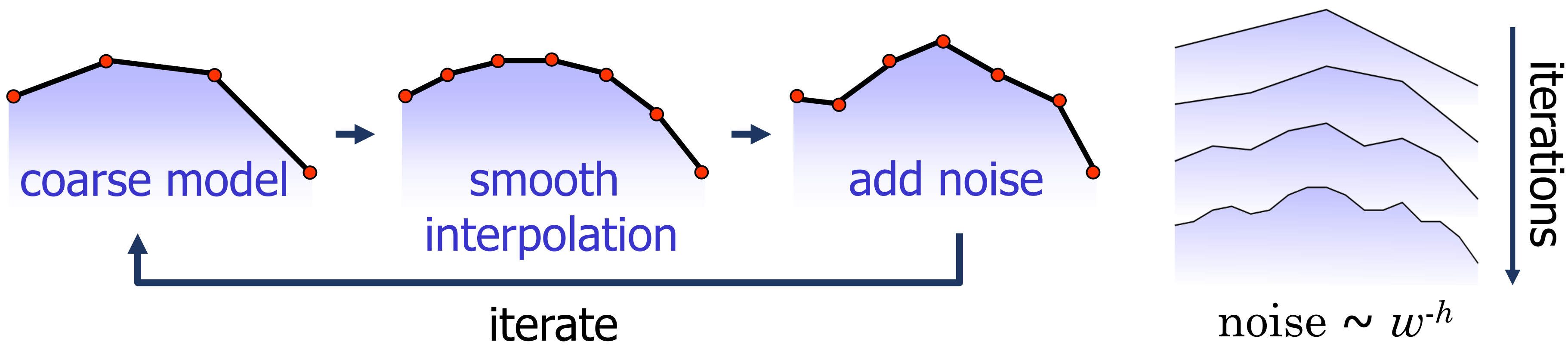
- The phase is purely random.
- The speed of decay h (typ. $h \in [1..2]$) controls the roughness ("fractal exponent").

Stochastic Subdivision

Modeling rough surfaces with subdivision:

- A coarse base mesh models the shape.
- Then apply a smooth subdivision scheme (e.g., Catmull-Clark)
- At each subdivision level, add random noise to the new control points, with amplitude proportional to 2^{-hl}.
- Idea:
  - The large subdivision steps (small l) control the low frequency bands, the small scale steps control the higher frequencies.
  - Create an FBM spectrum function incrementally.

Stochastic Subdivision

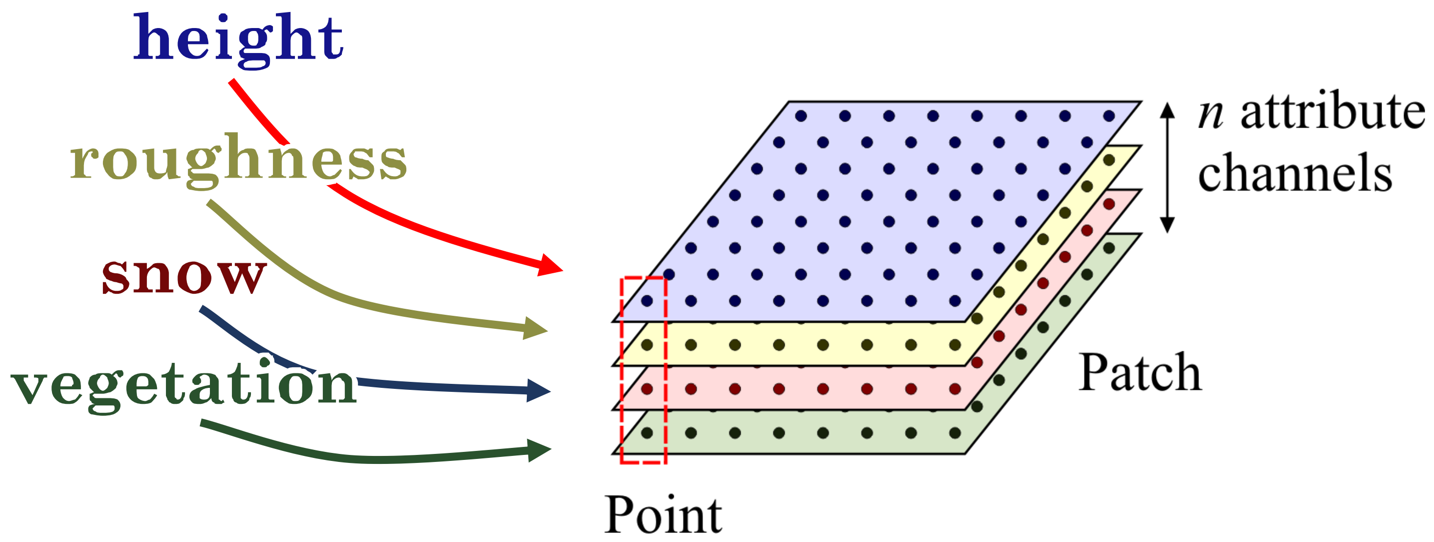


Example

Example Application:

- Creating “nice looking” landscapes with “multi-channel fractals”.
- This is rather art than science...

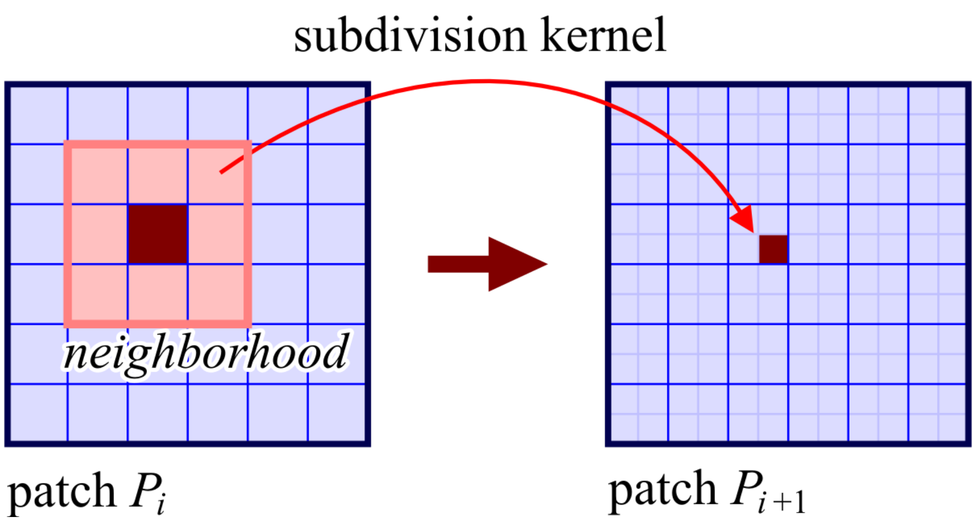
Multi-Channel Fractals



Modeling Primitive: Regular Patch

- Represent geometry as regularly sampled patches
- Multiple attributes per point

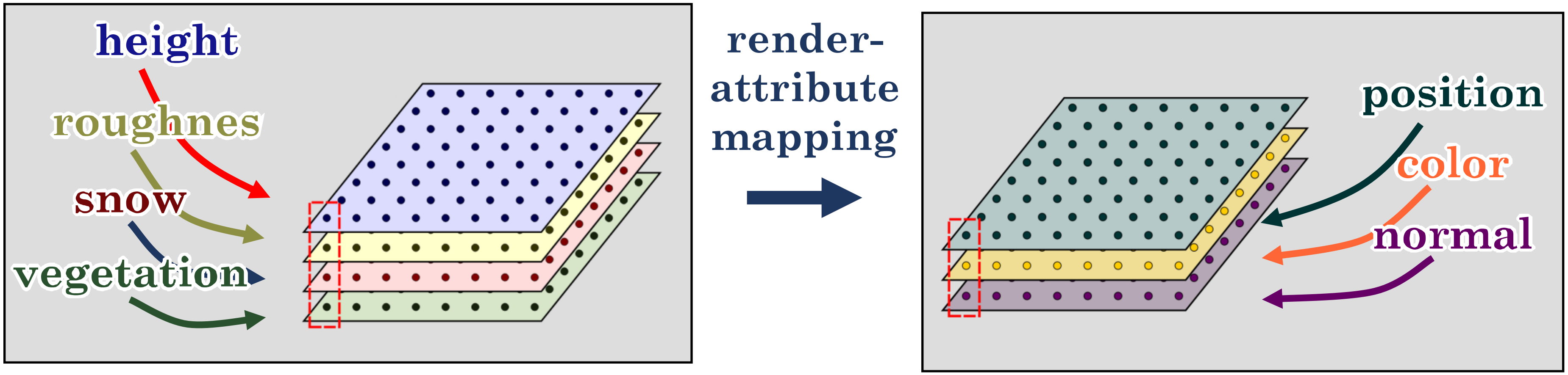
Modeling by Subdivision



Modeling by Subdivision:

- User defined initial patch
- Double resolution (iteratively)
- New values depend on local neighborhood

Rendering Mapping



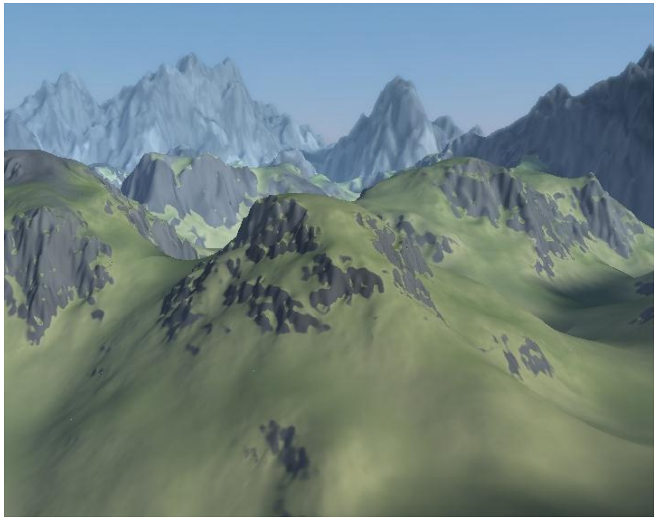
Final Mapping before Rendering:

- Convert point attributes to rendering attributes
- Transform from “semantic” to “visual” representation

Fractal Modeling

Multi-Channel Fractals:

- Channels for *height*, *roughness*, *vegetation*, etc.
- Each channel is a fractal “heightfield”
- Mutual influence of channels during subdivision
  - roughness $\rightarrow$ noise decay
  - height $\rightarrow$ roughness
  - vegetation $\rightarrow$ roughness
  - height $\rightarrow$ vegetation



Example