Interpolatory Surface Subdivision based on Geometry-Driven Parameterizations

C. Beccari[*], G. Casciola[*], L. Romani[‡]

[*] University of Bologna, Italy

[‡] University of Milano-Bicocca, Italy

Minisymposium on New Trends in Numerical Approximation and Applications

SIMAI, Rome, 17 September 2008

Outline

Subdivision surfaces

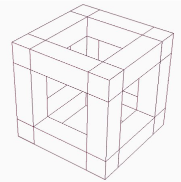
Basic idea:

- An initial 2-manifold network of vertices, edges and facets - called *control mesh* or *control polyhedron* - is refined by computing new vertices and joining them up to form a new mesh with the same connectivity.
- By repeating recursively this operation (if the refinement construction is suitable), the mesh converges towards a *limit surface*.

Subdivision surfaces

Basic idea:

- An initial 2-manifold network of vertices, edges and facets - called *control mesh* or *control polyhedron* - is refined by computing new vertices and joining them up to form a new mesh with the same connectivity.
- By repeating recursively this operation (if the refinement construction is suitable), the mesh converges towards a *limit surface*.

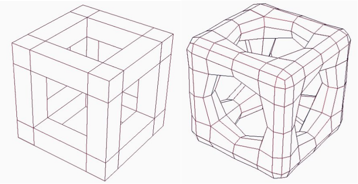


Subdivision surfaces

Basic idea:

- An initial 2-manifold network of vertices, edges and facets - called *control mesh* or *control polyhedron* - is refined by computing new vertices and joining them up to form a new mesh with the same connectivity.

- By repeating recursively this operation (if the refinement construction is suitable), the mesh converges towards a *limit surface*.

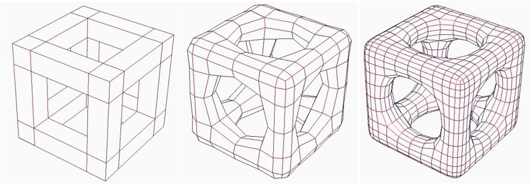


Subdivision surfaces

Basic idea:

- An initial 2-manifold network of vertices, edges and facets - called *control mesh* or *control polyhedron* - is refined by computing new vertices and joining them up to form a new mesh with the same connectivity.

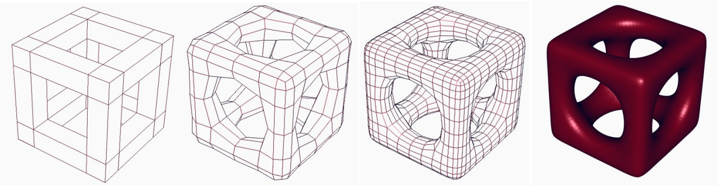
- By repeating recursively this operation (if the refinement construction is suitable), the mesh converges towards a *limit surface*.



Subdivision surfaces

Basic idea:

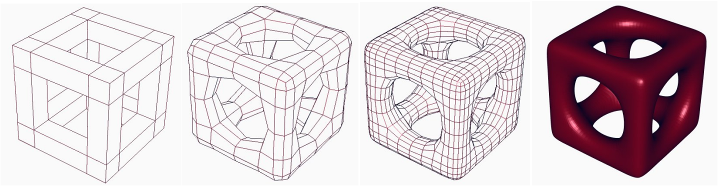
- An initial 2-manifold network of vertices, edges and facets - called *control mesh* or *control polyhedron* - is refined by computing new vertices and joining them up to form a new mesh with the same connectivity.
- By repeating recursively this operation (if the refinement construction is suitable), the mesh converges towards a *limit surface*.



Subdivision surfaces

Basic idea:

- An initial 2-manifold network of vertices, edges and facets - called *control mesh* or *control polyhedron* - is refined by computing new vertices and joining them up to form a new mesh with the same connectivity.
- By repeating recursively this operation (if the refinement construction is suitable), the mesh converges towards a *limit surface*.



Main advantage over splines: subdivision naturally handles meshes of *arbitrary topology* (i.e. made of any number of quadrilateral or triangular facets meeting at a vertex).

... and applications

Catmull-Clark (C-C) subdivision:

- approximating
- local
- C^2 almost everywhere
- easy to implement.

... and applications

Catmull-Clark (C-C) subdivision:

- approximating
- local
- C^2 almost everywhere
- easy to implement.

- Widely-established in animation, computer graphics, videogames design, etc.
- Already used in modeling systems (Maya, 3dStudioMax, Lightwave, etc.).
- Recently introduced in CAD systems (CATIA).

... and applications

Catmull-Clark (C-C) subdivision:

- approximating
- local
- C^2 almost everywhere
- easy to implement.

- Widely-established in animation, computer graphics, videogames design, etc.
- Already used in modeling systems (Maya, 3dStudioMax, Lightwave, etc.).
- Recently introduced in CAD systems (CATIA).

Subdivision surfaces & CAD

- interpolation

Subdivision surfaces & CAD

- interpolation
- spline quality
  (non-uniform parameterization)

Subdivision surfaces & CAD

- interpolation
- spline quality (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)

Subdivision surfaces & CAD

- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction

Subdivision surfaces & CAD

- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features

Subdivision surfaces & CAD

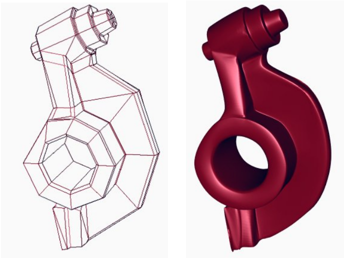
- interpolation
- spline quality (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values

Subdivision surfaces & CAD

- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness $(C^k, k \geq 1)$
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values

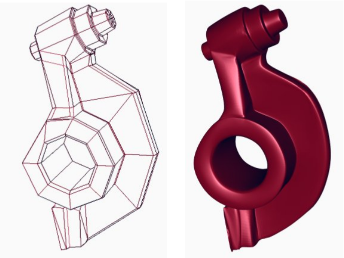
Interpolatory subdivision in CAD systems

- Interpolation is highly desirable because of its intuitive link with the starting mesh.



Interpolatory subdivision in CAD systems

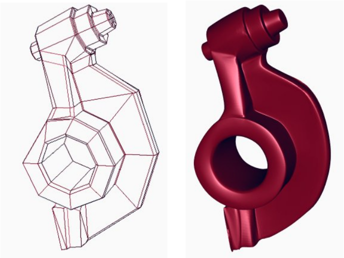
- Interpolation is highly desirable because of its intuitive link with the starting mesh.



- Achievable through:
  - modification of the standard C-C subdivision rules, so that the limit surface fits the interpolation constraints (Halstead et al. '93, Levin '99, Nasri et al. '01, Schaefer et al. '04)

Interpolatory subdivision in CAD systems

- Interpolation is highly desirable because of its intuitive link with the starting mesh.



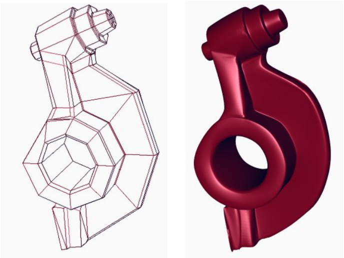
- Achievable through:
  - modification of the standard C-C subdivision rules, so that the limit surface fits the interpolation constraints (Halstead et al. '93, Levin '99, Nasri et al. '01, Schaefer et al. '04)

OR

  - **"natural" interpolation:** at each refinement iteration all the current vertices are retained, becoming points of the limit surface itself.

Interpolatory subdivision in CAD systems

- Interpolation is highly desirable because of its intuitive link with the starting mesh.



- Achievable through:
  - modification of the standard C-C subdivision rules, so that the limit surface fits the interpolation constraints (Halstead et al. '93, Levin '99, Nasri et al. '01, Schaefer et al. '04)

OR

  - **"natural" interpolation:** at each refinement iteration all the current vertices are retained, becoming points of the limit surface itself.

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.

Surface subdivision from curve subdivision

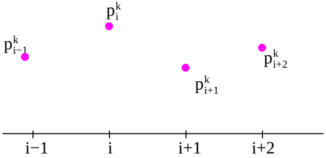
- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h = -1, ..., 2$, defined on a grid $2^{-k}\mathbb{Z}$

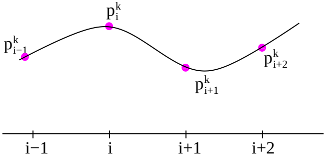


Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h = -1, \dots, 2$, defined on a grid $2^{-k}\mathbb{Z}$



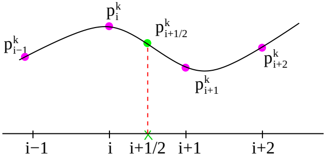
- interpolate the data by a function $f(x)$

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h = -1, ..., 2$, defined on a grid $2^{-k}\mathbb{Z}$



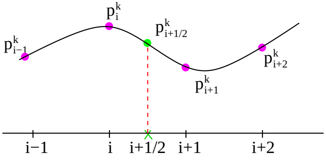
- interpolate the data by a function $f(x)$
- compute $p^k_{i+\frac{1}{2}} = f\left(\frac{1}{2^{k+1}}\right)$

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h=-1,...,2$, defined on a grid $2^{-k}\mathbb{Z}$



- interpolate the data by a function $f(x)$
- compute $p^k_{i+\frac{1}{2}} = f\left(\frac{1}{2^{k+1}}\right)$
- set $p^{k+1}_{2i} = p^k_i$

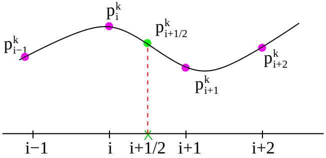
$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}}$$

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h = -1, \ldots, 2$, defined on a grid $2^{-k}\mathbb{Z}$



- interpolate the data by a function $f(x)$
- compute $p^k_{i+\frac{1}{2}} = f\left(\frac{1}{2^{k+1}}\right)$
- set $p^{k+1}_{2i} = p^k_i$

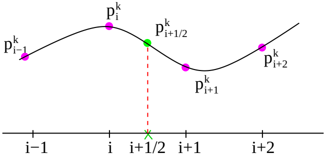
$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = \sum_{h=-1}^{2} a^k_{h+1} p^k_{i+h}$$

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h = -1, \ldots, 2$, defined on a grid $2^{-k}\mathbb{Z}$



- interpolate the data by a function $f(x)$
- compute $p^k_{i+\frac{1}{2}} = f\left(\frac{1}{2^{k+1}}\right)$
- set $p^{k+1}_{2i} = p^k_i$

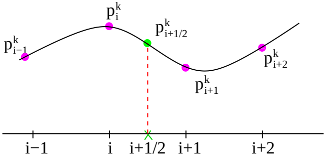
$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = \sum_{h=-1}^{2} a^k_{h+1} p^k_{i+h}$$

Surface subdivision from curve subdivision

- A spline surface is defined as tensor product of spline curves.
- Also the design of surface subdivision rules often starts from curve refinement equations.

- **INTERPOLATORY 4-pt SUBDIVISION**

Given the 4 points p^k_{i+h}, $h = -1, ..., 2$, defined on a grid $2^{-k}\mathbb{Z}$



- interpolate the data by a function $f(x)$
- compute $p^k_{i+\frac{1}{2}} = f\left(\frac{1}{2^{k+1}}\right)$
- set $p^{k+1}_{2i} = p^k_i$

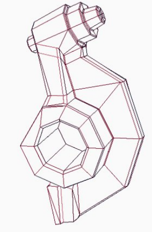
  $p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = \sum_{h=-1}^{2} a^k_{h+1} p^k_{i+h}$

- The curve scheme should be:
  - local
  - non-uniform: to follow closely the behavior of the starting polyline and modify the shape in a local way
  - non-stationary: to reproduce conic sections.

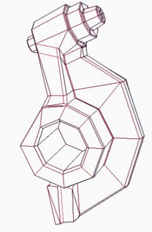
Subdivision surfaces & CAD

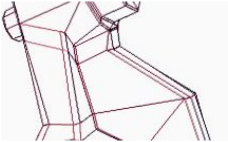
- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values

Interpolatory curve subdivision with spline quality
(B., Casciola, Romani - Tønsberg '08)

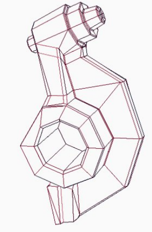


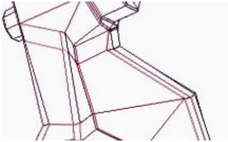
Interpolatory curve subdivision with spline quality
(B., Casciola, Romani - Tønsberg '08)

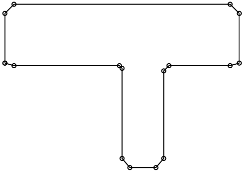




Interpolatory curve subdivision with spline quality
(B., Casciola, Romani - Tønsberg ’08)

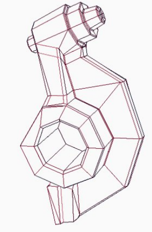


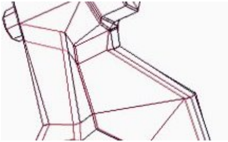


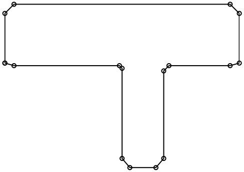


Interpolatory curve subdivision with spline quality

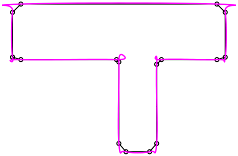
(B., Casciola, Romani - Tønsberg ’08)



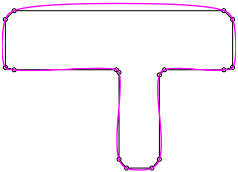




Splines naturally support non-uniform parameterizations!



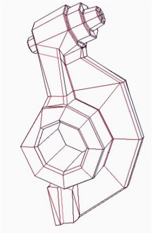
UNIFORM SPLINE INTERPOLATION

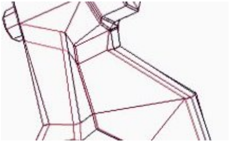


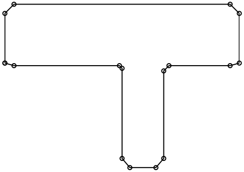
NON-UNIFORM SPLINE INTERPOLATION
centripetal parameterization

Interpolatory curve subdivision with spline quality

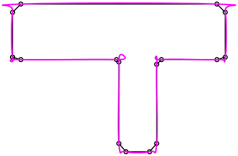
(B., Casciola, Romani - Tønsberg '08)



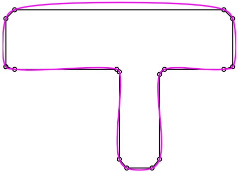




Splines naturally support non-uniform parameterizations!



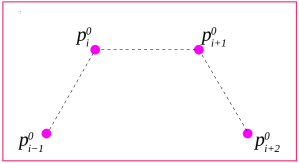
UNIFORM SPLINE INTERPOLATION



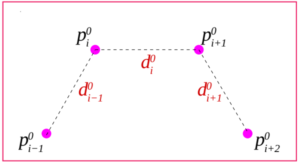
NON-UNIFORM SPLINE INTERPOLATION
centripetal parameterization

➥ Non-uniform parameters improve shape quality and alleviate undesired ondulations (Floater '08, Dyn et al. '07,...).

Spline based 4-pt subdivision



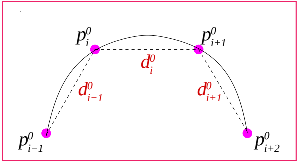
Spline based 4-pt subdivision



- initial parameterization

$$d_i^0 = \|p_{i+1}^0 - p_i^0\|_2^{\frac{1}{2}}$$ **centripetal**

Spline based 4-pt subdivision



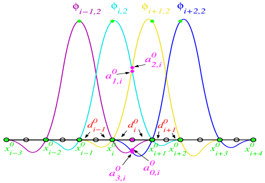
- initial parameterization

  $d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$ **centripetal**

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h}\phi_{i+h,2}(x),\ x \in [x_i, x_{i+1}]$

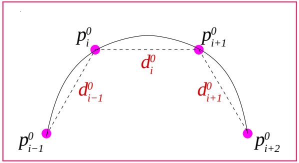
  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

Spline based 4-pt subdivision



PROPERTIES of $\phi(x)$:

- arbitrary knots
- compact support $[x_{i-2}, x_{i+2}]$
- C^1 continuity
- polynomial reproduction up to deg. 2



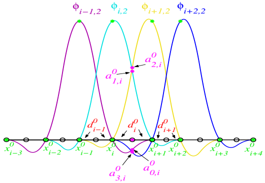
- initial parameterization

  $d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$ **centripetal**

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h} \phi_{i+h,2}(x),\ x \in [x_i, x_{i+1}]$

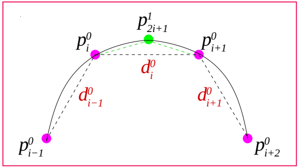
  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

Spline based 4-pt subdivision



PROPERTIES of $\phi(x)$:

- arbitrary knots
- compact support $[x_{i-2}, x_{i+2}]$
- C^1 continuity
- polynomial reproduction up to deg. 2



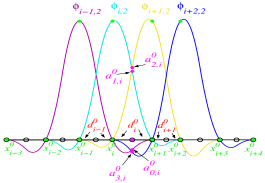
- initial parameterization

  $d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$ **centripetal**

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h}\phi_{i+h,2}(x),\ x \in [x_i, x_{i+1}]$
  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

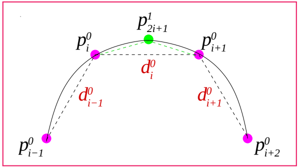
$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f(\tfrac{x}{2}) \ldots$$

Spline based 4-pt subdivision



PROPERTIES of $\phi(x)$:

- arbitrary knots
- compact support $[x_{i-2}, x_{i+2}]$
- C^1 continuity
- polynomial reproduction up to deg. 2



- initial parameterization

$$d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}} \qquad \textbf{centripetal}$$

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h} \phi_{i+h,2}(x), \; x \in [x_i, x_{i+1}]$
  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

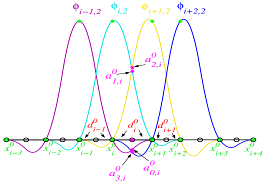
$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f\left(\frac{x}{2}\right) \ldots$$

➡ **REFINEMENT EQUATIONS** at step k

$$p^{k+1}_{2i} = p^k_i$$

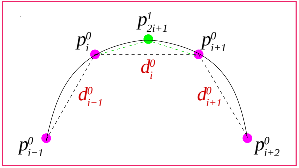
$$p^{k+1}_{2i+1} = a^k_{0,i}\, p^k_{i-1} + a^k_{1,i}\, p^k_i + a^k_{2,i}\, p^k_{i+1} + a^k_{3,i}\, p^k_{i+2}$$

Spline based 4-pt subdivision



PROPERTIES of $\phi(x)$:

- arbitrary knots
- compact support $[x_{i-2}, x_{i+2}]$
- C^1 continuity
- polynomial reproduction up to deg. 2



- initial parameterization

$$d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$$ **centripetal**

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h}\phi_{i+h,2}(x),\ x \in [x_i, x_{i+1}]$

  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f\left(\frac{x}{2}\right) \ldots$$ ➡ **REFINEMENT EQUATIONS** at step k

$$p^{k+1}_{2i} = p^k_i$$

$$p^{k+1}_{2i+1} = a^k_{0,i}\, p^k_{i-1} + a^k_{1,i}\, p^k_i + a^k_{2,i}\, p^k_{i+1} + a^k_{3,i}\, p^k_{i+2}$$

$$\lambda^k_i = \frac{d^k_{i-1}}{d^k_i},\ \mu^k_i = \frac{d^k_{i+1}}{d^k_i} \Longrightarrow$$

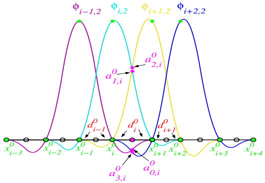
$$a^k_{0,i} = -\frac{1}{8\lambda^k_i(\lambda^k_i+1)}$$

$$a^k_{1,i} = \frac{1+3\lambda^k_i+\mu^k_i+4\lambda^k_i\mu^k_i}{\lambda^k_i(\mu^k_i+1)}$$

$$a^k_{2,i} = \frac{1+3\mu^k_i+\lambda^k_i+4\lambda^k_i\mu^k_i}{\mu^k_i(\lambda^k_i+1)}$$

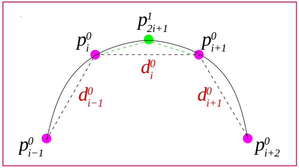
$$a^k_{3,i} = -\frac{1}{8\mu^k_i(1+\mu^k_i)}$$

Spline based 4-pt subdivision



PROPERTIES of $\phi(x)$:

- arbitrary knots
- compact support $[x_{i-2}, x_{i+2}]$
- C^1 continuity
- polynomial reproduction up to deg. 2



- initial parameterization

  $$d_i^0 = \|p_{i+1}^0 - p_i^0\|_2^{\frac{1}{2}}$$ **centripetal**

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h} \phi_{i+h,2}(x),\ x \in [x_i, x_{i+1}]$

  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f(\frac{x}{2})$ ... ➠ **REFINEMENT EQUATIONS** at step k

$$p^{k+1}_{2i} = p^k_i$$

$$p^{k+1}_{2i+1} = a^k_{0,i}\, p^k_{i-1} + a^k_{1,i}\, p^k_i + a^k_{2,i}\, p^k_{i+1} + a^k_{3,i}\, p^k_{i+2}$$

$$\lambda^k_i = \frac{d^k_{i-1}}{d^k_i},\ \mu^k_i = \frac{d^k_{i+1}}{d^k_i} \Longrightarrow$$

$$a^k_{0,i} = -\frac{1}{8\lambda^k_i(\lambda^k_i+1)}$$

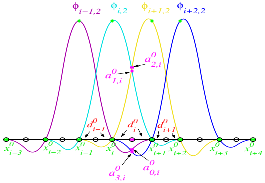
$$a^k_{1,i} = \frac{1+3\lambda^k_i+\mu^k_i+4\lambda^k_i\mu^k_i}{\lambda^k_i(\mu^k_i+1)}$$

$$a^k_{2,i} = \frac{1+3\mu^k_i+\lambda^k_i+4\lambda^k_i\mu^k_i}{\mu^k_i(\lambda^k_i+1)}$$

$$a^k_{3,i} = -\frac{1}{8\mu^k_i(1+\mu^k_i)}$$

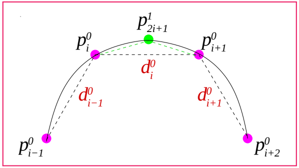
$a^k_{0,i}, \ldots, a^k_{3,i}$ coincide with the values of the deg-2 basis functions $\phi_{i-1,2}, \ldots, \phi_{i+2,2}$ defined on the refined knot-partition with $d^0_i/2^k$-length intervals

Spline based 4-pt subdivision



PROPERTIES of $\phi(x)$:

- arbitrary knots
- compact support $[x_{i-2}, x_{i+2}]$
- C^1 continuity
- polynomial reproduction up to deg. 2



- initial parameterization

  $d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$ **centripetal**

- $f(x) = \sum_{h=-1}^{2} p^0_{i+h} \phi_{i+h,2}(x),\ x \in [x_i, x_{i+1}]$

  $\phi(x)$ non-uniform local interpolatory cardinal spline basis (Chui et. al. '96)

$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f(\frac{x}{2})$ ... ➡ **REFINEMENT EQUATIONS** at step k

PARAMETERS UPDATING

$$p^{k+1}_{2i} = p^k_i$$

$$p^{k+1}_{2i+1} = a^k_{0,i}\, p^k_{i-1} + a^k_{1,i}\, p^k_i + a^k_{2,i}\, p^k_{i+1} + a^k_{3,i}\, p^k_{i+2}$$

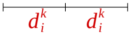
$$\lambda^k_i = \frac{d^k_{i-1}}{d^k_i},\ \mu^k_i = \frac{d^k_{i+1}}{d^k_i} \quad \Longrightarrow$$

$$a^k_{0,i} = -\frac{1}{8\lambda^k_i(\lambda^k_i+1)}$$

$$a^k_{1,i} = \frac{1+3\lambda^k_i+\mu^k_i+4\lambda^k_i\mu^k_i}{\lambda^k_i(\mu^k_i+1)}$$

$$a^k_{2,i} = \frac{1+3\mu^k_i+\lambda^k_i+4\lambda^k_i\mu^k_i}{\mu^k_i(\lambda^k_i+1)}$$

$$a^k_{3,i} = -\frac{1}{8\mu^k_i(1+\mu^k_i)}$$



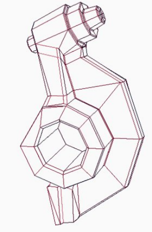
$$d^{k+1}_{2i} = d^{k+1}_{2i+1} = d^k_i$$

$a^k_{0,i}, \ldots, a^k_{3,i}$ coincide with the values of the deg-2 basis functions $\phi_{i-1,2}, \ldots, \phi_{i+2,2}$ defined on the refined knot-partition with $d^0_i / 2^k$-length intervals

Subdivision surfaces & CAD

- interpolation
- spline quality (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values

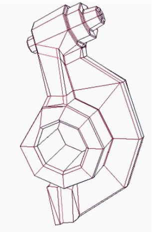
Subdivision surfaces & CAD

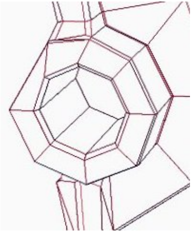


- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values

Subdivision surfaces & CAD

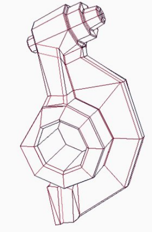
- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values

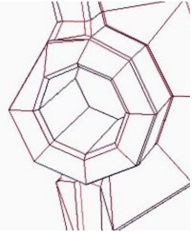


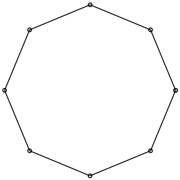


Subdivision surfaces & CAD

- interpolation
- spline quality
  (non-uniform parameterization)
- smoothness (C^k, $k \geq 1$)
- polynomial reproduction
- exact reproduction of circular features
- exact evaluation at arbitrary parameter values



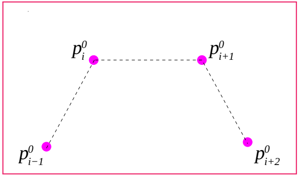




➥ Whenever the initial points of a section curve lie on a circular arc, this arc must be automatically detected and reproduced.

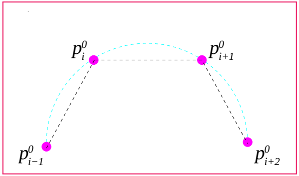
4-pt subdivision exact for exponentials

(B., Casciola, Romani - CAGD ’07)



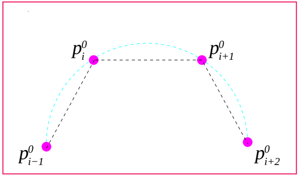
4-pt subdivision exact for exponentials

(B., Casciola, Romani - CAGD '07)



4-pt subdivision exact for exponentials
(B., Casciola, Romani - CAGD '07)

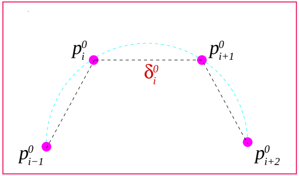
- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx},\ s > 0$



4-pt subdivision exact for exponentials

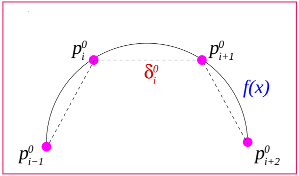
(B., Casciola, Romani - CAGD '07)

- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx}$, $s > 0$
- initial parameterization $\delta_i^0 = e^{i\frac{sx}{2}} + e^{-i\frac{sx}{2}}$



4-pt subdivision exact for exponentials
(B., Casciola, Romani - CAGD '07)

- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx},\ s > 0$
- initial parameterization $\delta_i^0 = e^{i\frac{sx}{2}} + e^{-i\frac{sx}{2}}$

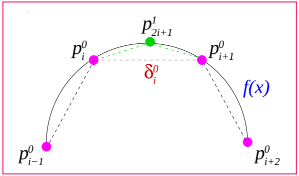


- **REMARK:** if the initial points are placed at equally-spaced parameter values on a circle, $f(x)$ reproduces the circle passing through them

4-pt subdivision exact for exponentials

(B., Casciola, Romani - CAGD '07)

- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx},\ s > 0$
- initial parameterization $\delta_i^0 = e^{i\frac{sx}{2}} + e^{-i\frac{sx}{2}}$



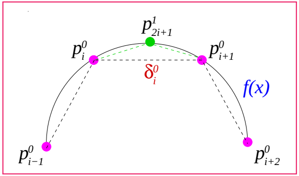
$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f(\tfrac{x}{2})\ \ldots$$

- **REMARK:** if the initial points are placed at equally-spaced parameter values on a circle, $f(x)$ reproduces the circle passing through them

4-pt subdivision exact for exponentials

(B., Casciola, Romani - CAGD '07)

- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx},\ s > 0$
- initial parameterization $\quad \delta_i^0 = e^{i\frac{sx}{2}} + e^{-i\frac{sx}{2}}$



$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f\left(\frac{x}{2}\right) \ldots$$

- **REMARK:** if the initial points are placed at equally-spaced parameter values on a circle, $f(x)$ reproduces the circle passing through them

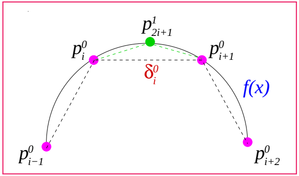
- **REFINEMENT EQUATIONS** at step k:

$$p^{k+1}_{2i} = p^k_i$$
$$p^{k+1}_{2i+1} = a^k_{0,i}\, p^k_{i-1} + a^k_{1,i}\, p^k_i + a^k_{2,i}\, p^k_{i+1} + a^k_{3,i}\, p^k_{i+2}$$

4-pt subdivision exact for exponentials

(B., Casciola, Romani - CAGD '07)

- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx}$, $s > 0$
- initial parameterization $\quad \delta_i^0 = e^{i\frac{sx}{2}} + e^{-i\frac{sx}{2}}$



- **REMARK:** if the initial points are placed at equally-spaced parameter values on a circle, $f(x)$ reproduces the circle passing through them

$$p^{k+1}_{2i+1} = p^k_{i+\frac{1}{2}} = f\left(\frac{x}{2}\right) \ldots$$

- **REFINEMENT EQUATIONS** at step k:

$$p^{k+1}_{2i} = p^k_i$$
$$p^{k+1}_{2i+1} = a^k_{0,i}\, p^k_{i-1} + a^k_{1,i}\, p^k_i + a^k_{2,i}\, p^k_{i+1} + a^k_{3,i}\, p^k_{i+2}$$

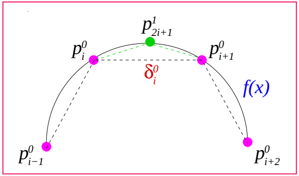
$$\delta^k_i = e^{i\frac{s}{2^{k+1}}} + e^{-i\frac{s}{2^{k+1}}} \quad \Longrightarrow$$

$$a^k_{0,i} = -\frac{1}{8\delta^k_i(1+\delta^k_i)}$$
$$a^k_{1,i} = \frac{(2\delta^k_i+1)^2}{8\delta^k_i(1+\delta^k_i)}$$
$$a^k_{2,i} = \frac{(2\delta^k_i+1)^2}{8\delta^k_i(1+\delta^k_i)}$$
$$a^k_{3,i} = -\frac{1}{8\delta^k_i(1+\delta^k_i)}$$

4-pt subdivision exact for exponentials

(B., Casciola, Romani - CAGD '07)

- $f(x) = c_0 + c_1\, x + c_2\, e^{isx} + c_3\, e^{-isx},\ s > 0$
- initial parameterization $\quad \delta_i^0 = e^{i\frac{sx}{2}} + e^{-i\frac{sx}{2}}$



- **REMARK:** if the initial points are placed at equally-spaced parameter values on a circle, $f(x)$ reproduces the circle passing through them

$$p_{2i+1}^{k+1} = p_{i+\frac{1}{2}}^k = f\left(\tfrac{x}{2}\right) \ldots$$

- **REFINEMENT EQUATIONS** at step k:

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k\, p_{i-1}^k + a_{1,i}^k\, p_i^k + a_{2,i}^k\, p_{i+1}^k + a_{3,i}^k\, p_{i+2}^k$$

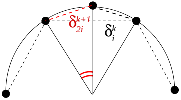
$$\delta_i^k = e^{i\frac{s}{2^{k+1}}} + e^{-i\frac{s}{2^{k+1}}} \quad \Longrightarrow$$

$$a_{0,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{1,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{2,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{3,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$

- **PARAMETERS UPDATING**

$$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = \sqrt{\frac{1+\delta_i^k}{2}} \quad \text{(half "arc length")}$$

parameters are not recomputed at each step, but they are updated automatically!



The NULI4^{++} curve scheme

- **4-pt REFINEMENT EQUATIONS**

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k\, p_{i-1}^k + a_{1,i}^k\, p_i^k + a_{2,i}^k\, p_{i+1}^k + a_{3,i}^k\, p_{i+2}^k$$

The NULI4^{++} curve scheme

- **4-pt REFINEMENT EQUATIONS**

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k \, p_{i-1}^k + a_{1,i}^k \, p_i^k + a_{2,i}^k \, p_{i+1}^k + a_{3,i}^k \, p_{i+2}^k$$

- **REFINEMENT COEFFICIENTS**

The NULI4^{++} curve scheme

- **4-pt REFINEMENT EQUATIONS**

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k \, p_{i-1}^k + a_{1,i}^k \, p_i^k + a_{2,i}^k \, p_{i+1}^k + a_{3,i}^k \, p_{i+2}^k$$

- **REFINEMENT COEFFICIENTS**

SPLINE-BASED 4-pt

$$a_{0,i}^k = -\frac{1}{8\lambda_i^k(\lambda_i^k+1)}$$
$$a_{1,i}^k = \frac{1+3\lambda_i^k+\mu_i^k+4\lambda_i^k\mu_i^k}{\lambda_i^k(\mu_i^k+1)}$$
$$a_{2,i}^k = \frac{1+3\mu_i^k+\lambda_i^k+4\lambda_i^k\mu_i^k}{\mu_i^k(\lambda_i^k+1)}$$
$$a_{3,i}^k = -\frac{1}{8\mu_i^k(1+\mu_i^k)}$$

CIRCLE-PRESERVING 4-pt

$$a_{0,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{1,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{2,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{3,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$

The NULI4^{++} curve scheme

- **4-pt REFINEMENT EQUATIONS**

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k\, p_{i-1}^k + a_{1,i}^k\, p_i^k + a_{2,i}^k\, p_{i+1}^k + a_{3,i}^k\, p_{i+2}^k$$

- **REFINEMENT COEFFICIENTS**

SPLINE-BASED 4-pt

$$a_{0,i}^k = -\frac{1}{8\lambda_i^k(\lambda_i^k+1)}$$
$$a_{1,i}^k = \frac{1+3\lambda_i^k+\mu_i^k+4\lambda_i^k\mu_i^k}{\lambda_i^k(\mu_i^k+1)}$$
$$a_{2,i}^k = \frac{1+3\mu_i^k+\lambda_i^k+4\lambda_i^k\mu_i^k}{\mu_i^k(\lambda_i^k+1)}$$
$$a_{3,i}^k = -\frac{1}{8\mu_i^k(1+\mu_i^k)}$$

$$\lambda_i = \mu_i$$

$$=$$

CIRCLE-PRESERVING 4-pt

$$a_{0,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{1,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{2,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{3,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$

The NULI4^{++} curve scheme

- **4-pt REFINEMENT EQUATIONS**

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k\, p_{i-1}^k + a_{1,i}^k\, p_i^k + a_{2,i}^k\, p_{i+1}^k + a_{3,i}^k\, p_{i+2}^k$$

- **REFINEMENT COEFFICIENTS**

SPLINE-BASED 4-pt

$$a_{0,i}^k = -\frac{1}{8\lambda_i^k(\lambda_i^k+1)}$$
$$a_{1,i}^k = \frac{1+3\lambda_i^k+\mu_i^k+4\lambda_i^k\mu_i^k}{\lambda_i^k(\mu_i^k+1)}$$
$$a_{2,i}^k = \frac{1+3\mu_i^k+\lambda_i^k+4\lambda_i^k\mu_i^k}{\mu_i^k(\lambda_i^k+1)}$$
$$a_{3,i}^k = -\frac{1}{8\mu_i^k(1+\mu_i^k)}$$

$$\lambda_i = \mu_i$$

$$=$$

CIRCLE-PRESERVING 4-pt

$$a_{0,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{1,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{2,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{3,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$

- Define $\lambda_i^k = \frac{d_{i-1}^k}{d_i^k}\,\delta_i^k$ and $\mu_i^k = \frac{d_{i+1}^k}{d_i^k}\,\delta_i^k$
- Then the two subdivision schemes are represented by the same set of coefficients

The NULI4^{++} curve scheme

- **4-pt REFINEMENT EQUATIONS**

$$p_{2i}^{k+1} = p_i^k$$
$$p_{2i+1}^{k+1} = a_{0,i}^k\, p_{i-1}^k + a_{1,i}^k\, p_i^k + a_{2,i}^k\, p_{i+1}^k + a_{3,i}^k\, p_{i+2}^k$$

- **REFINEMENT COEFFICIENTS**

SPLINE-BASED 4-pt

$$a_{0,i}^k = -\frac{1}{8\lambda_i^k(\lambda_i^k+1)}$$
$$a_{1,i}^k = \frac{1+3\lambda_i^k+\mu_i^k+4\lambda_i^k\mu_i^k}{\lambda_i^k(\mu_i^k+1)}$$
$$a_{2,i}^k = \frac{1+3\mu_i^k+\lambda_i^k+4\lambda_i^k\mu_i^k}{\mu_i^k(\lambda_i^k+1)}$$
$$a_{3,i}^k = -\frac{1}{8\mu_i^k(1+\mu_i^k)}$$

$$\lambda_i = \mu_i$$

$$=$$

CIRCLE-PRESERVING 4-pt

$$a_{0,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{1,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{2,i}^k = \frac{(2\delta_i^k+1)^2}{8\delta_i^k(1+\delta_i^k)}$$
$$a_{3,i}^k = -\frac{1}{8\delta_i^k(1+\delta_i^k)}$$

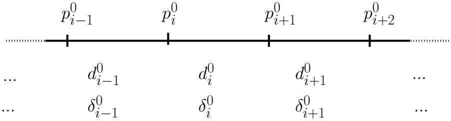
- Define $\lambda_i^k = \frac{d_{i-1}^k}{d_i^k}\,\delta_i^k$ and $\mu_i^k = \frac{d_{i+1}^k}{d_i^k}\,\delta_i^k$
- Then the two subdivision schemes are represented by the same set of coefficients

➥ **NULI4^{++} subdivision**

(Non-Uniform Local Interpolatory 4-pt Subdivision)

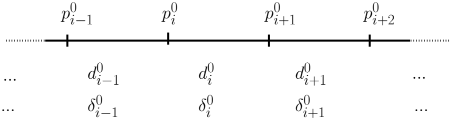
Initial parameters setting

- Two sets of parameters:



Initial parameters setting

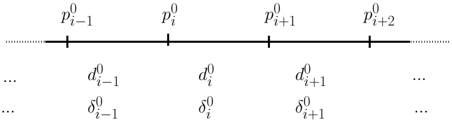
- Two sets of parameters:



- **step1:** $\forall i$, compute $d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$

Initial parameters setting

- Two sets of parameters:



- **step1:** $\forall i$, compute $d^0_i = \|p^0_{i+1} - p^0_i\|_2^{\frac{1}{2}}$

- **step2:** $\forall i$, compute δ^0_i as follows

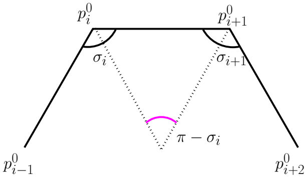
set $\delta^0_i = 1$

if $d^0_{i-1} = d^0_i = d^0_{i+1}$

compute σ_i and σ_{i+1}

if $\sigma_i = \sigma_{i+1}$

$\delta^0_i = \cos\left(\frac{\pi - \sigma_i}{2}\right)$



Parameters updating and refinement

For all $k \geq 1$

- **PARAMETERS UPDATING**

if $\delta_i^k \neq 1$ (i.e. $d_{i-1}^k = d_i^k = d_{i+1}^k$)

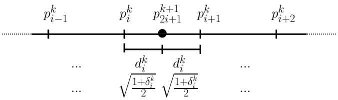
$$d_{2i}^{k+1} = d_{2i+1}^{k+1} = d_i^k$$

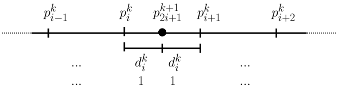
$$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = \sqrt{\frac{1+\delta_i^k}{2}}$$

else $(\delta_i^k = 1)$

$$d_{2i}^{k+1} = d_{2i+1}^{k+1} = d_i^k$$

$$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = 1$$





Parameters updating and refinement

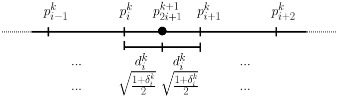
For all $k \geq 1$

- **PARAMETERS UPDATING**

  if $\delta_i^k \neq 1$ (i.e. $d_{i-1}^k = d_i^k = d_{i+1}^k$)

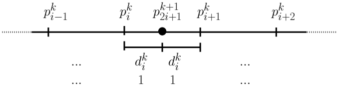
  $$d_{2i}^{k+1} = d_{2i+1}^{k+1} = d_i^k$$

  $$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = \sqrt{\frac{1+\delta_i^k}{2}}$$



  else $(\delta_i^k = 1)$

  $$d_{2i}^{k+1} = d_{2i+1}^{k+1} = d_i^k$$
  $$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = 1$$



- **COEFFICIENTS COMPUTATION**

  $$\lambda_i^k = \frac{d_{i-1}^k}{d_i^k}\delta_i^k, \quad \mu_i^k = \frac{d_{i+1}^k}{d_i^k}\delta_i^k \quad \Longrightarrow$$

REFINEMENT COEFFICIENTS

$$a_{0,i}^k = -\frac{1}{8\lambda_i^k(\lambda_i^k+1)}$$

$$a_{1,i}^k = \frac{1+3\lambda_i^k+\mu_i^k+4\lambda_i^k\mu_i^k}{\lambda_i^k(\mu_i^k+1)}$$

$$a_{2,i}^k = \frac{1+3\mu_i^k+\lambda_i^k+4\lambda_i^k\mu_i^k}{\mu_i^k(\lambda_i^k+1)}$$

$$a_{3,i}^k = -\frac{1}{8\mu_i^k(1+\mu_i^k)}$$

Parameters updating and refinement

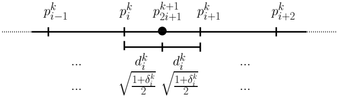
For all $k \geq 1$

- **PARAMETERS UPDATING**

  if $\delta_i^k \neq 1$ (i.e. $d_{i-1}^k = d_i^k = d_{i+1}^k$)

  $$d_{2i}^{k+1} = d_{2i+1}^{k+1} = d_i^k$$

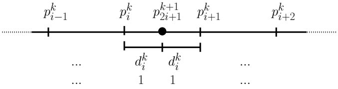
  $$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = \sqrt{\frac{1+\delta_i^k}{2}}$$



  else $(\delta_i^k = 1)$

  $$d_{2i}^{k+1} = d_{2i+1}^{k+1} = d_i^k$$

  $$\delta_{2i}^{k+1} = \delta_{2i+1}^{k+1} = 1$$



- **COEFFICIENTS COMPUTATION**

  $$\lambda_i^k = \frac{d_{i-1}^k}{d_i^k}\delta_i^k, \quad \mu_i^k = \frac{d_{i+1}^k}{d_i^k}\delta_i^k \quad \Longrightarrow$$

REFINEMENT COEFFICIENTS

$$a_{0,i}^k = -\frac{1}{8\lambda_i^k(\lambda_i^k+1)}$$

$$a_{1,i}^k = \frac{1+3\lambda_i^k+\mu_i^k+4\lambda_i^k\mu_i^k}{\lambda_i^k(\mu_i^k+1)}$$

$$a_{2,i}^k = \frac{1+3\mu_i^k+\lambda_i^k+4\lambda_i^k\mu_i^k}{\mu_i^k(\lambda_i^k+1)}$$

$$a_{3,i}^k = -\frac{1}{8\mu_i^k(1+\mu_i^k)}$$

- **REFINEMENT**

  $$p_{2i}^{k+1} = p_i^k$$

  $$p_{2i+1}^{k+1} = a_{0,i}^k\, p_{i-1}^k + a_{1,i}^k\, p_i^k + a_{2,i}^k\, p_{i+1}^k + a_{3,i}^k\, p_{i+2}^k$$

Properties of the NULI4^{++} subdivision scheme

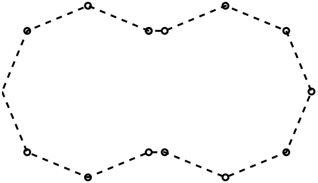
- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples

Properties of the NULI4^{++} subdivision scheme

- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples
- whenever a circular section is present, it is automatically reproduced; otherwise, the scheme is non-uniform and centripetally parameterized.

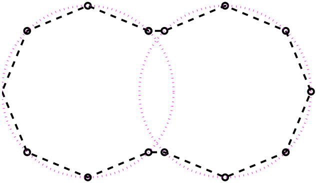
Properties of the NULI4^{++} subdivision scheme

- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples
- whenever a circular section is present, it is automatically reproduced; otherwise, the scheme is non-uniform and centripetally parameterized.



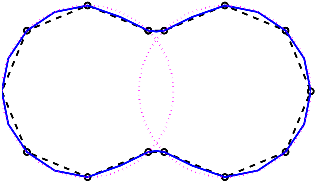
Properties of the NULI4^{++} subdivision scheme

- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples
- whenever a circular section is present, it is automatically reproduced; otherwise, the scheme is non-uniform and centripetally parameterized.



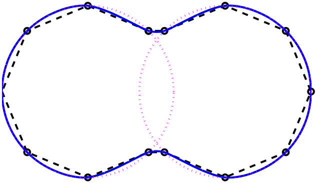
Properties of the NULI4^{++} subdivision scheme

- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples
- whenever a circular section is present, it is automatically reproduced; otherwise, the scheme is non-uniform and centripetally parameterized.



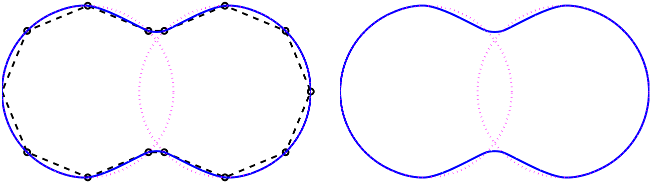
Properties of the NULI4^{++} subdivision scheme

- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples
- whenever a circular section is present, it is automatically reproduced; otherwise, the scheme is non-uniform and centripetally parameterized.



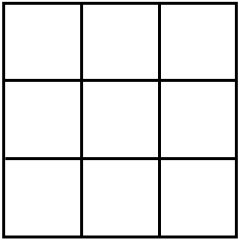
Properties of the NULI4^{++} subdivision scheme

- non-uniform parameterization
- local support [-3,3]
- C^1 smoothness
- polynomials reproduction:
  - up to deg-2 starting from *non-equispaced* samples
  - up to deg-3 starting from *equispaced* samples
- whenever a circular section is present, it is automatically reproduced; otherwise, the scheme is non-uniform and centripetally parameterized.



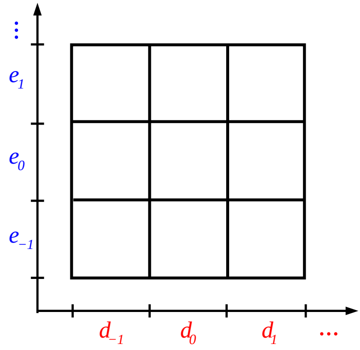
From curve subdivision to surface subdivision

- Quadrilateral meshes



From curve subdivision to surface subdivision

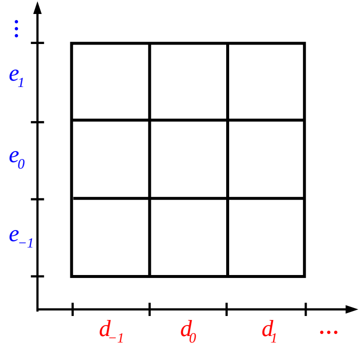
- Quadrilateral meshes
- Uniform parameterization



From curve subdivision to surface subdivision

- Quadrilateral meshes
- Uniform parameterization

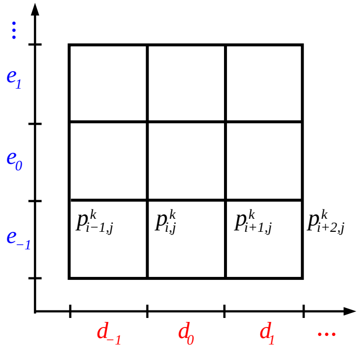
} refinement equations are the **tensor-product** of the corresponding curve subdivision scheme



From curve subdivision to surface subdivision

- Quadrilateral meshes
- Uniform parameterization

} refinement equations are the **tensor-product** of the corresponding curve subdivision scheme

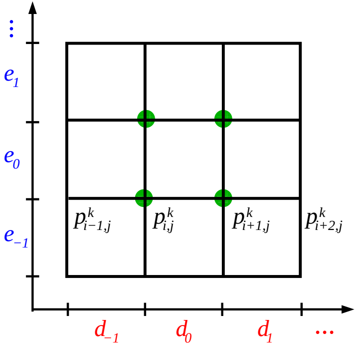


- Given the initial net of points $\{p^0_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$, we get:

From curve subdivision to surface subdivision

- Quadrilateral meshes
- Uniform parameterization

} refinement equations are the **tensor-product** of the corresponding curve subdivision scheme

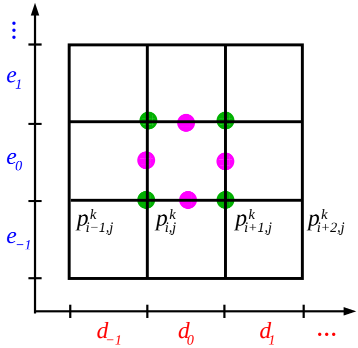


- Given the initial net of points $\{p^0_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$, we get:
  - $p^{k+1}_{2i,2j} = p^k_{i,j}$ **VERTEX POINTS**

From curve subdivision to surface subdivision

- Quadrilateral meshes
- Uniform parameterization

} refinement equations are the **tensor-product** of the corresponding curve subdivision scheme

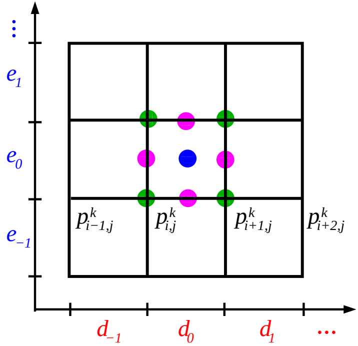


- Given the initial net of points $\{p^0_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$, we get:
  - $p^{k+1}_{2i,2j} = p^k_{i,j}$ **VERTEX POINTS**
  - $p^{k+1}_{2i+1,2j} = \sum_h a^k_h \, p^k_{i-h,j}$ **EDGE POINTS**
    $p^{k+1}_{2i,2j+1} = \sum_h b^k_h \, p^k_{i,j-h}$

From curve subdivision to surface subdivision

- Quadrilateral meshes
- Uniform parameterization

} refinement equations are the **tensor-product** of the corresponding curve subdivision scheme

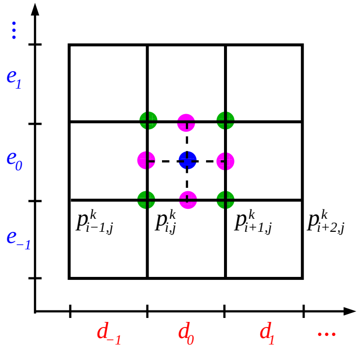


- Given the initial net of points $\{p^0_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$, we get:
  - $p^{k+1}_{2i,2j} = p^k_{i,j}$ **VERTEX POINTS**
  - $p^{k+1}_{2i+1,2j} = \sum_h a^k_h \, p^k_{i-h,j}$ **EDGE POINTS**
    $p^{k+1}_{2i,2j+1} = \sum_h b^k_h \, p^k_{i,j-h}$
  - $p^{k+1}_{2i+1,2j+1} = \sum_{h,\ell} a^k_h b^k_\ell \, p^k_{i-h,j-\ell}$ **FACE POINTS**

From curve subdivision to surface subdivision

- Quadrilateral meshes
- Uniform parameterization

} refinement equations are the **tensor-product** of the corresponding curve subdivision scheme



- Given the initial net of points $\{p^0_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$, we get:

  - $p^{k+1}_{2i,2j} = p^k_{i,j}$ **VERTEX POINTS**

  - $p^{k+1}_{2i+1,2j} = \sum_h a^k_h \, p^k_{i-h,j}$ **EDGE POINTS**

    $p^{k+1}_{2i,2j+1} = \sum_h b^k_h \, p^k_{i,j-h}$

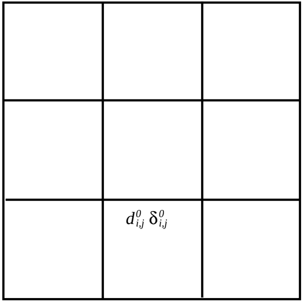
  - $p^{k+1}_{2i+1,2j+1} = \sum_{h,\ell} a^k_h b^k_\ell \, p^k_{i-h,j-\ell}$ **FACE POINTS**

NULISS^{++} refinement algorithm

- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh



NULISS^{++} refinement algorithm

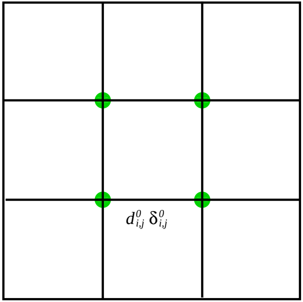
- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh

- **For each refinement iteration $k \geq 1$:**

1. Retain each vertex point

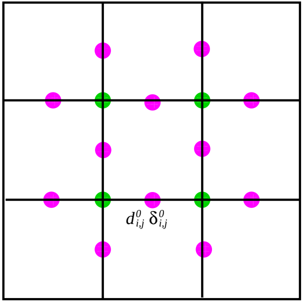


NULISS^{++} refinement algorithm

- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh



- **For each refinement iteration $k \geq 1$:**

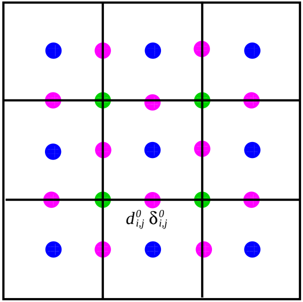
1. Retain each vertex point
2. Compute a new edge point for each edge

NULISS^{++} refinement algorithm

- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh



- **For each refinement iteration $k \geq 1$:**

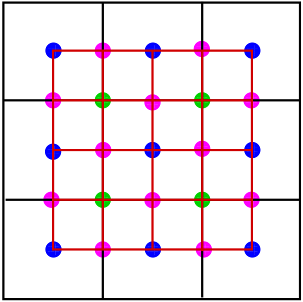
1. Retain each vertex point
2. Compute a new edge point for each edge
3. Compute a new face point for each face

NULISS^{++} refinement algorithm

- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh



- **For each refinement iteration $k \geq 1$:**

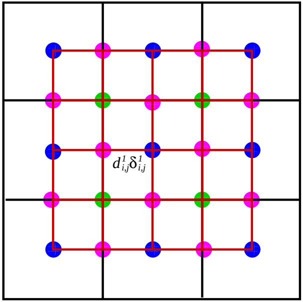
1. Retain each vertex point
2. Compute a new edge point for each edge
3. Compute a new face point for each face
4. Create the new mesh

NULISS^{++} refinement algorithm

- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh



- **For each refinement iteration $k \geq 1$:**

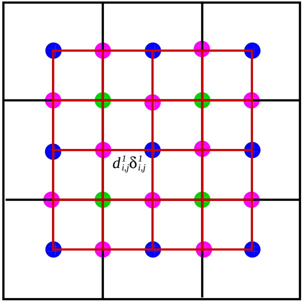
1. Retain each vertex point
2. Compute a new edge point for each edge
3. Compute a new face point for each face
4. Create the new mesh
5. Update the edge parameters and assign them to the new edges

NULISS^{++} refinement algorithm

- A couple of parameters

$$\boldsymbol{d_j} = (d^0_{i,j}, \delta^0_{i,j})$$

is associated to each edge of the initial mesh

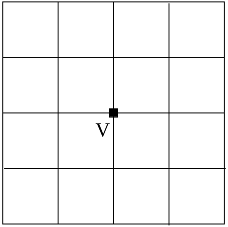


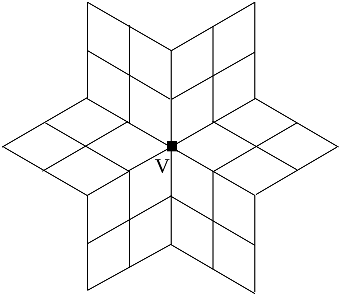
- **For each refinement iteration $k \geq 1$:**

1. Retain each vertex point
2. Compute a new edge point for each edge
3. Compute a new face point for each face
4. Create the new mesh
5. Update the edge parameters and assign them to the new edges

Taxonomy and parameters setting

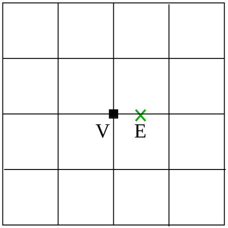
- **REGULAR VERTICES ($n = 4$):** the refinement equations are derived by generalizing the tensor prouct of the curve scheme
- **EXTRAORDINARY VERTICES ($n \neq 4$):** the refinement equations are generalized, s. t. they reduce to the regular case when $n = 4$.





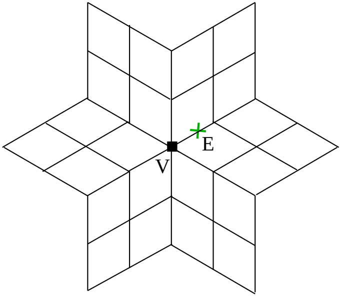
Taxonomy and parameters setting

- **REGULAR VERTICES ($n = 4$):** the refinement equations are derived by generalizing the tensor prouct of the curve scheme
- **EXTRAORDINARY VERTICES ($n \neq 4$):** the refinement equations are generalized, s. t. they reduce to the regular case when $n = 4$.



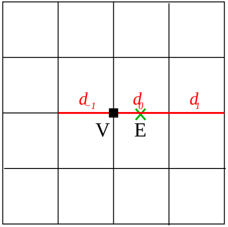
Parameters to compute the refinement coefficients:

- **EDGE POINTS E**: are found on the edges of the mesh



Taxonomy and parameters setting

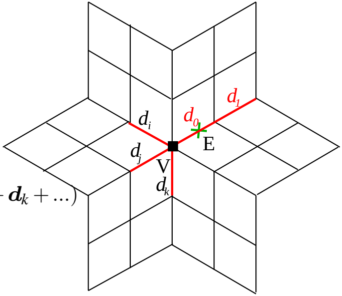
- **REGULAR VERTICES ($n = 4$):** the refinement equations are derived by generalizing the tensor prouct of the curve scheme
- **EXTRAORDINARY VERTICES ($n \neq 4$):** the refinement equations are generalized, s. t. they reduce to the regular case when $n = 4$.



Parameters to compute the refinement coefficients:

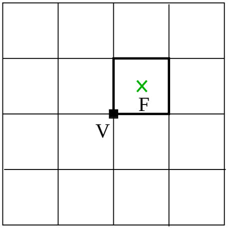
- **EDGE POINTS E**: are found on the edges of the mesh

$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k + \ldots)$$



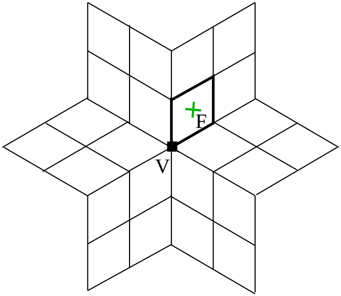
Taxonomy and parameters setting

- **REGULAR VERTICES ($n = 4$):** the refinement equations are derived by generalizing the tensor prouct of the curve scheme
- **EXTRAORDINARY VERTICES ($n \neq 4$):** the refinement equations are generalized, s. t. they reduce to the regular case when $n = 4$.



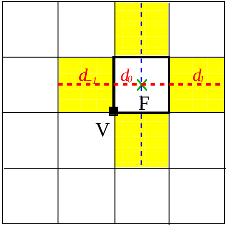
Parameters to compute the refinement coefficients:

- **EDGE POINTS E**: are found on the edges of the mesh
- **FACE POINTS F**: are found by averaging the parameters on opposite edges in the two directions



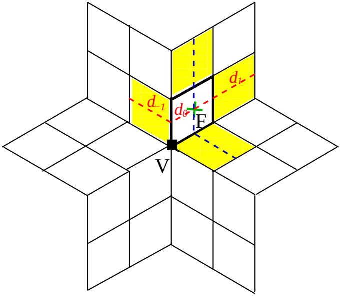
Taxonomy and parameters setting

- **REGULAR VERTICES ($n = 4$):** the refinement equations are derived by generalizing the tensor prouct of the curve scheme
- **EXTRAORDINARY VERTICES ($n \neq 4$):** the refinement equations are generalized, s. t. they reduce to the regular case when $n = 4$.

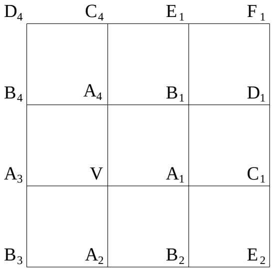


Parameters to compute the refinement coefficients:

- **EDGE POINTS E**: are found on the edges of the mesh
- **FACE POINTS F**: are found by averaging the parameters on opposite edges in the two directions

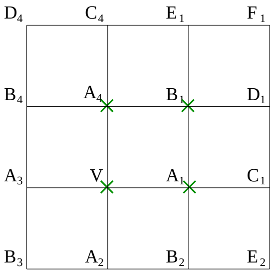


Regular vertices



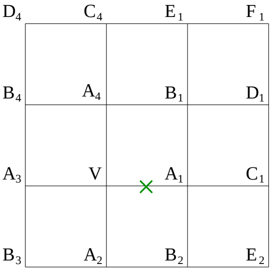
Regular vertices

- **VERTEX POINTS** are interpolated



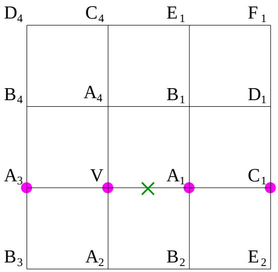
Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**



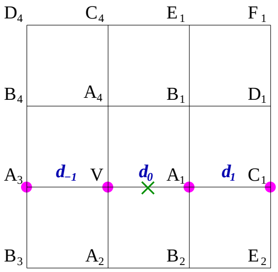
Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**



Regular vertices

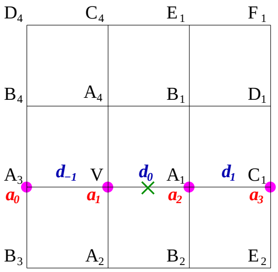
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**



Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

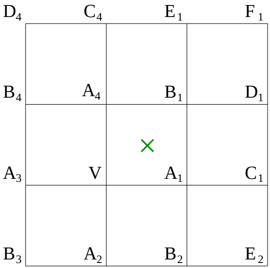


Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

  $$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

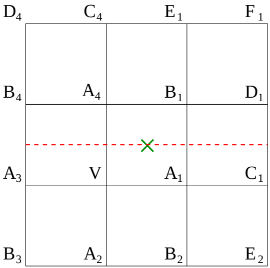


Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

  $$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

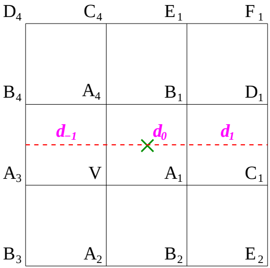


Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

  $$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

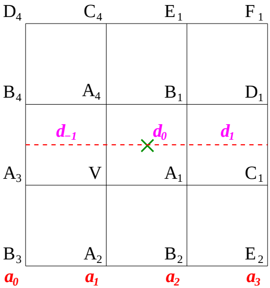
- **FACE POINTS**



Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

  $\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$
- **FACE POINTS**

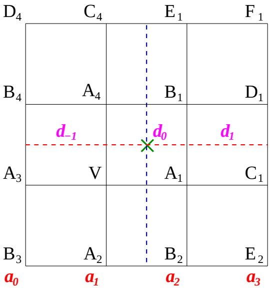


Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

  $$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

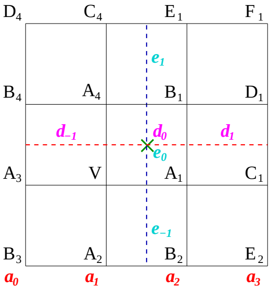


Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**



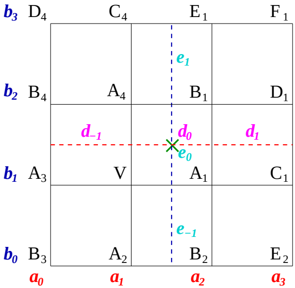
Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

$$\begin{aligned}\mathbf{F} &= b_0(a_0 B_3 + a_1 A_2 + a_2 B_2 + a_3 E_2)\\ &+ b_1(a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1)\\ &+ b_2(a_0 B_4 + a_1 A_4 + a_2 B_1 + a_3 D_1)\\ &+ b_3(a_0 D_4 + a_1 C_4 + a_2 E_1 + a_3 F_1)\end{aligned}$$



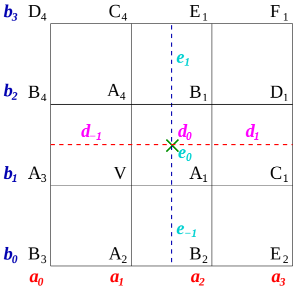
Regular vertices

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\mathbf{E} = a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1$$

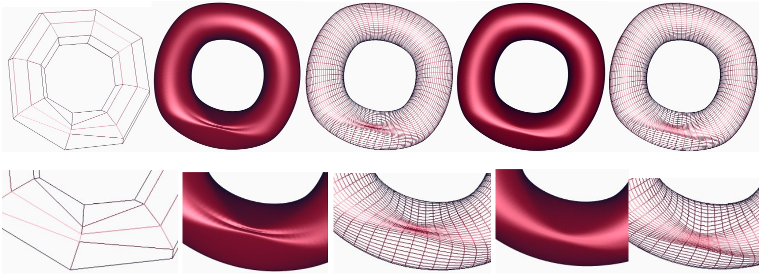
- **FACE POINTS**

$$\begin{aligned}\mathbf{F} &= b_0(a_0 B_3 + a_1 A_2 + a_2 B_2 + a_3 E_2) \\ &+ b_1(a_0 A_3 + a_1 V + a_2 A_1 + a_3 C_1) \\ &+ b_2(a_0 B_4 + a_1 A_4 + a_2 B_1 + a_3 D_1) \\ &+ b_3(a_0 D_4 + a_1 C_4 + a_2 E_1 + a_3 F_1)\end{aligned}$$



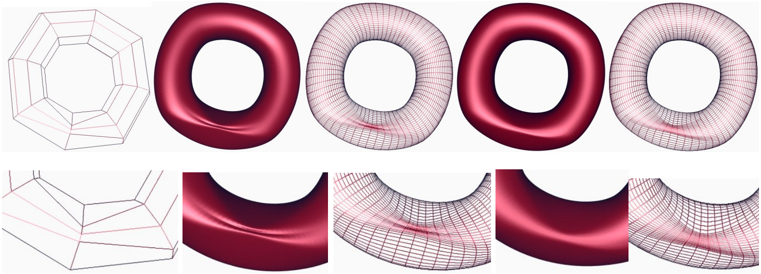
- When the parameters d_i are set compatibly with the tensor-product structure, the surface scheme NULISS^{++} coincides with the tensor-product of the NULI4^{++} curve scheme.

Regular case, non tensor-product

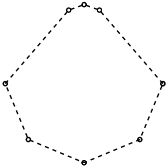


INITIAL MESH **SPLINE SURFACE** **NULISS^{++}**

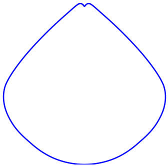
Regular case, non tensor-product



INITIAL MESH

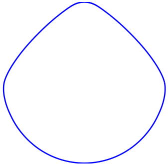


SPLINE SURFACE



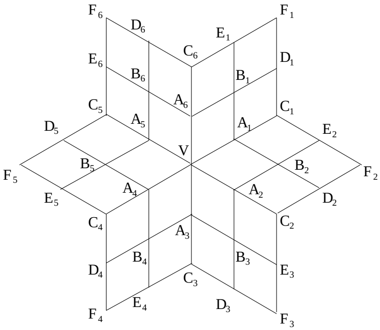
Each section curve is parameterized by the mean of the parameterizations in the corresponding direction

NULISS^{++}



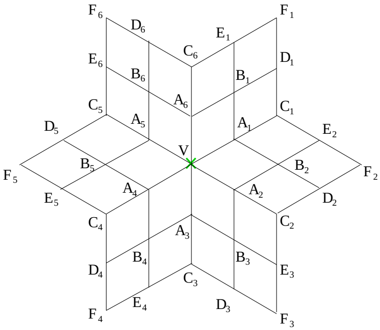
Each section curve maintains its own (centripetal) parameterization

Extraordinary vertices, $n \geqslant 5$



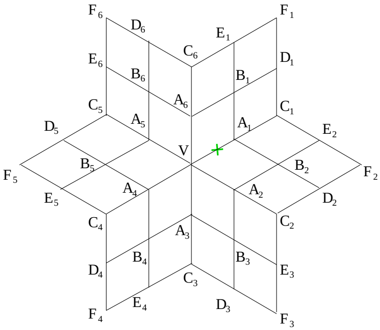
Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated



Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

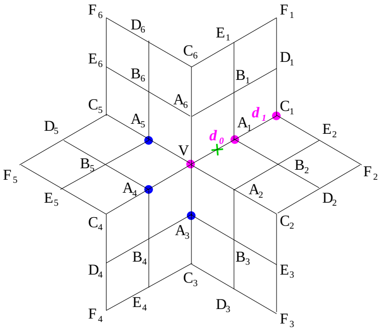


Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$d_{-1} = ?$$

$$\mathbf{E} = ? + ?\, V + a_2 A_1 + a_3 C_1$$

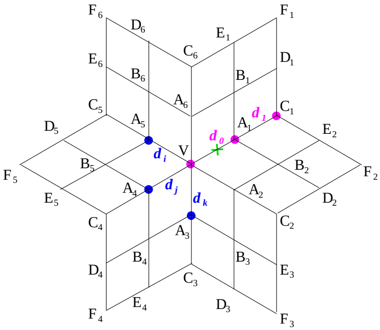


Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$d_{-1} = \frac{1}{n-3}(d_i + d_j + d_k)$$

$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$



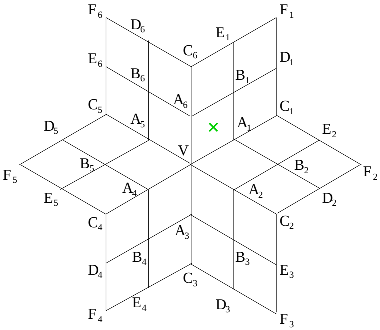
Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**



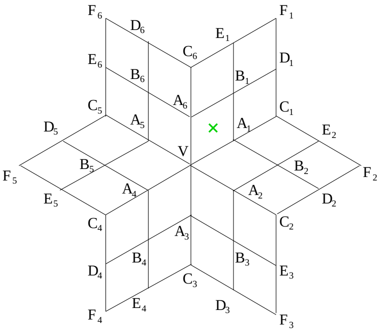
Extraordinary vertices, $n \geqslant 5$

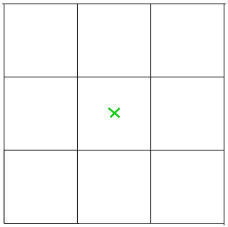
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**





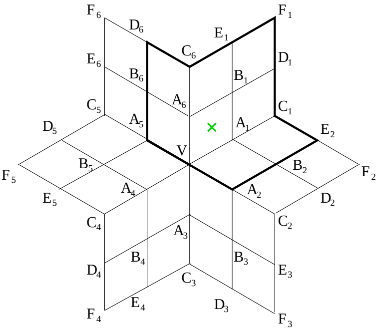
Extraordinary vertices, $n \geqslant 5$

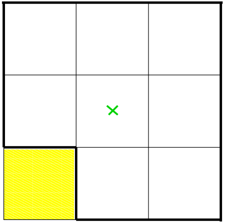
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**





Extraordinary vertices, $n \geqslant 5$

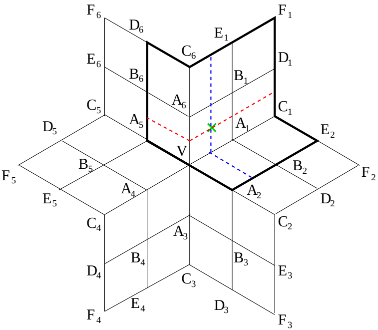
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

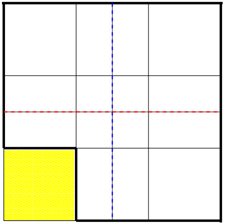
$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

  parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions





Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated

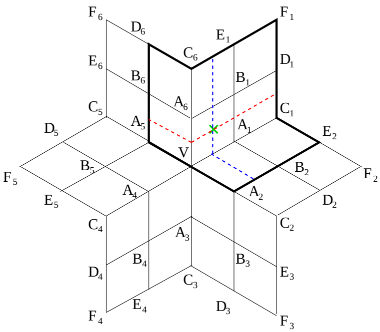
- **EDGE POINTS**

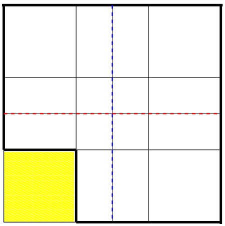
$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

  parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions





Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

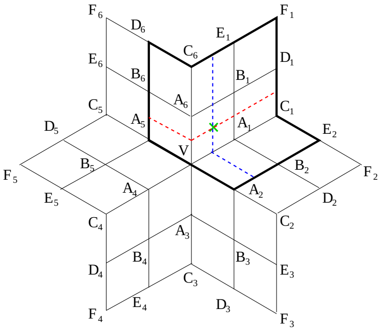
$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

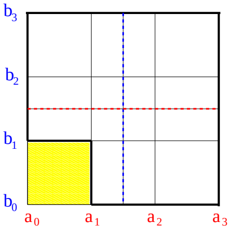
$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\begin{aligned}\mathbf{F} = \; & b_0(a_0 ? + a_1 ? + a_2 B_2 + a_3 E_2) \\ + \; & b_1(a_0 ? + a_1 V + a_2 A_1 + a_3 C_1) \\ + \; & b_2(a_0 B_6 + a_1 A_6 + a_2 B_1 + a_3 D_1) \\ + \; & b_3(a_0 D_6 + a_1 C_6 + a_2 E_1 + a_3 F_1)\end{aligned}$$





Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

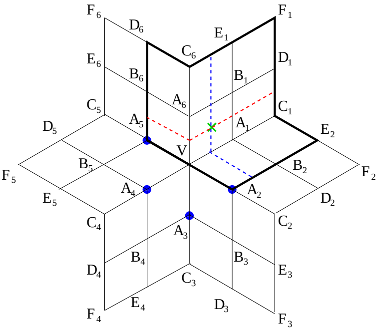
$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

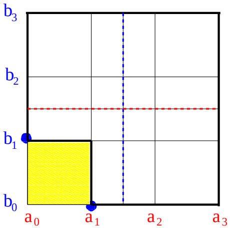
$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

  parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\begin{aligned}\mathbf{F} = \; & b_0(a_0 ? + a_1 ? + a_2 B_2 + a_3 E_2) \\ + \; & b_1(a_0 ? + a_1 V + a_2 A_1 + a_3 C_1) \\ + \; & b_2(a_0 B_6 + a_1 A_6 + a_2 B_1 + a_3 D_1) \\ + \; & b_3(a_0 D_6 + a_1 C_6 + a_2 E_1 + a_3 F_1)\end{aligned}$$





Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

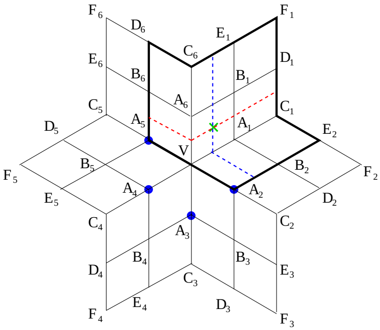
$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

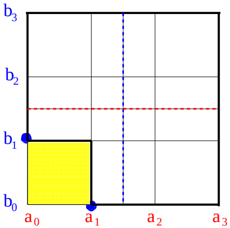
$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\begin{aligned} \mathbf{F} = \; & b_0(a_0 ? + a_1 \frac{1}{n-3} \sum_{i=3}^{n-1} A_i + a_2 B_2 + a_3 E_2) \\ + \; & b_1(a_0 \frac{1}{n-3} \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1) \\ + \; & b_2(a_0 B_6 + a_1 A_6 + a_2 B_1 + a_3 D_1) \\ + \; & b_3(a_0 D_6 + a_1 C_6 + a_2 E_1 + a_3 F_1) \end{aligned}$$





Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

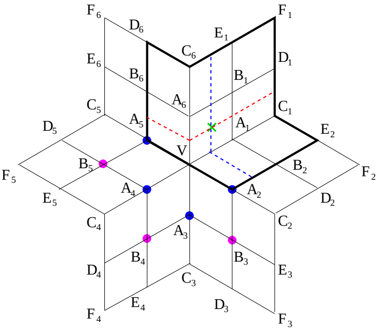
$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

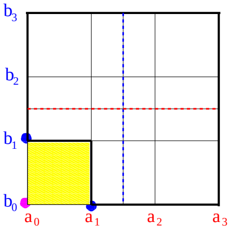
$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\begin{aligned}\mathbf{F} = \; & b_0(a_0 ? + a_1 \frac{1}{n-3} \sum_{i=3}^{n-1} A_i + a_2 B_2 + a_3 E_2) \\ + \; & b_1(a_0 \frac{1}{n-3} \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1) \\ + \; & b_2(a_0 B_6 + a_1 A_6 + a_2 B_1 + a_3 D_1) \\ + \; & b_3(a_0 D_6 + a_1 C_6 + a_2 E_1 + a_3 F_1)\end{aligned}$$



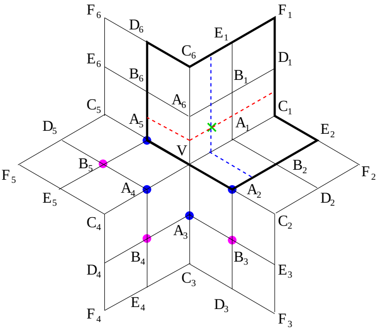


Extraordinary vertices, $n \geqslant 5$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{n-3}(\boldsymbol{d}_i + \boldsymbol{d}_j + \boldsymbol{d}_k)$$

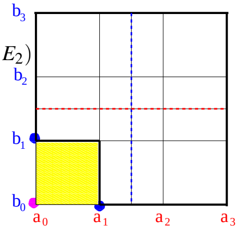
$$\mathbf{E} = a_0 \sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1$$



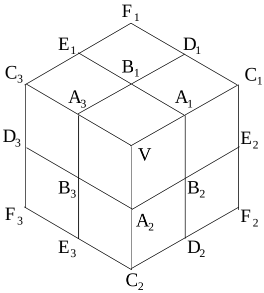
- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\mathbf{F} = b_0\left(a_0 \frac{1}{n-3}\sum_{i=3}^{n-1} B_i + a_1 \frac{1}{n-3}\sum_{i=2}^{n-2} A_i + a_2 B_2 + a_3 E_2\right)$$
$$+ b_1\left(a_0 \frac{1}{n-3}\sum_{i=3}^{n-1} A_i + a_1 V + a_2 A_1 + a_3 C_1\right)$$
$$+ b_2(a_0 B_6 + a_1 A_6 + a_2 B_1 + a_3 D_1)$$
$$+ b_3(a_0 D_6 + a_1 C_6 + a_2 E_1 + a_3 F_1)$$

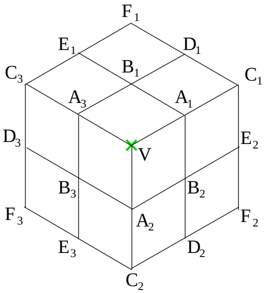


Extraordinary vertices, $n = 3$



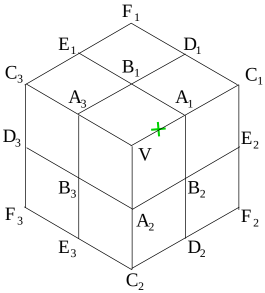
Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated



Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

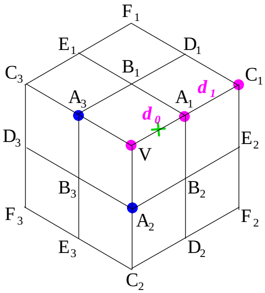


Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = ?$$

$$\mathbf{E} = ? + ?\ V + a_2 A_1 + a_3 C_1$$

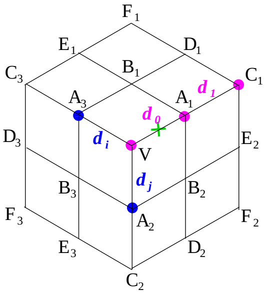


Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$



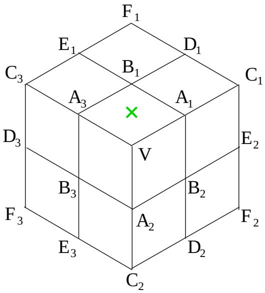
Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$d_{-1} = \frac{1}{2}(d_i + d_j)$$

$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**



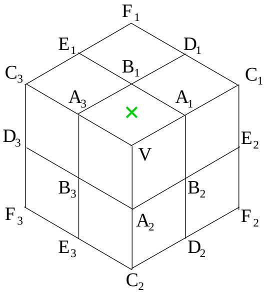
Extraordinary vertices, $n = 3$

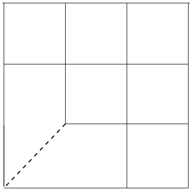
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$d_{-1} = \frac{1}{2}(d_i + d_j)$$

$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**





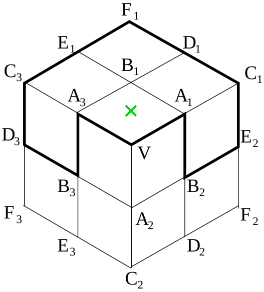
Extraordinary vertices, $n = 3$

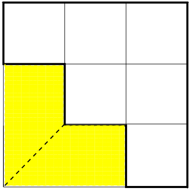
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**





Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated

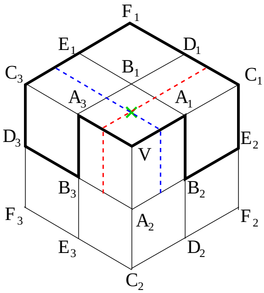
- **EDGE POINTS**

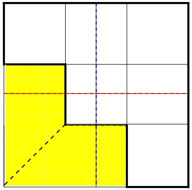
$$d_{-1} = \frac{1}{2}(d_i + d_j)$$

$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

  parameters d_i are computed by averaging edge parameters in the two directions





Extraordinary vertices, $n = 3$

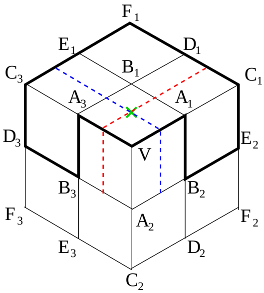
- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

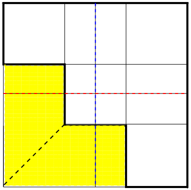
$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions





Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

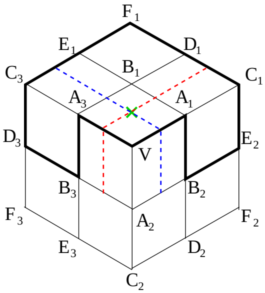
$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

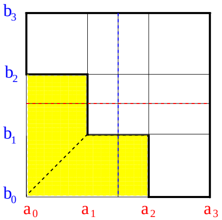
$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\begin{aligned}\mathbf{F} = \; & b_0(a_0 ? + a_1 ? + a_2 B_2 + a_3 E_2) \\ + \; & b_1(a_0 ? + a_1 V + a_2 A_1 + a_3 C_1) \\ + \; & b_2(a_0 B_3 + a_1 A_3 + a_2 B_1 + a_3 D_1) \\ + \; & b_3(a_0 D_3 + a_1 C_3 + a_2 E_1 + a_3 F_1)\end{aligned}$$





Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

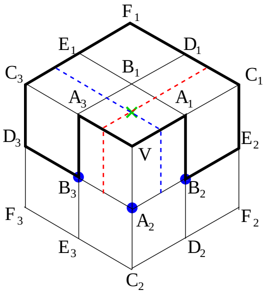
$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

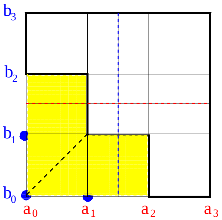
$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\mathbf{F} = b_0(a_0 ? + a_1 ? + a_2 B_2 + a_3 E_2) + b_1(a_0 ? + a_1 V + a_2 A_1 + a_3 C_1) + b_2(a_0 B_3 + a_1 A_3 + a_2 B_1 + a_3 D_1) + b_3(a_0 D_3 + a_1 C_3 + a_2 E_1 + a_3 F_1)$$





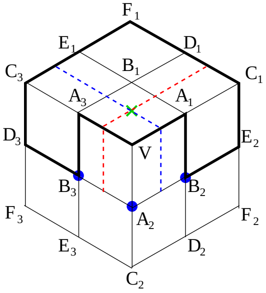
Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated

- **EDGE POINTS**

$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

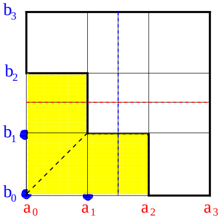
$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$



- **FACE POINTS**

  parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\begin{aligned}\mathbf{F} = {} & b_0(a_0 ? + a_1 \frac{1}{2}(A_2 + B_2) + a_2 B_2 + a_3 E_2) \\ & + b_1(a_0 \frac{1}{2}(A_2 + B_3) + a_1 V + a_2 A_1 + a_3 C_1) \\ & + b_2(a_0 B_3 + a_1 A_3 + a_2 B_1 + a_3 D_1) \\ & + b_3(a_0 D_3 + a_1 C_3 + a_2 E_1 + a_3 F_1)\end{aligned}$$



Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated
- **EDGE POINTS**

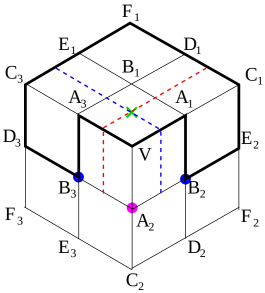
$$d_{-1} = \frac{1}{2}(d_i + d_j)$$

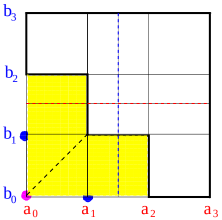
$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters d_i are computed by averaging edge parameters in the two directions

$$\mathbf{F} = b_0(a_0 ? + a_1 \frac{1}{2}(A_2 + B_2) + a_2 B_2 + a_3 E_2)$$
$$+ b_1(a_0 \frac{1}{2}(A_2 + B_3) + a_1 V + a_2 A_1 + a_3 C_1)$$
$$+ b_2(a_0 B_3 + a_1 A_3 + a_2 B_1 + a_3 D_1)$$
$$+ b_3(a_0 D_3 + a_1 C_3 + a_2 E_1 + a_3 F_1)$$





Extraordinary vertices, $n = 3$

- **VERTEX POINTS** are interpolated

- **EDGE POINTS**

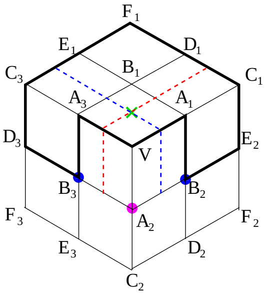
$$\boldsymbol{d}_{-1} = \frac{1}{2}(\boldsymbol{d}_i + \boldsymbol{d}_j)$$

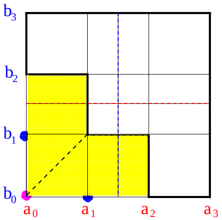
$$\mathbf{E} = a_0 \frac{1}{2}(A_2 + A_3) + a_1 V + a_2 A_1 + a_3 C_1$$

- **FACE POINTS**

parameters $\boldsymbol{d}_i$ are computed by averaging edge parameters in the two directions

$$\mathbf{F} = b_0(a_0\, A_2 + a_1 \frac{1}{2}(A_2 + B_2) + a_2 B_2 + a_3 E_2) + b_1(a_0 \frac{1}{2}(A_2 + B_3) + a_1 V + a_2 A_1 + a_3 C_1) + b_2(a_0 B_3 + a_1 A_3 + a_2 B_1 + a_3 D_1) + b_3(a_0 D_3 + a_1 C_3 + a_2 E_1 + a_3 F_1)$$





Some remarks

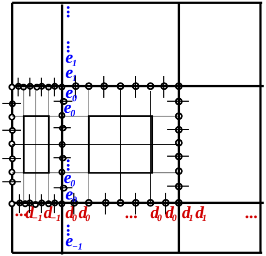
- the NULISS^{++} scheme is non-stationary and non-uniform

Some remarks

- the NULISS^{++} scheme is non-stationary and non-uniform
- after a few subdivision steps, the scheme tends to become
  - stationary $\lim_{k \to +\infty} \delta_i^k = 1, \forall i$

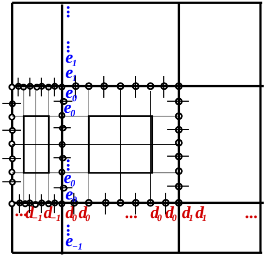
Some remarks

- the NULISS^{++} scheme is non-stationary and non-uniform
- after a few subdivision steps, the scheme tends to become
  - stationary $\lim_{k\to+\infty} \delta_i^k = 1, \; \forall i$
  - uniform



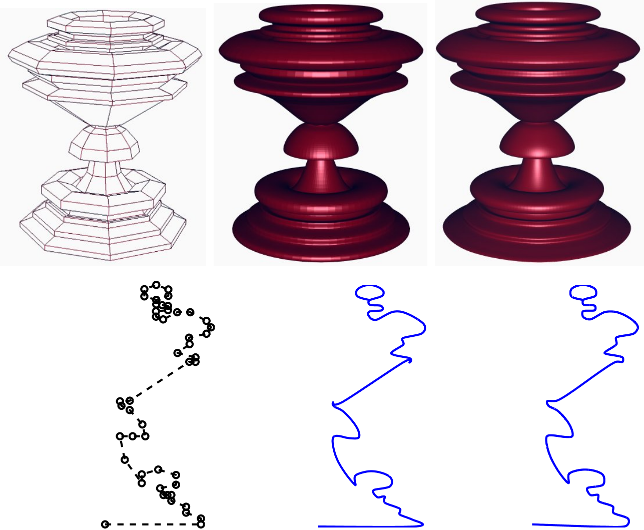
Some remarks

- the NULISS^{++} scheme is non-stationary and non-uniform
- after a few subdivision steps, the scheme tends to become
  - stationary $\lim_{k\to+\infty} \delta_i^k = 1, \; \forall i$
  - uniform



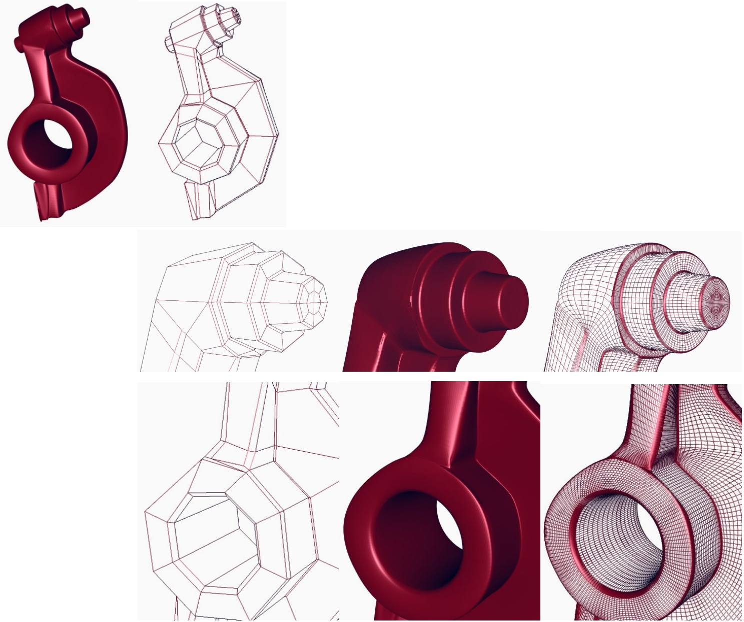
- after a few subdivision steps, evaluation in quadrilateral, tensor-product regions can be performed by the corresponding spline basis

Examples: uniform and non-uniform subdivision

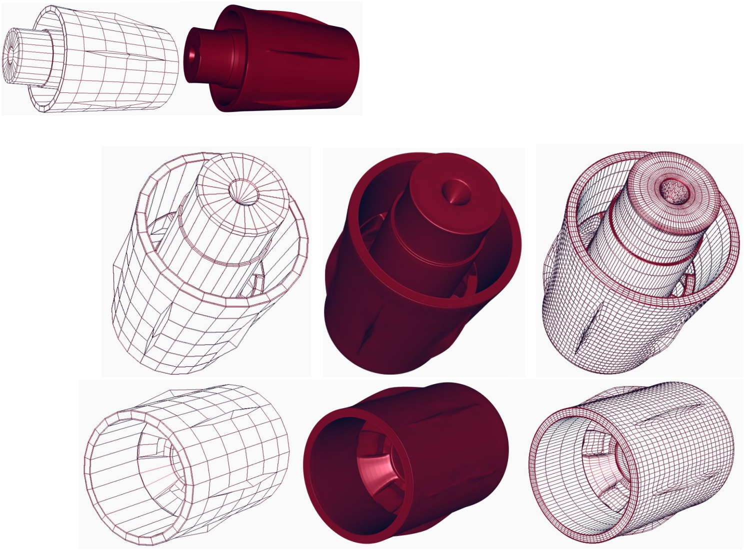


initial mesh/polyline — uniform subdivision — non-uniform subdivision

Examples: extraordinary vertices and circular features



Examples: extraordinary vertices and circular features



Conclusions and future work

Subdivision surfaces & CAD

- ✔ interpolation
- ✔ spline quality (non-uniform parameterization)
- ✔ exact reproduction of circular features
- ✔ polynomial reproduction
- ✔ C^1 smoothness in tensor-product regular regions
- ✔ exact evaluation in tensor-product regular regions

Conclusions and future work

Subdivision surfaces & CAD

- ✔ interpolation
- ✔ spline quality (non-uniform parameterization)
- ✔ exact reproduction of circular features
- ✔ polynomial reproduction
- ✔ C^1 smoothness in tensor-product regular regions
- ✔ exact evaluation in tensor-product regular regions

Future work:

- smoothness in non tensor-product regions
- evaluation at arbitrary parameter values