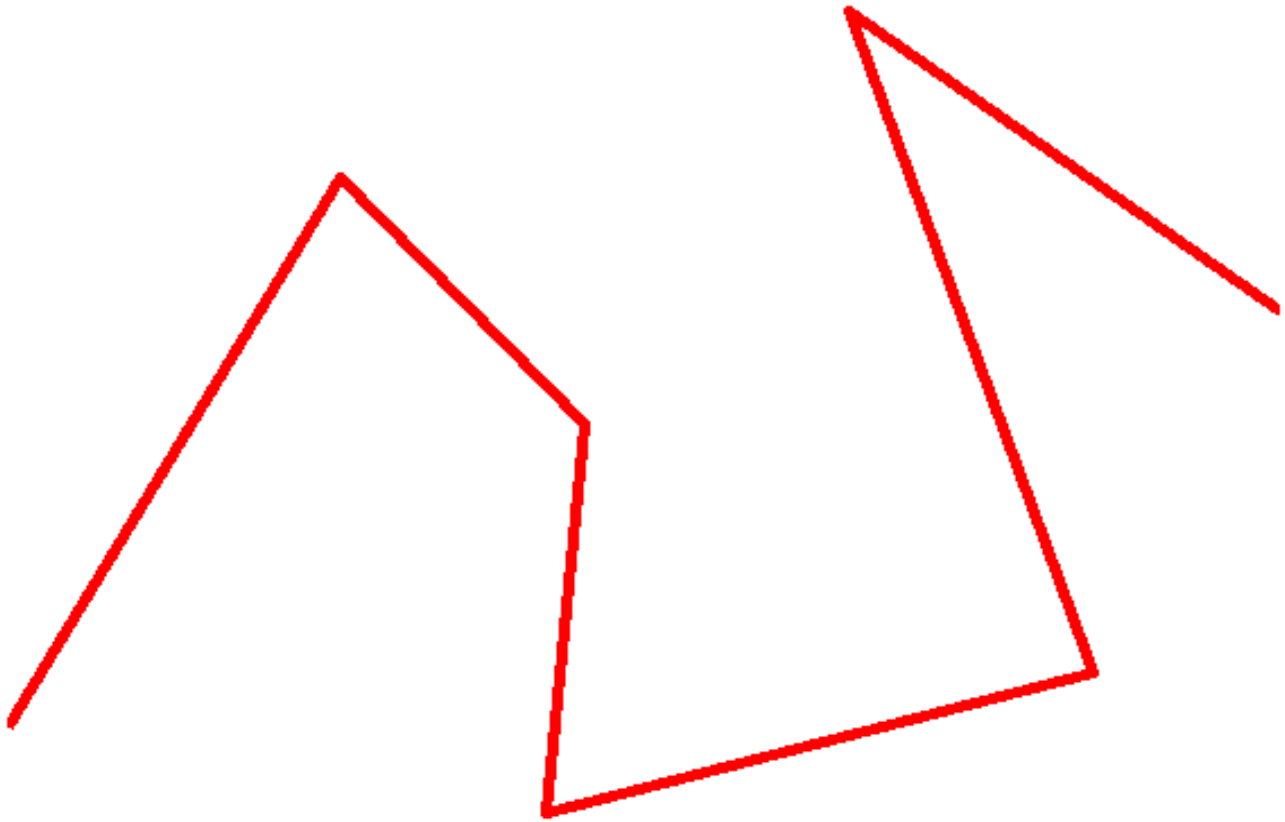
Subdivision Algorithms

(*Abridged* Crash-Course)

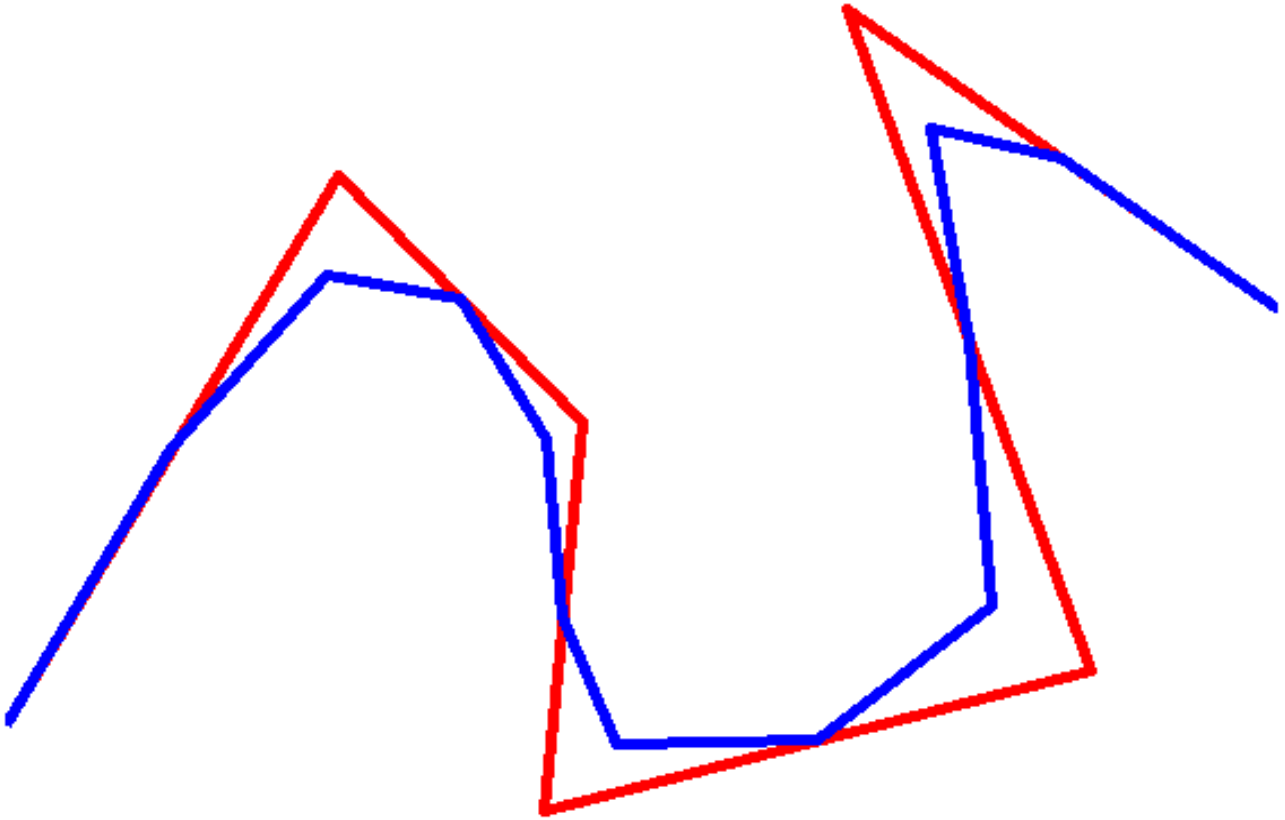
RWTH AACHEN

Informatik VIII
Computer Graphics & Multimedia
Leif P. Kobbelt

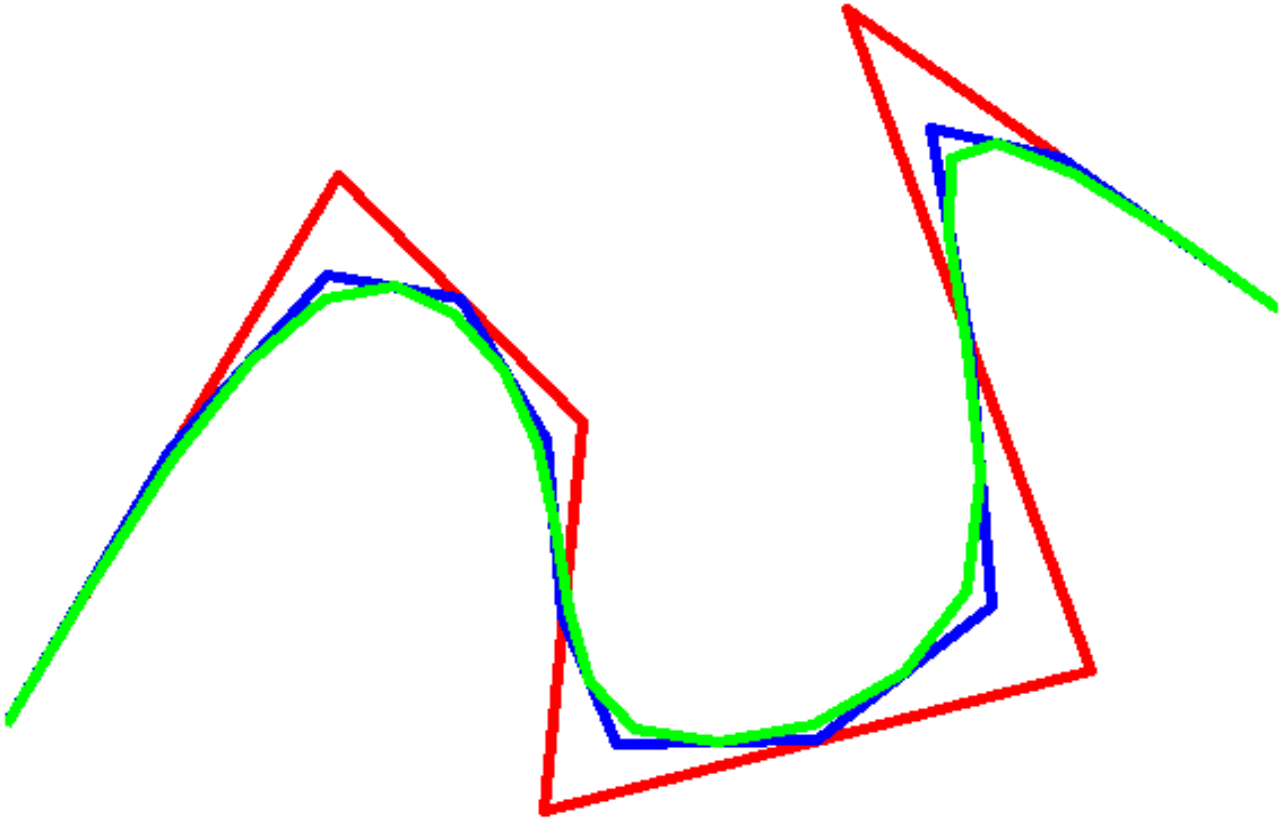
B-Spline Subdivision



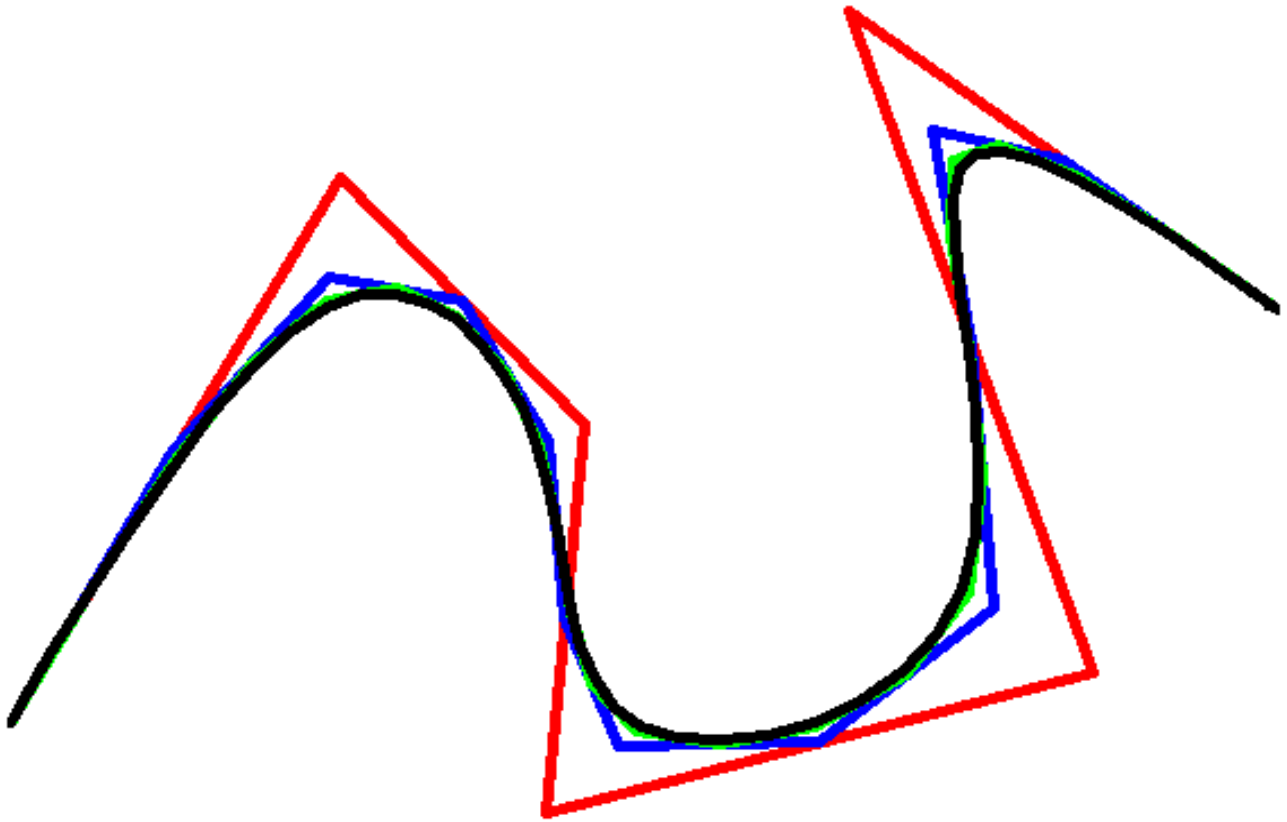
B-Spline Subdivision



B-Spline Subdivision



B-Spline Subdivision



Lane-Riesenfeld

- Double-Operator

  $D : [\ldots, c_i, \ldots] \rightarrow [\ldots, c_i, c_i, \ldots]$

- Average-Operator

  $A : [\ldots, c_i, \ldots] \rightarrow [\ldots, \frac{1}{2}(c_i + c_{i+1}), \ldots]$

Lane-Riesenfeld

- Combination

AD : […, c_i, c_{i+1}, …]

$\rightarrow$ […, c_i, c_i, c_{i+1}, c_{i+1}, …]

$\rightarrow$ […, c_i, $\frac{1}{2}(c_i + c_{i+1})$, c_{i+1}, …]

Lane-Riesenfeld

- Combination

$A^2D : [\ \ldots, c_{i-1}, c_i, c_{i+1}, \ldots\ ]$

$\rightarrow [\ \ldots, c_{i-1}, c_{i-1}, c_i, c_i, c_{i+1}, c_{i+1}, \ldots\ ]$

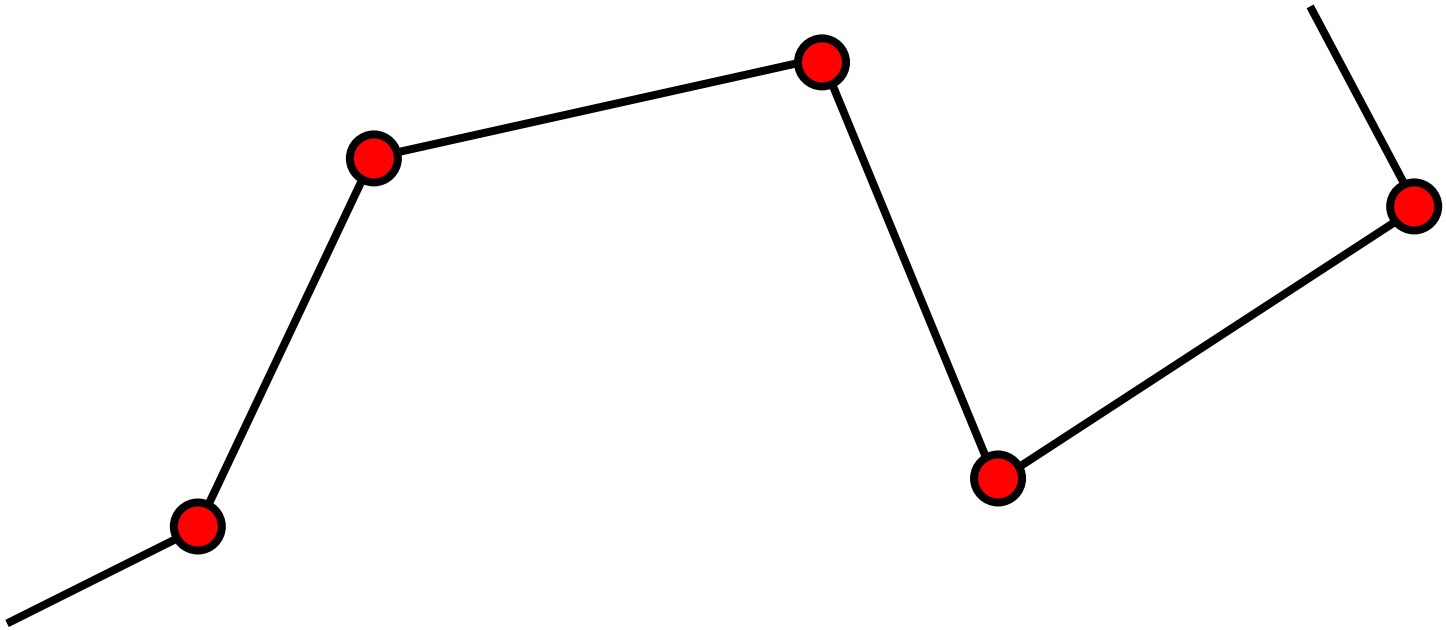
$\rightarrow [\ \ldots, \frac{1}{2}(c_{i-1} + c_i), c_i, \frac{1}{2}(c_i + c_{i+1}), \ldots\ ]$

$\rightarrow [\ \ldots, \frac{1}{4} c_{i-1} + \frac{3}{4} c_i, \frac{3}{4} c_i + \frac{1}{4} c_{i+1}, \ldots\ ]$

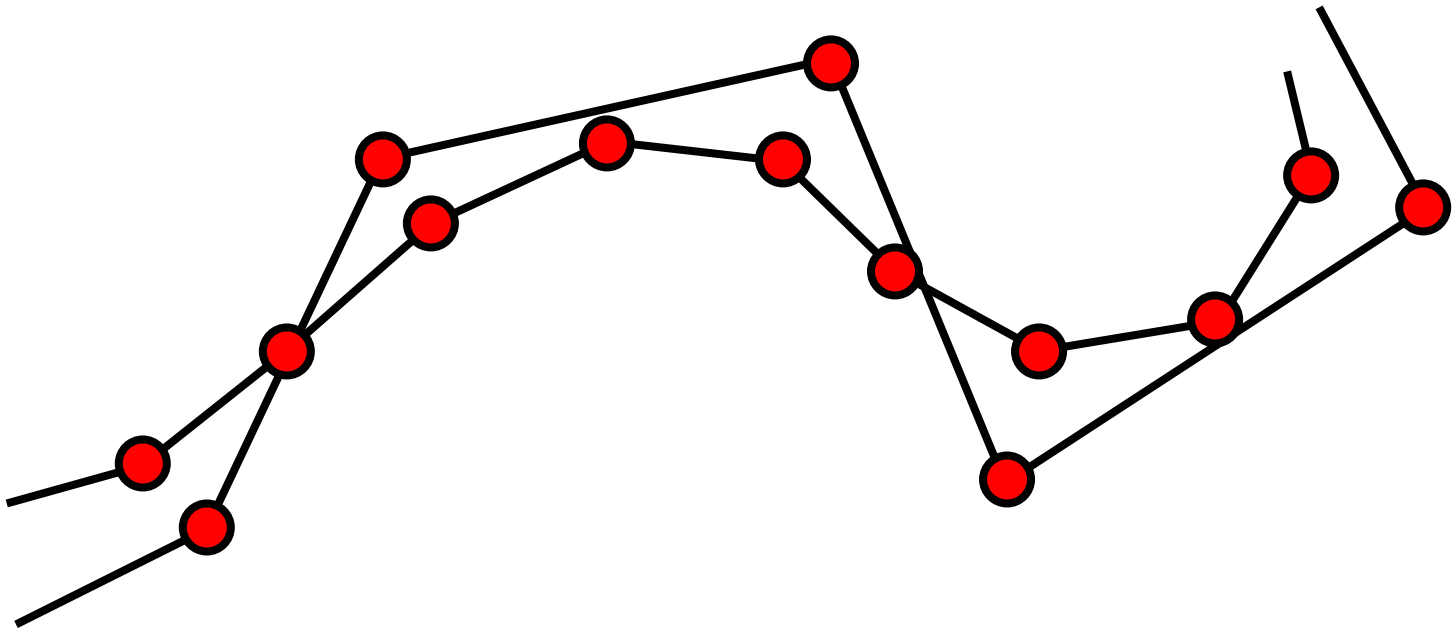
General Structure

- Splitting
  - Primal
  - Dual
- Smoothing
  - Interpolatory
  - Non-Interpolatory

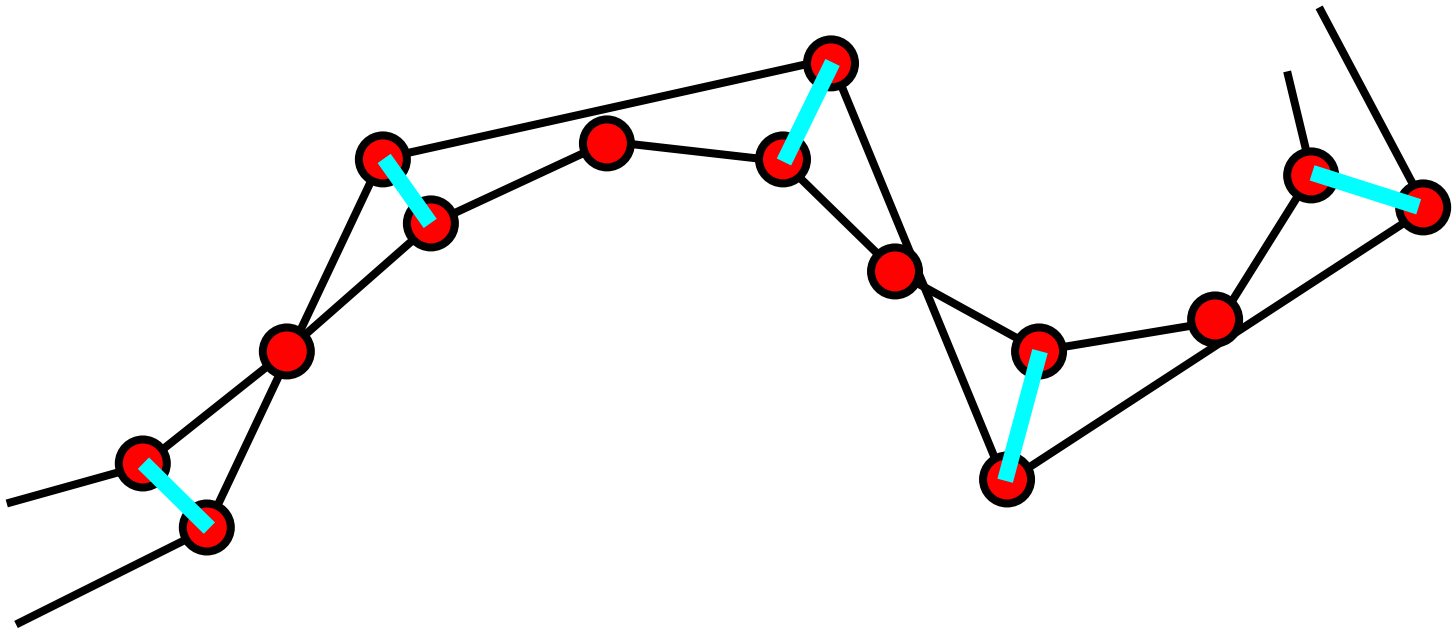
Primal Splitting



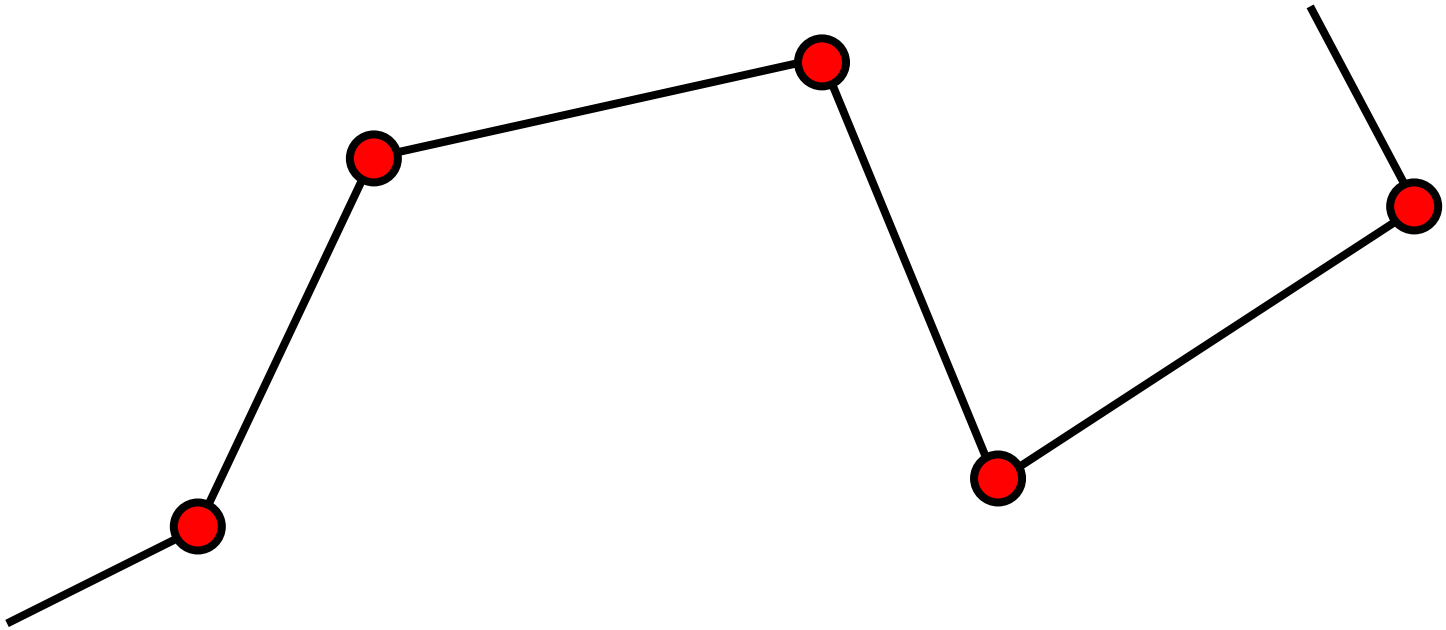
Primal Splitting



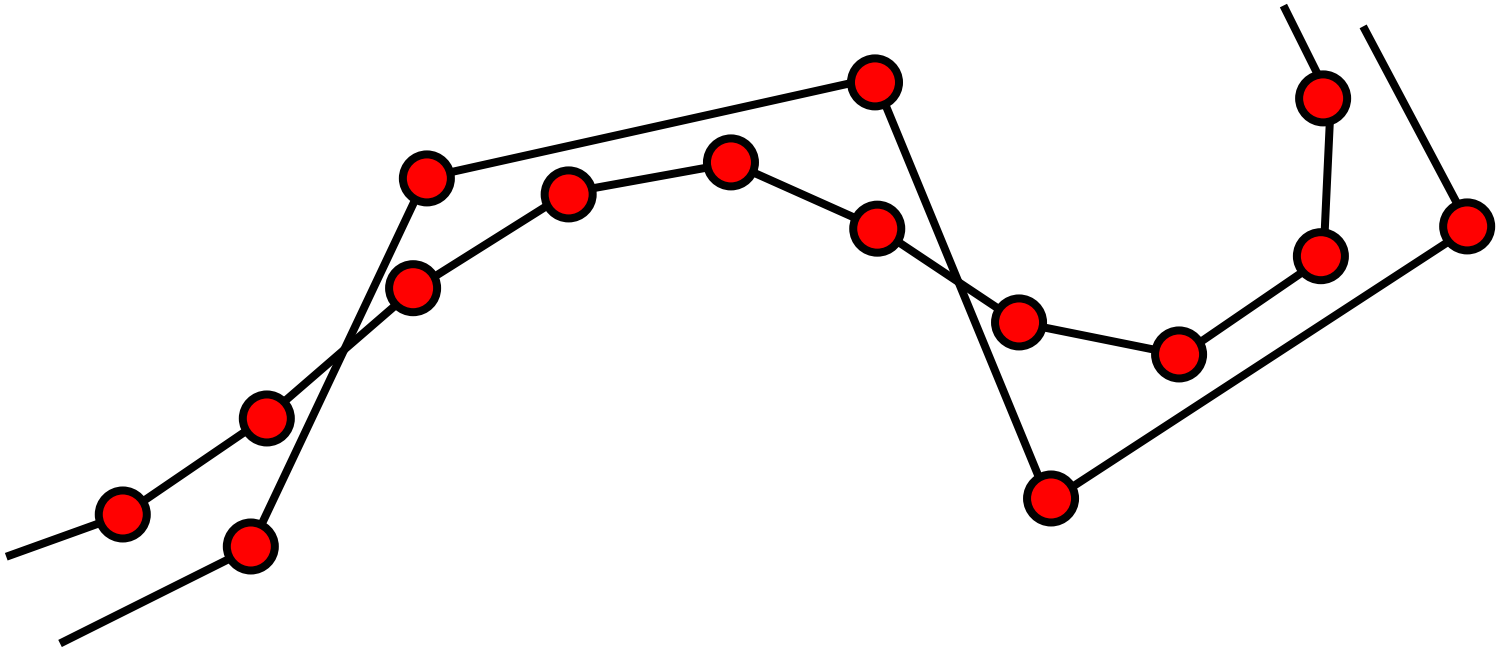
Primal Splitting



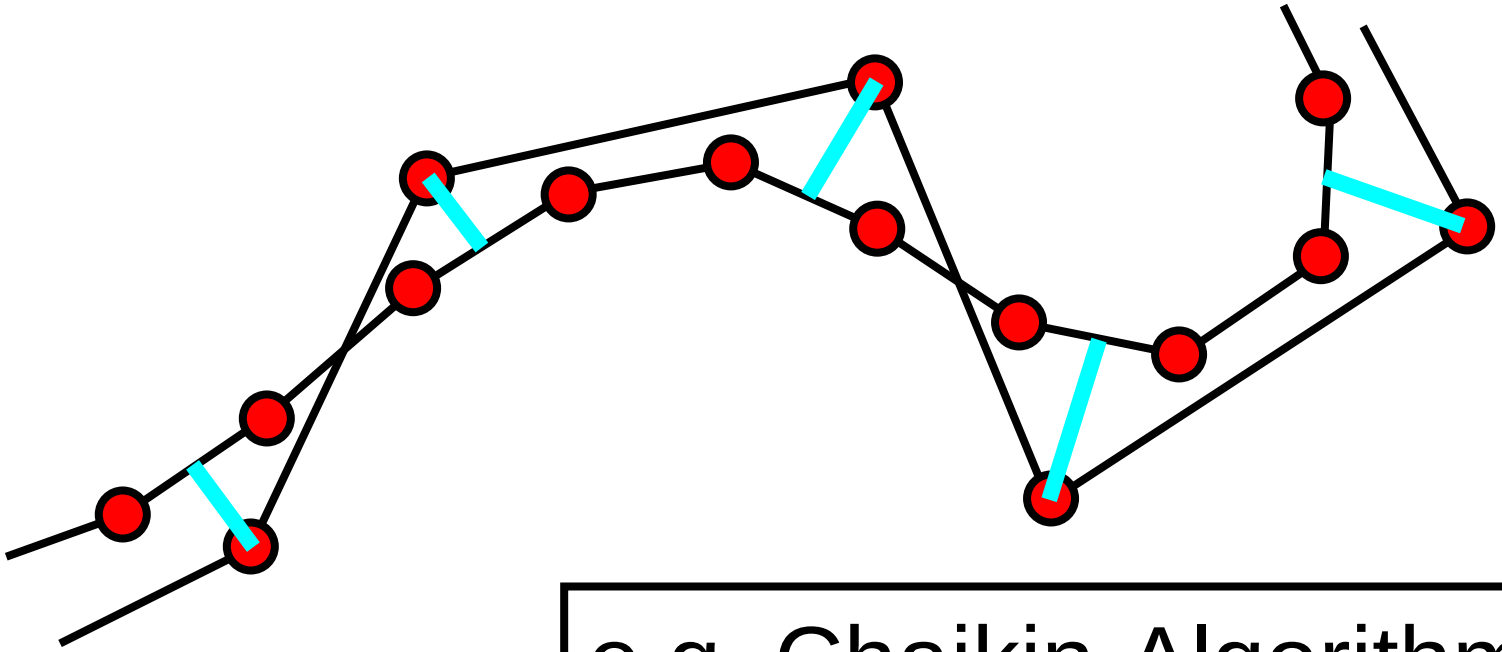
Dual Splitting



Dual Splitting



Dual Splitting

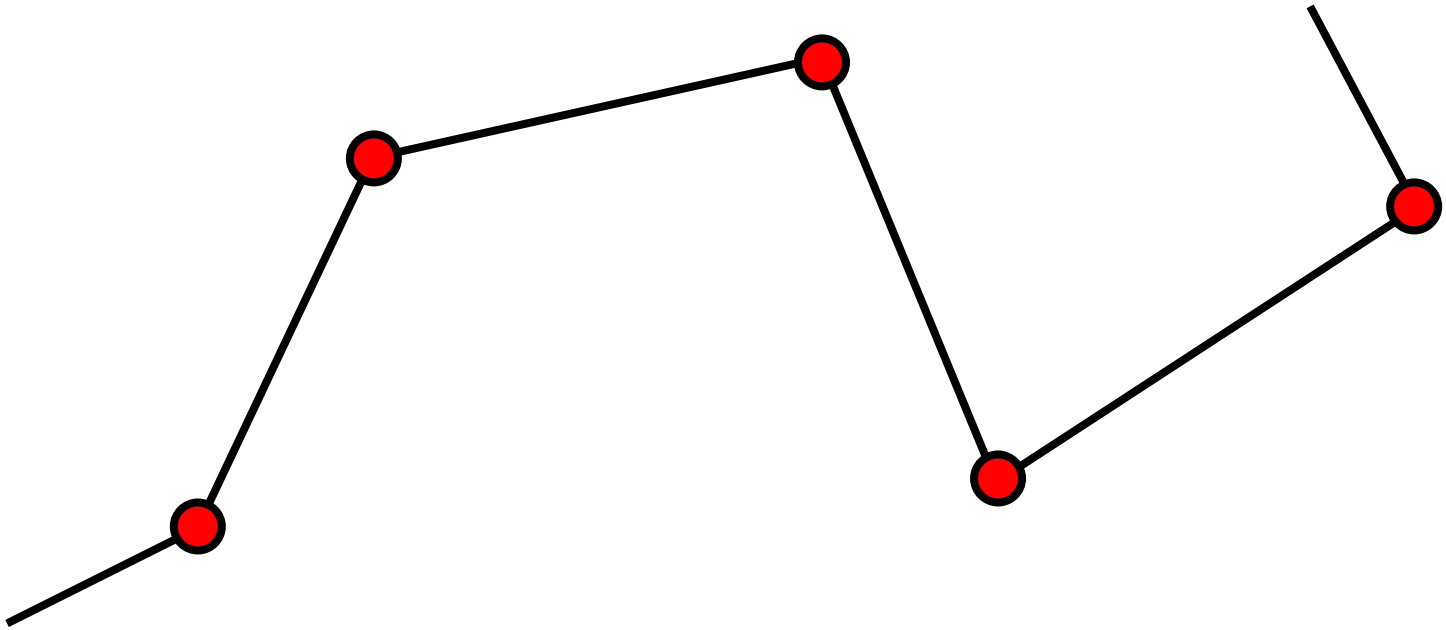


e.g. Chaikin-Algorithm

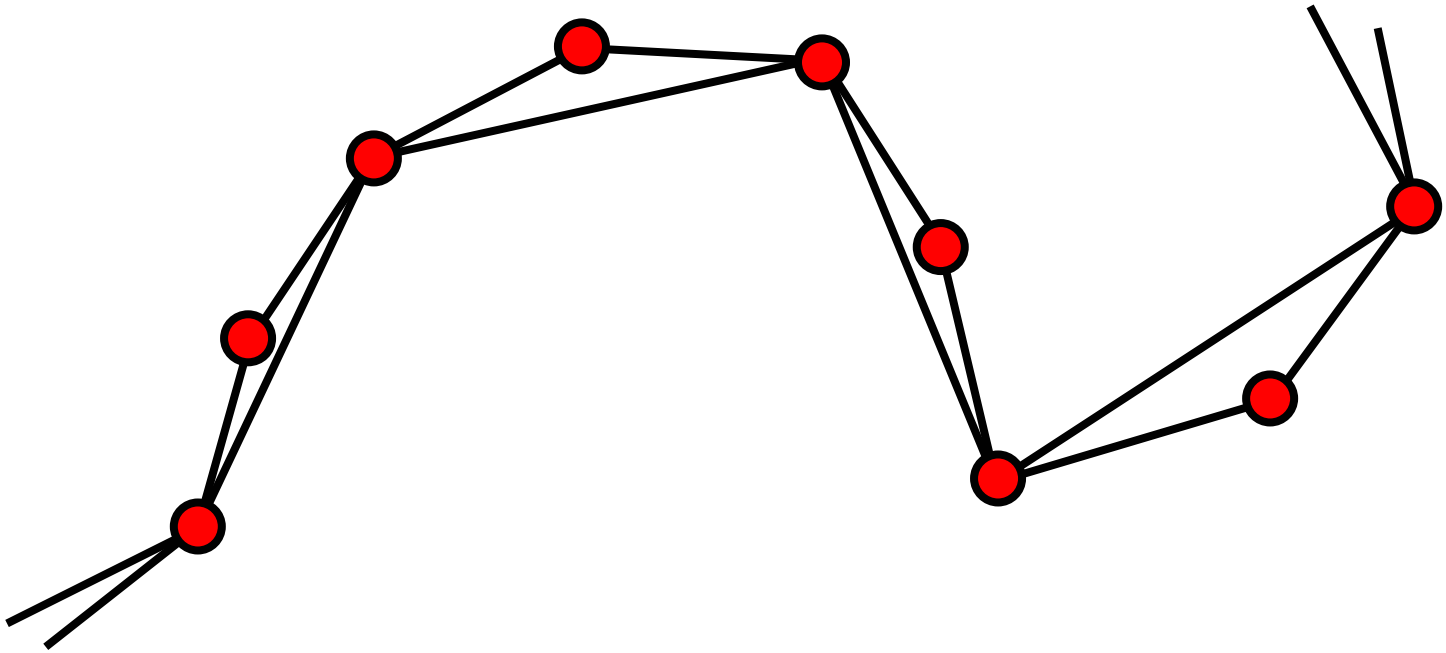
Smoothing Rules

- New control point positions are computed by linear combinations
- Two rules (odd / even)
- Generalization of spline masks
- Properties
  - C^k-smoothness
  - interpolation

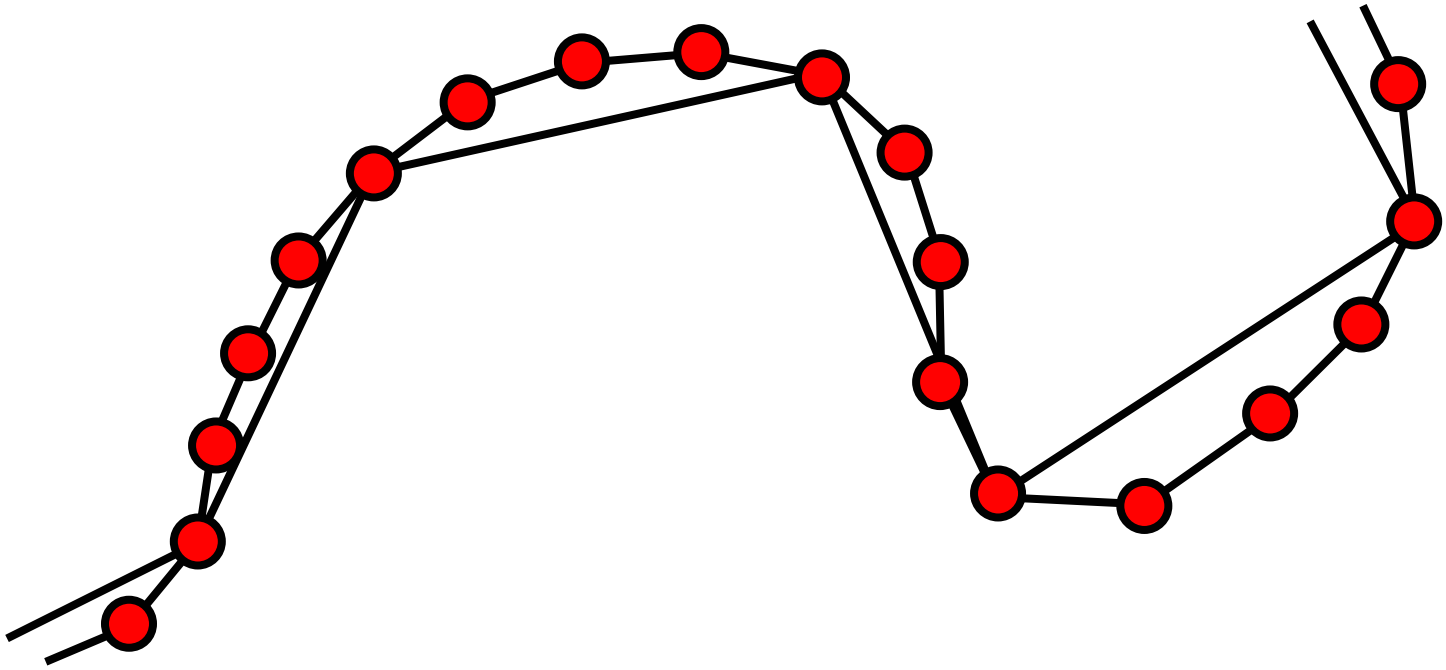
Interpolatory Subdivision



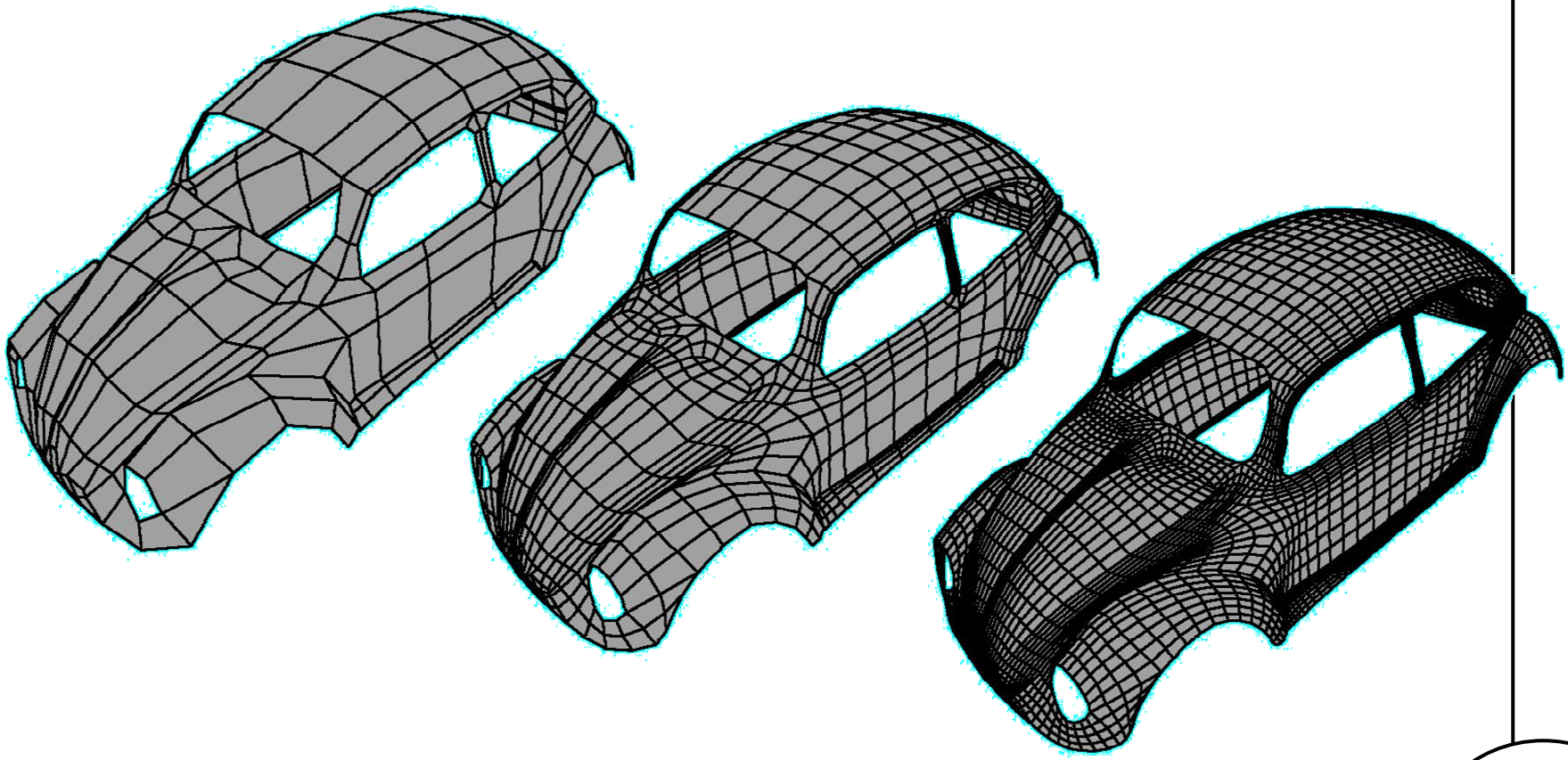
Interpolatory Subdivision



Interpolatory Subdivision



Surface Subdivision



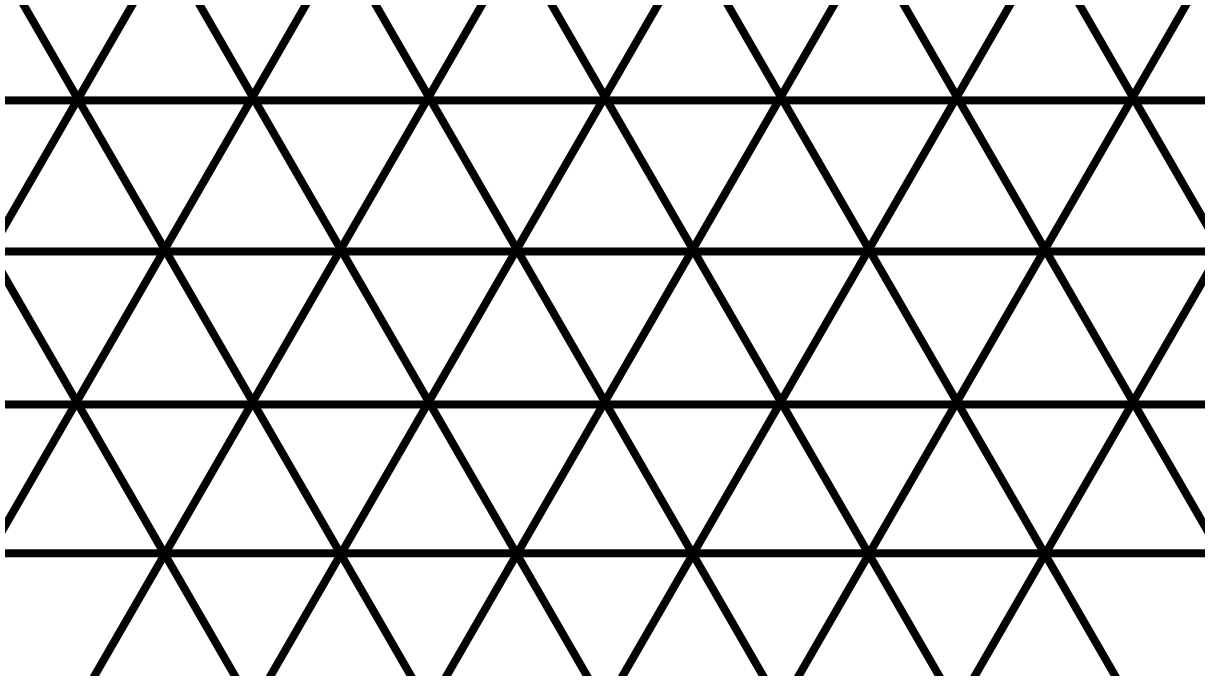
RWTH AACHEN

Informatik VIII
Computer Graphics & Multimedia
Leif P. Kobbelt

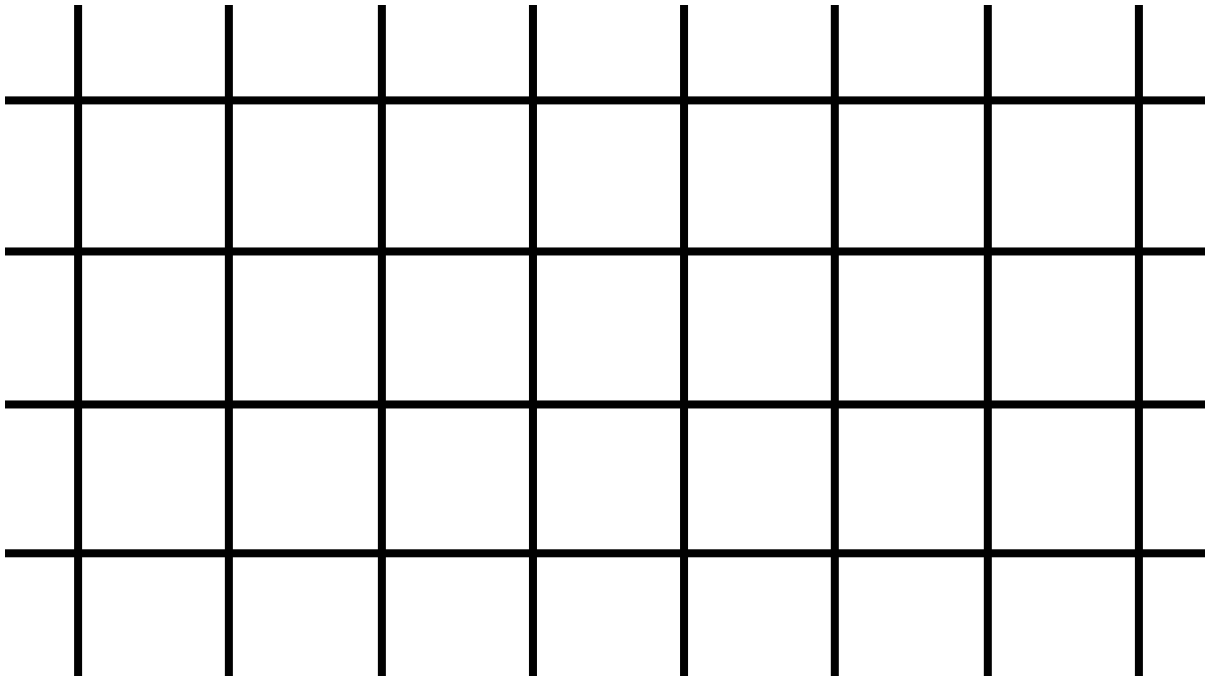
Surface Subdivision

- Splitting
  - Primal / Dual
  - Triangle / Quad - based
- Smoothing
  - Polynomial
  - Non-polynomial
  - Interpolatory

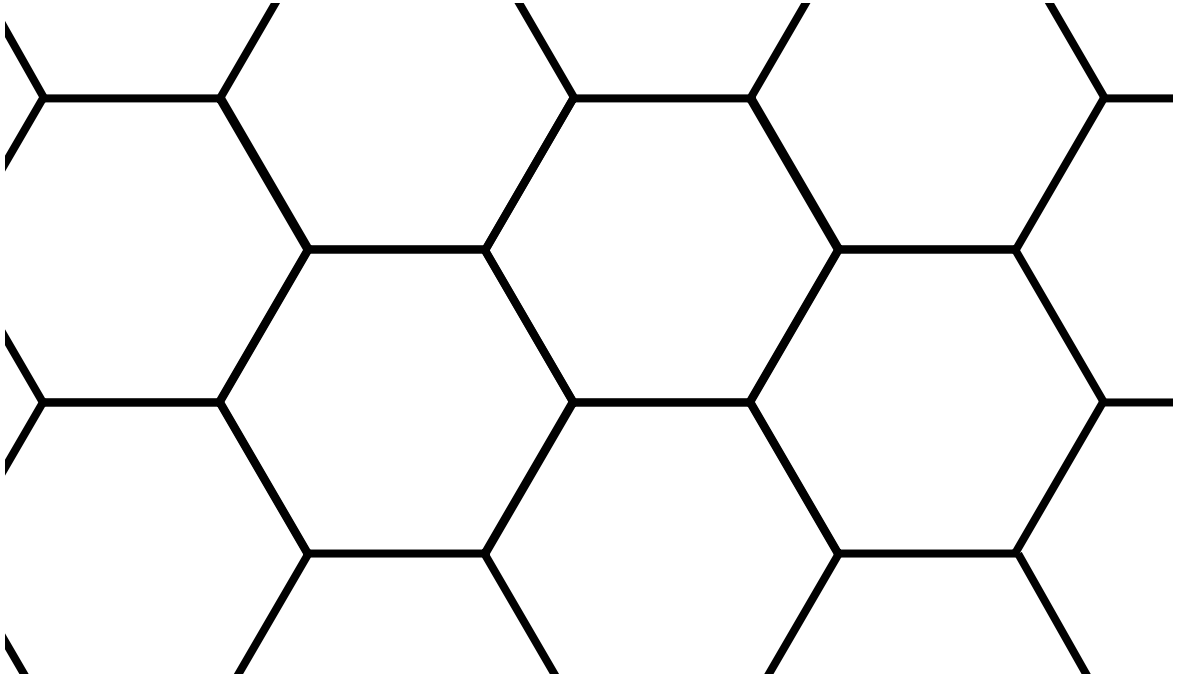
Tilings of the plane



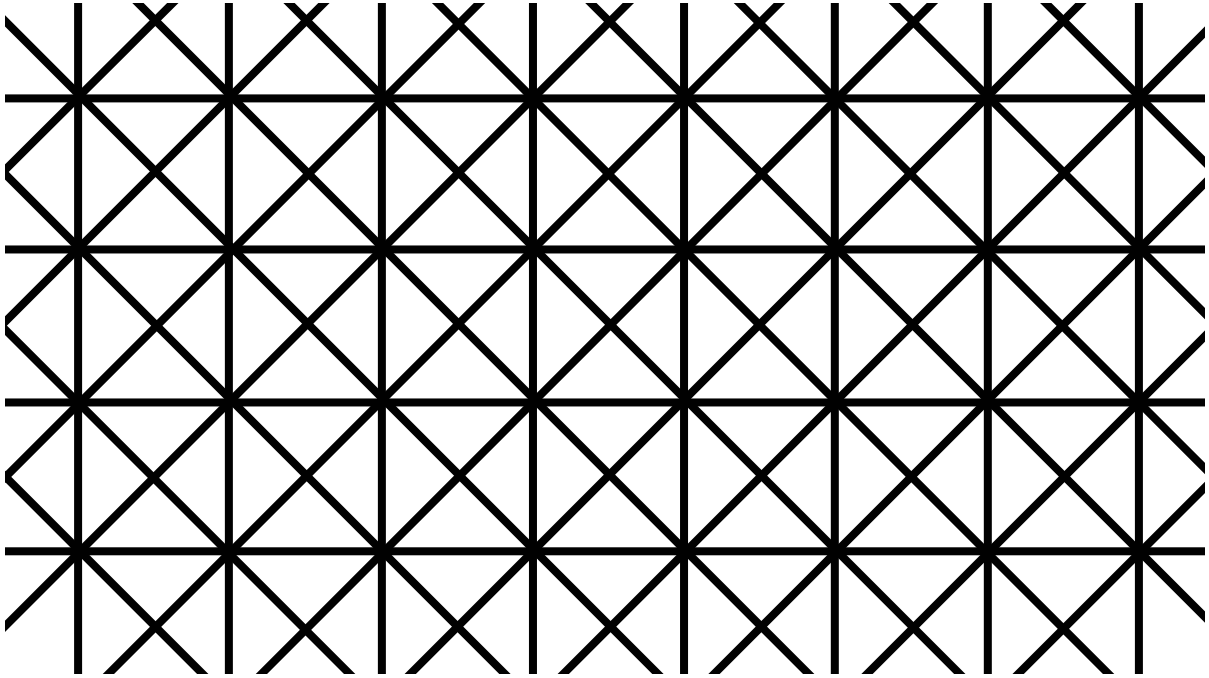
Tilings of the plane



Tilings of the plane



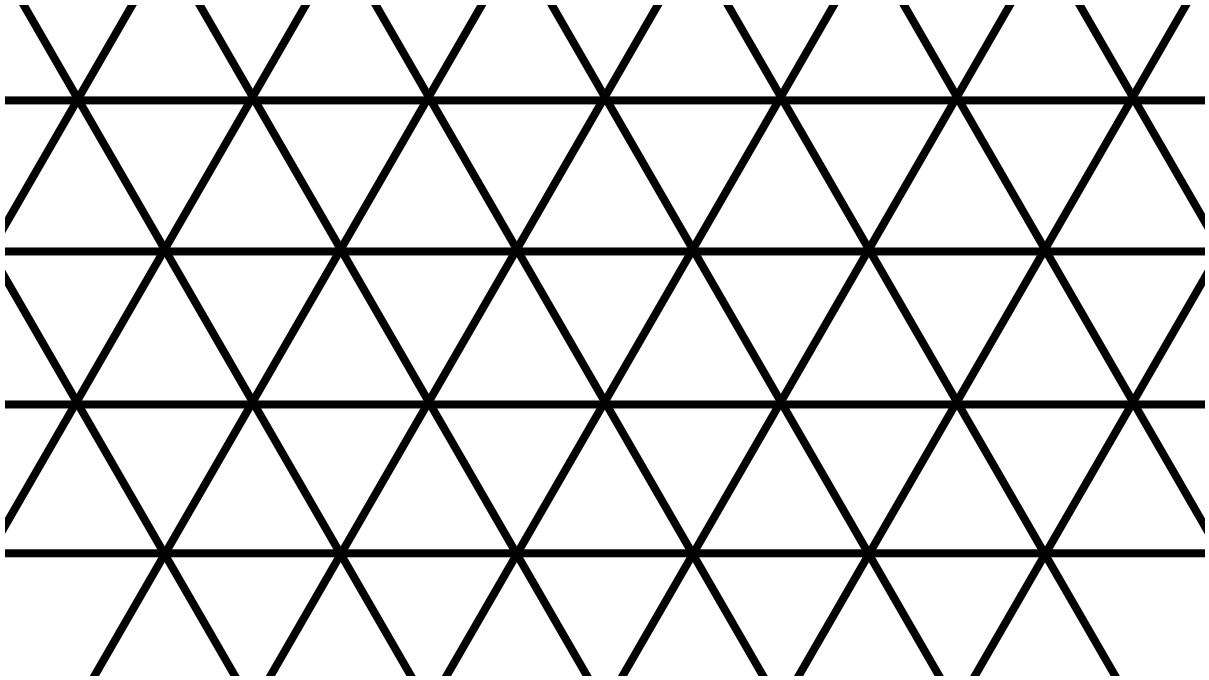
Tilings of the plane



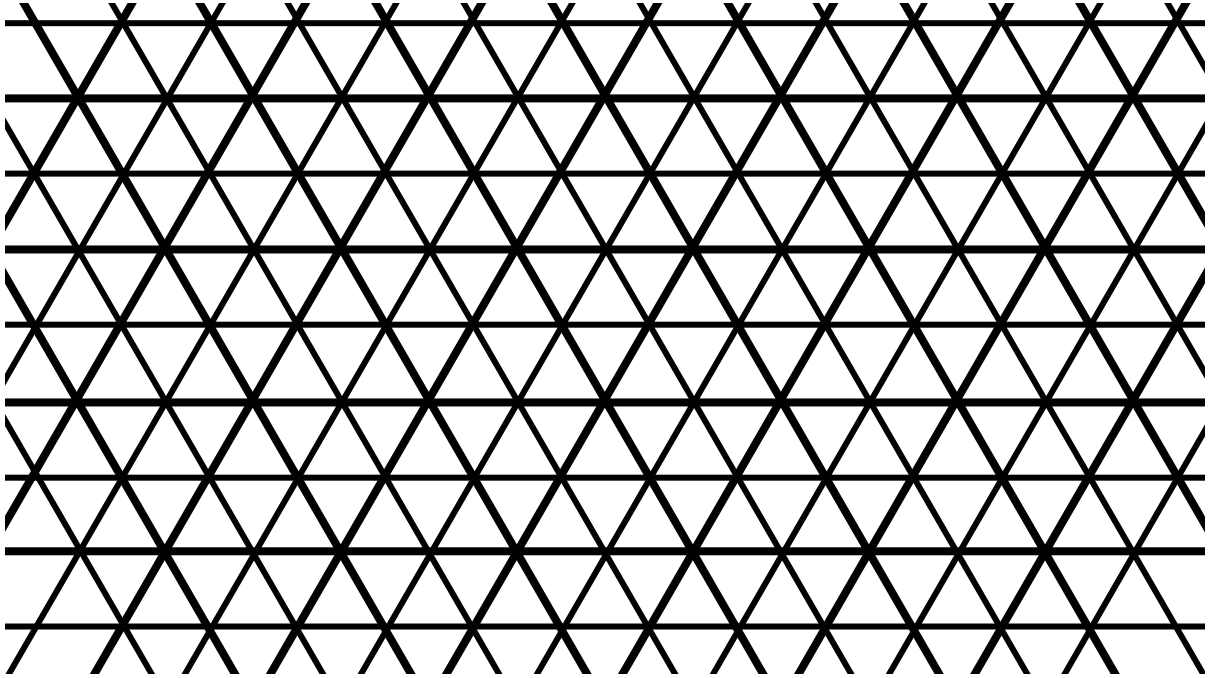
Tilings of the plane

- To each tiling corresponds a uniform refinement operator the maps the grid to a scaled grid.
- These operators form the basis of the subdivision splitting operators

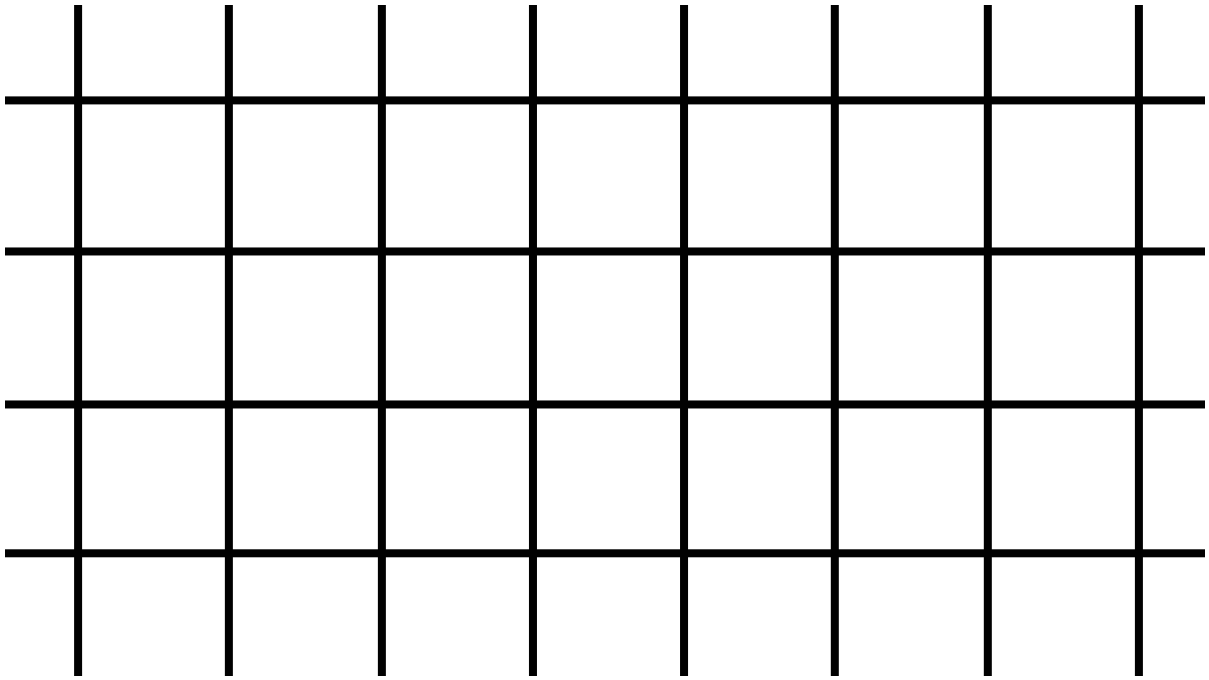
Uniform Refinement



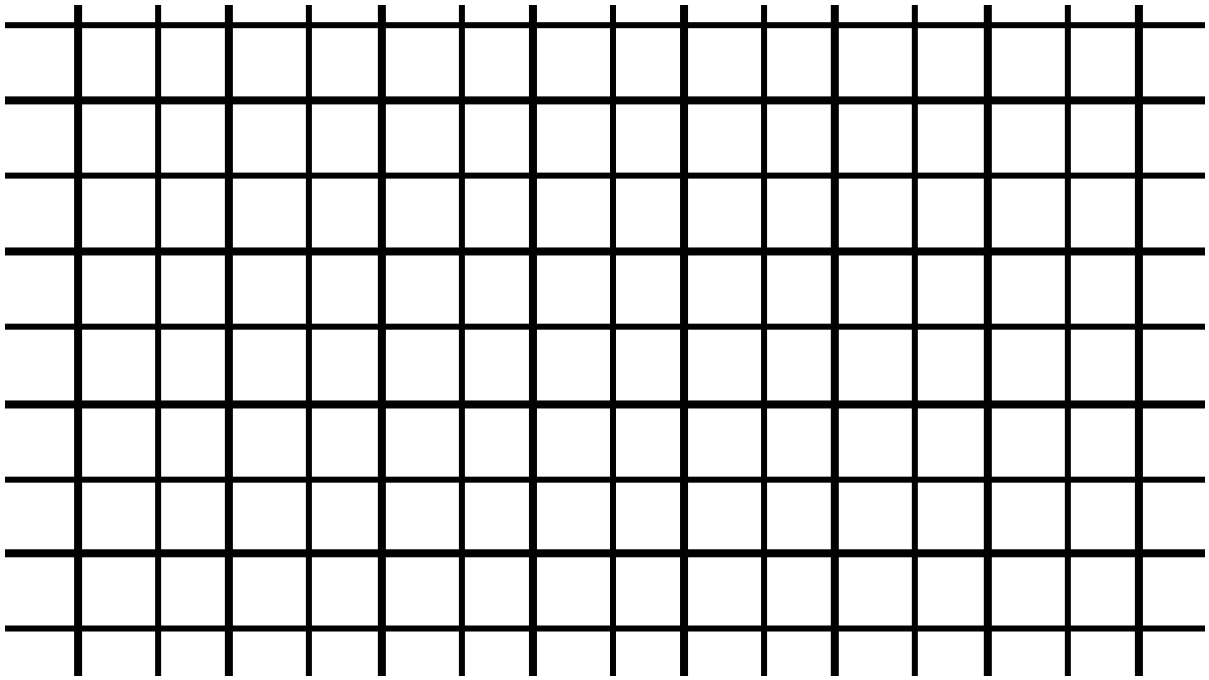
Uniform Refinement



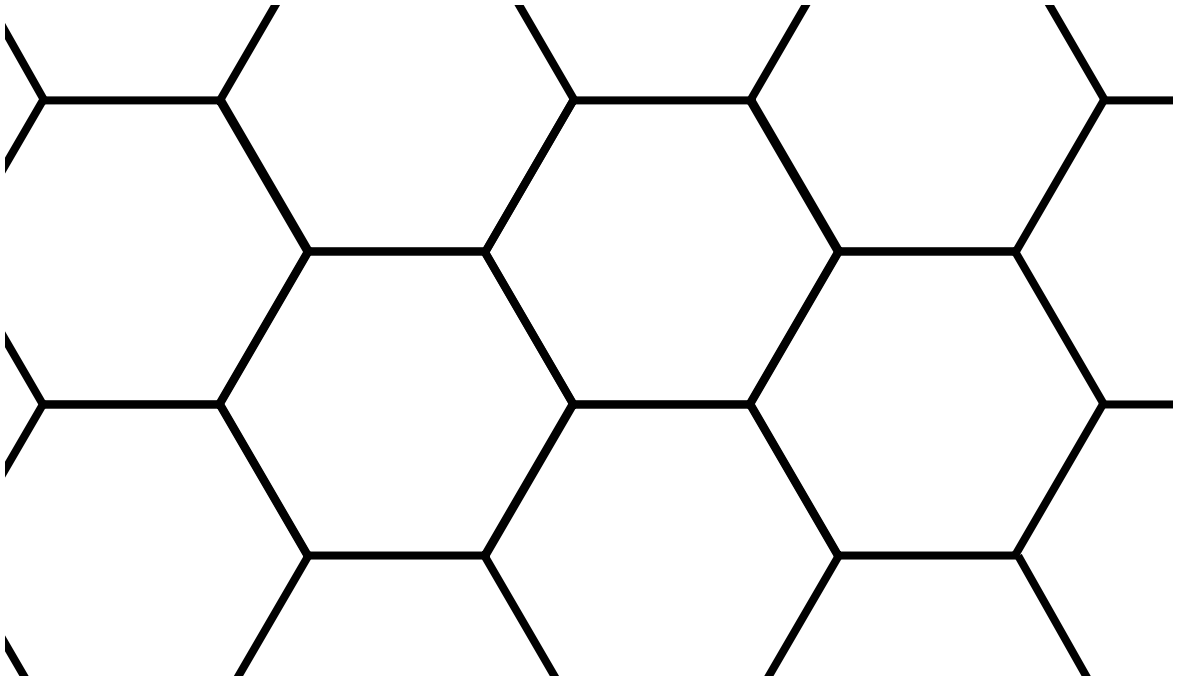
Uniform Refinement



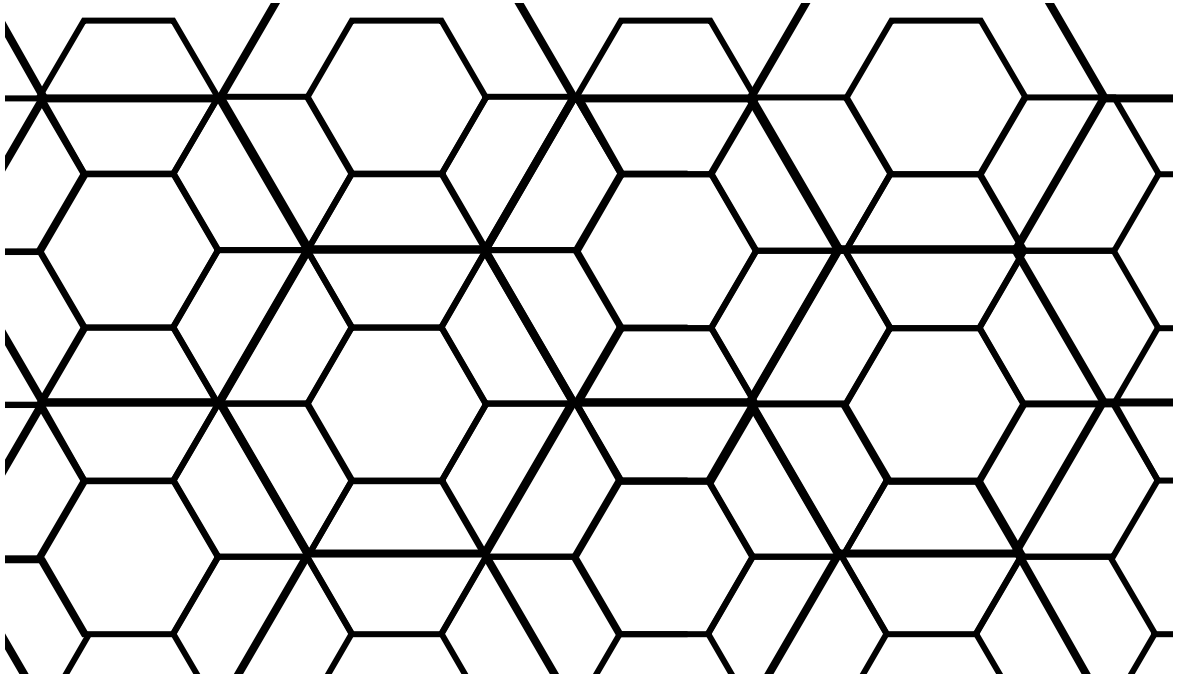
Uniform Refinement



Uniform Refinement



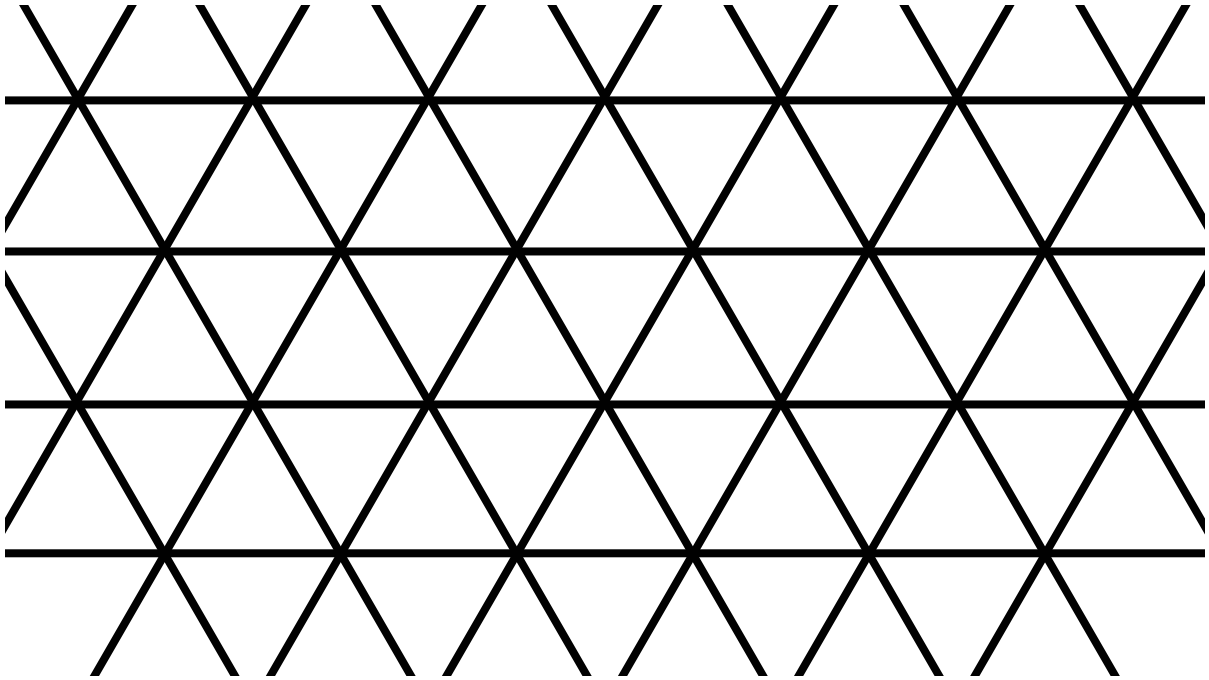
Uniform Refinement



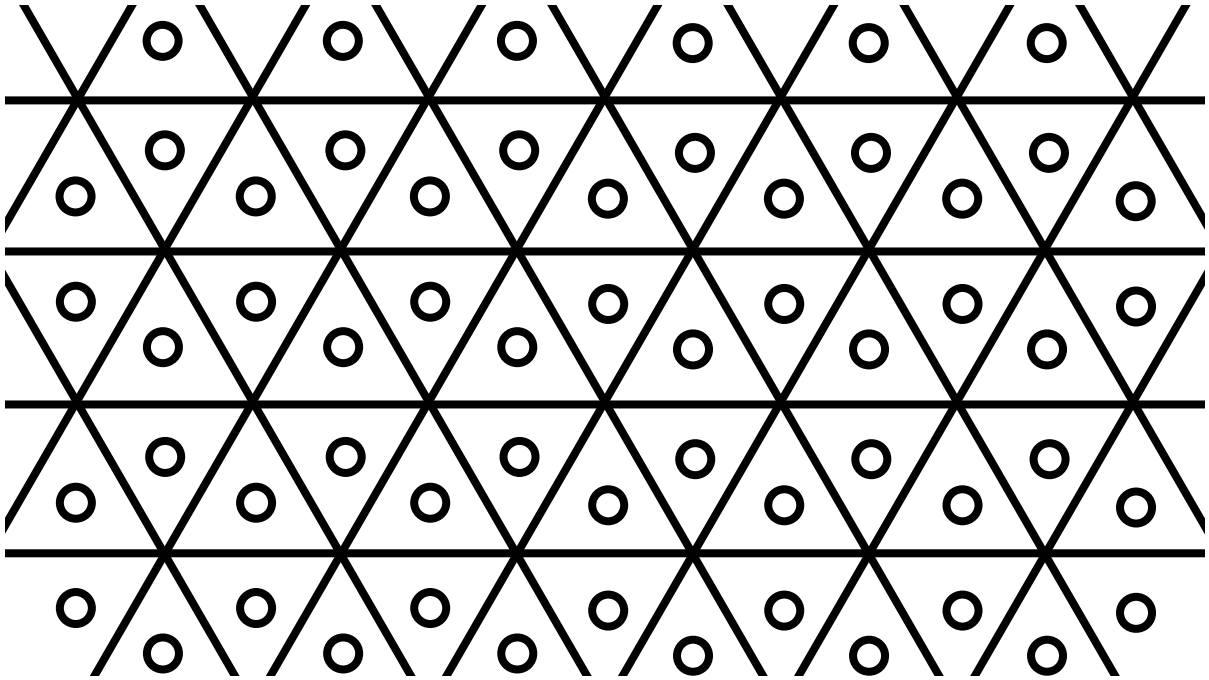
Mesh Duality

- Meshes consist of k-dim primitives
  - Vertices (0-dim)
  - Edges (1-dim)
  - Face (2-dim)
- A dual mesh is obtained by replacing each k-dim primitive with a (2-k)-dim one and keeping the graph topology unchanged

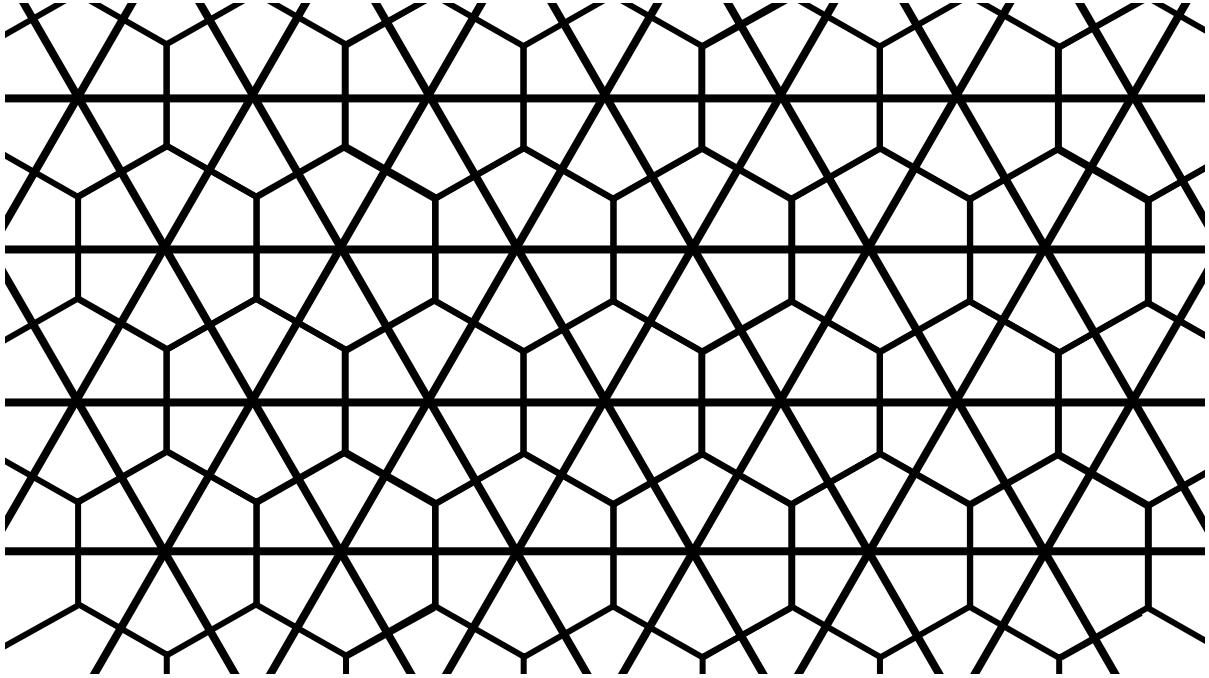
Mesh Duality



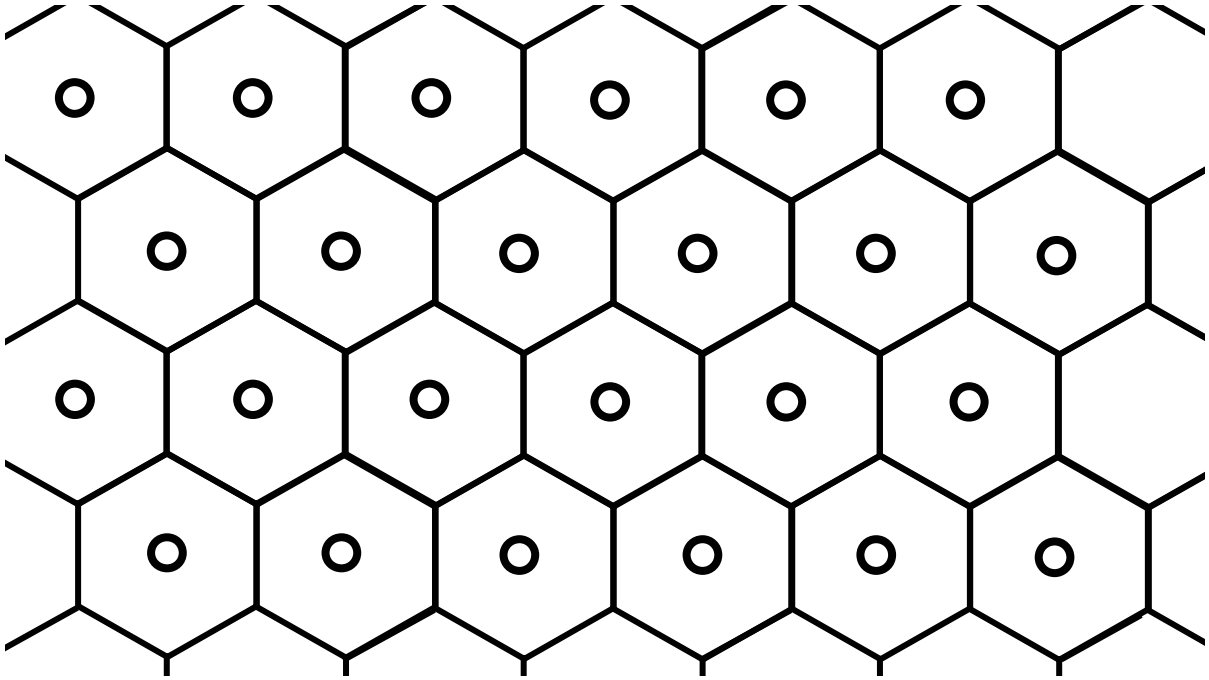
Mesh Duality



Mesh Duality



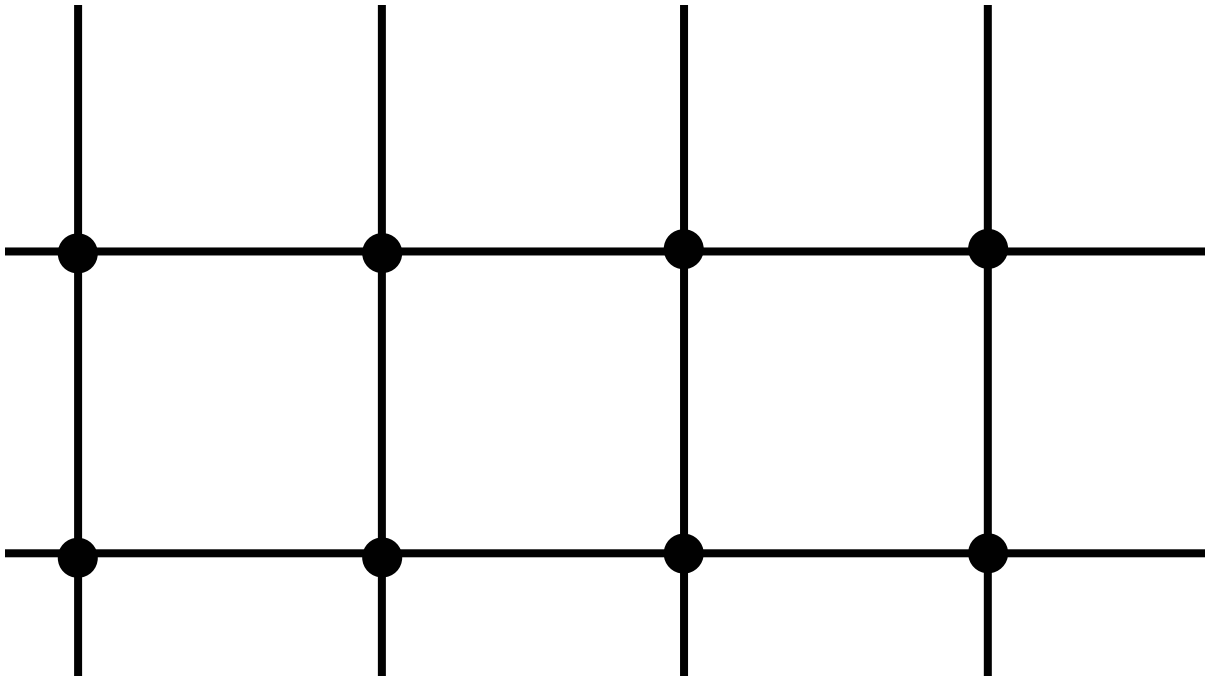
Mesh Duality



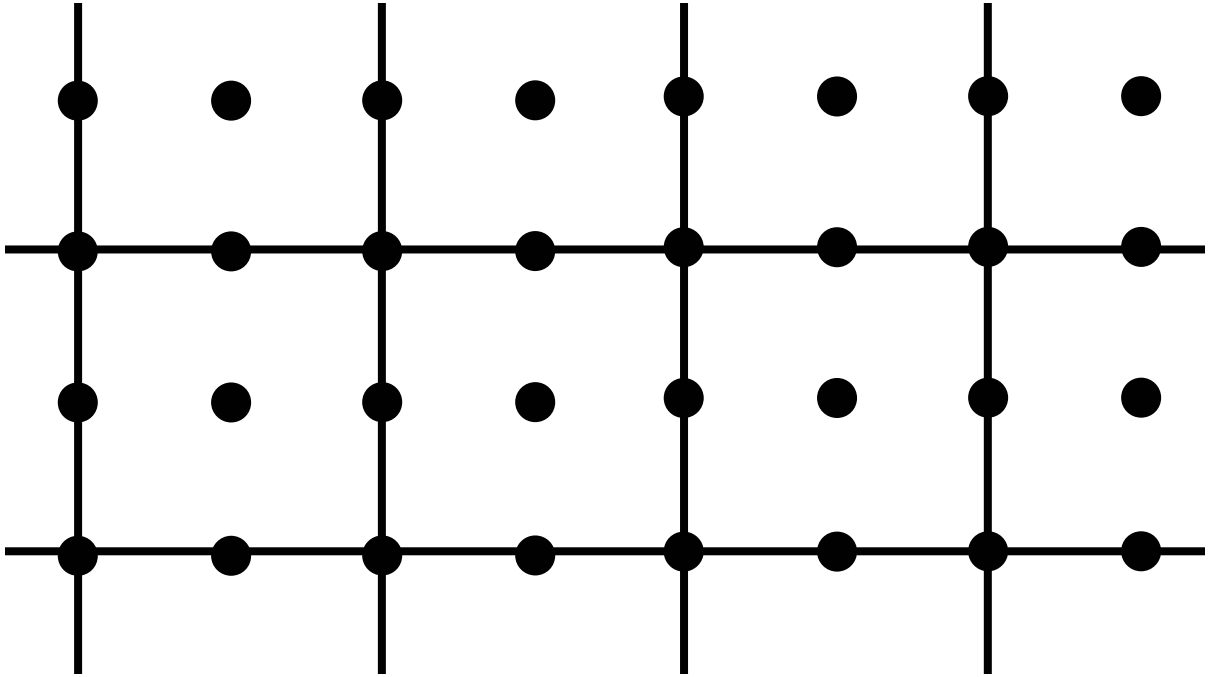
Splitting Operator

- Splitting operators for subdivision schemes can be generated by combinations of *uniform refinement* and *duality*.
  - Catmull / Clark
  - Doo / Sabin
  - Loop
  - Sqrt(3)

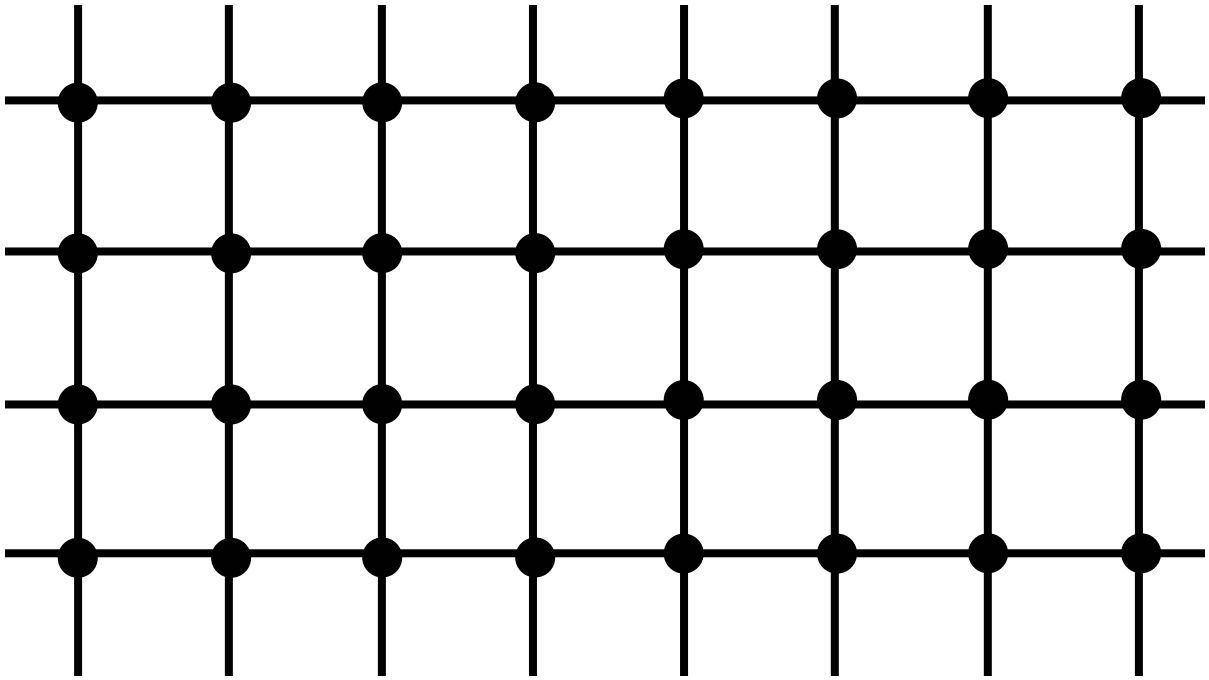
Primal / Quad



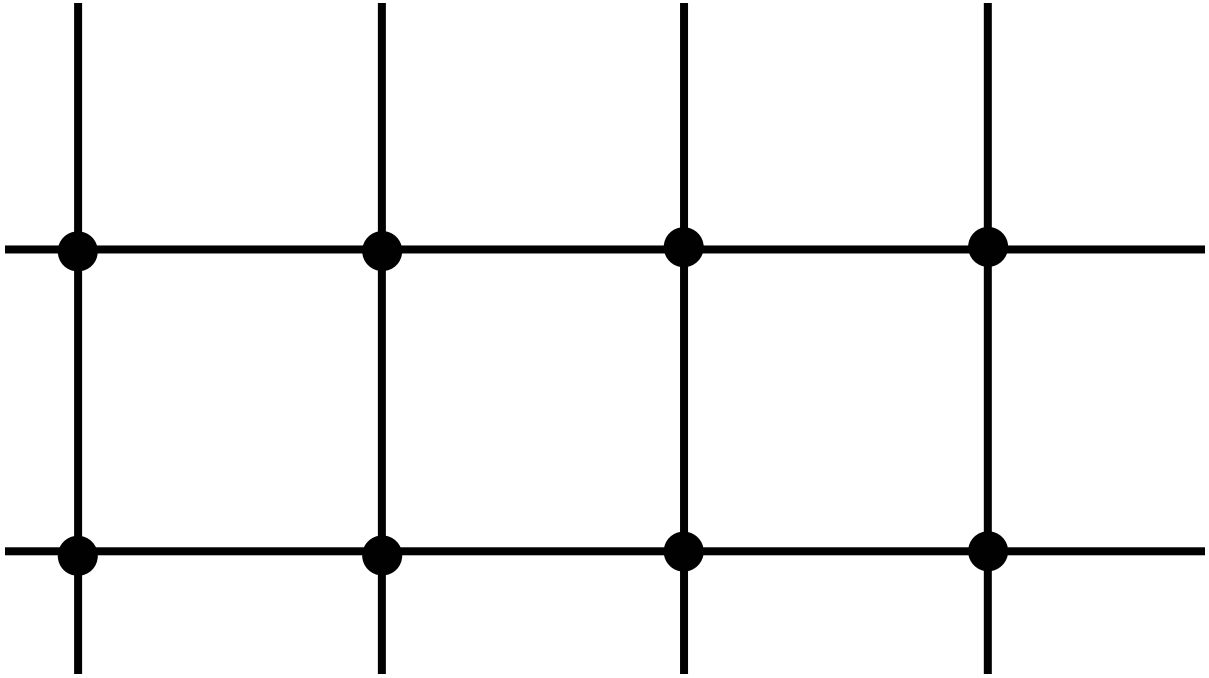
Primal / Quad



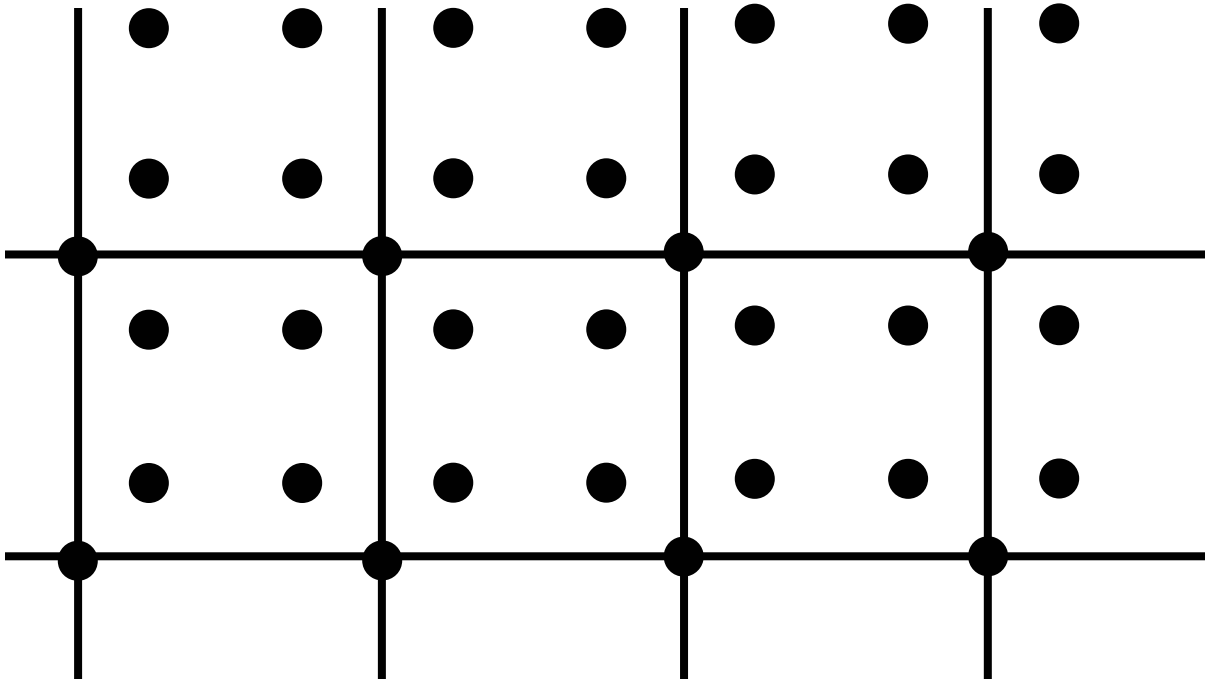
Primal / Quad



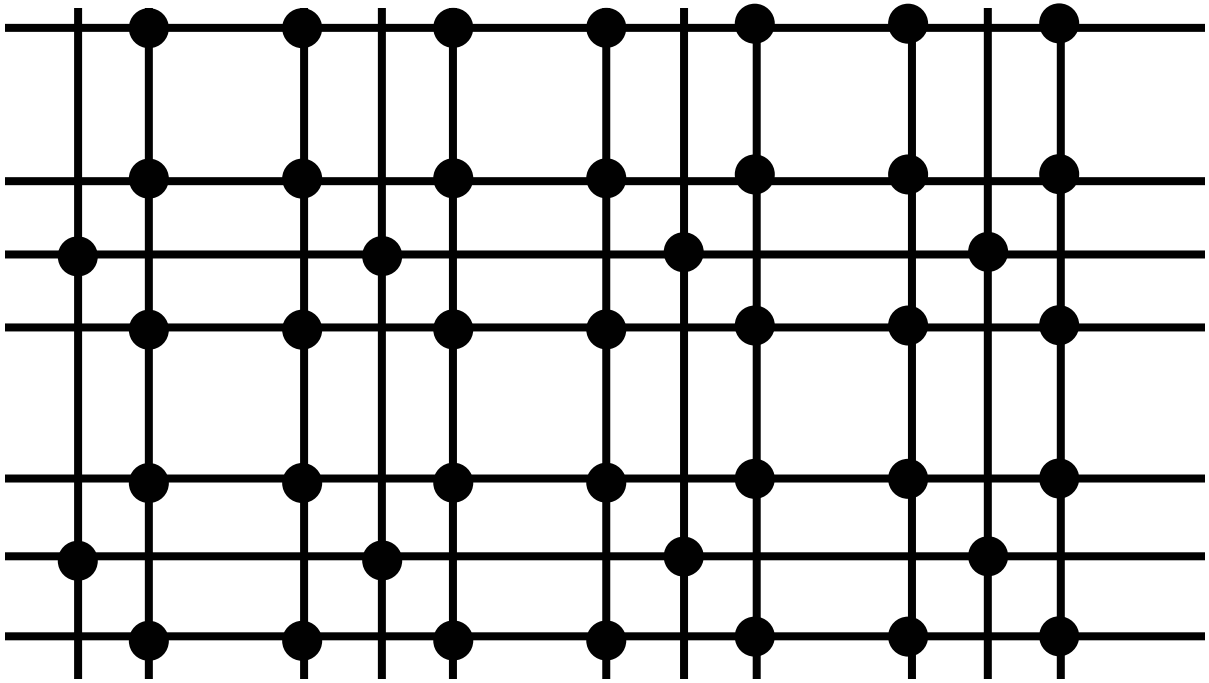
Dual / Quad



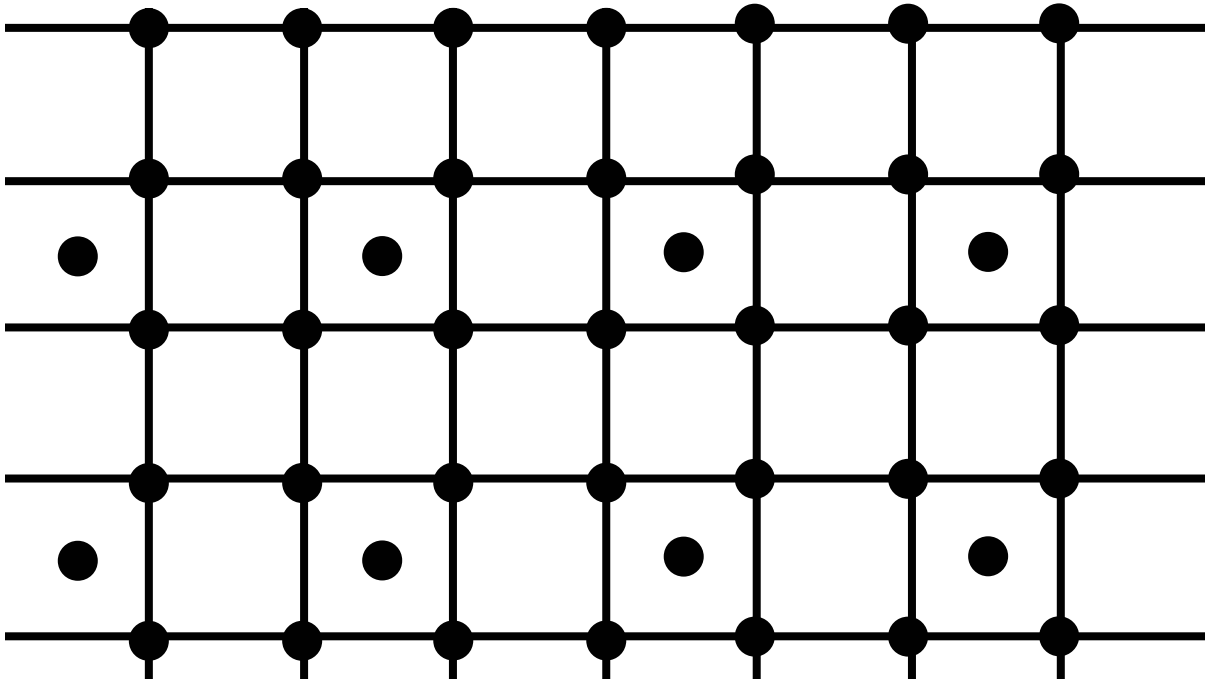
Dual / Quad



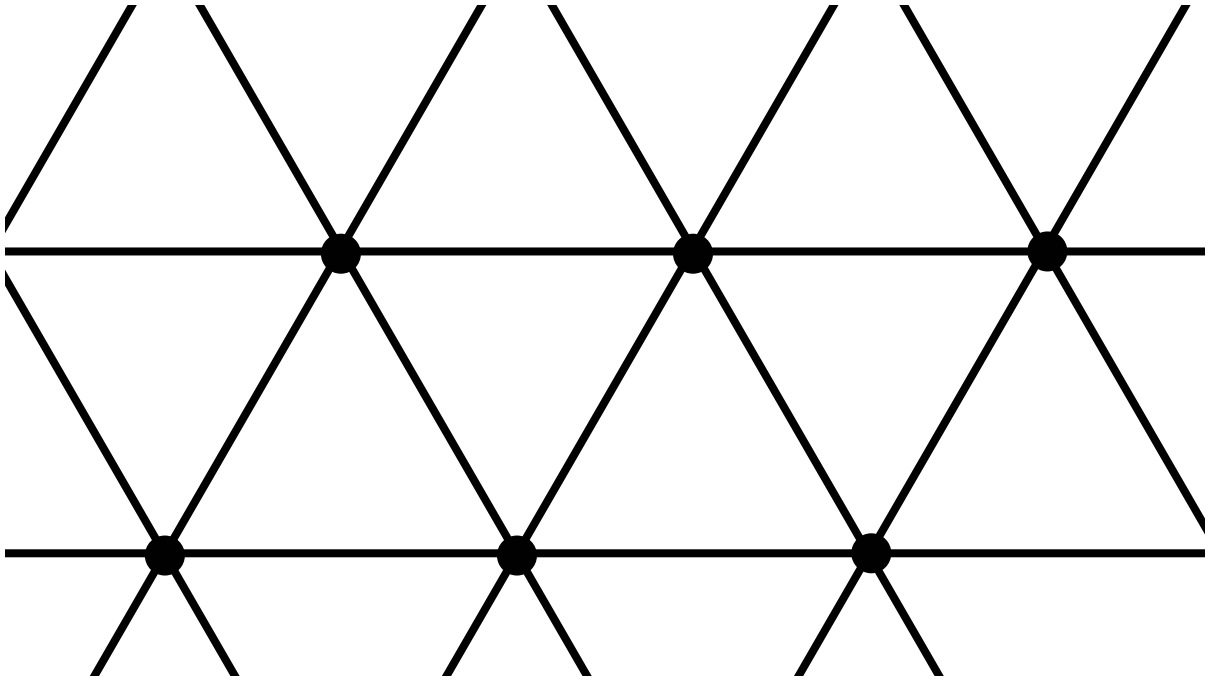
Dual / Quad



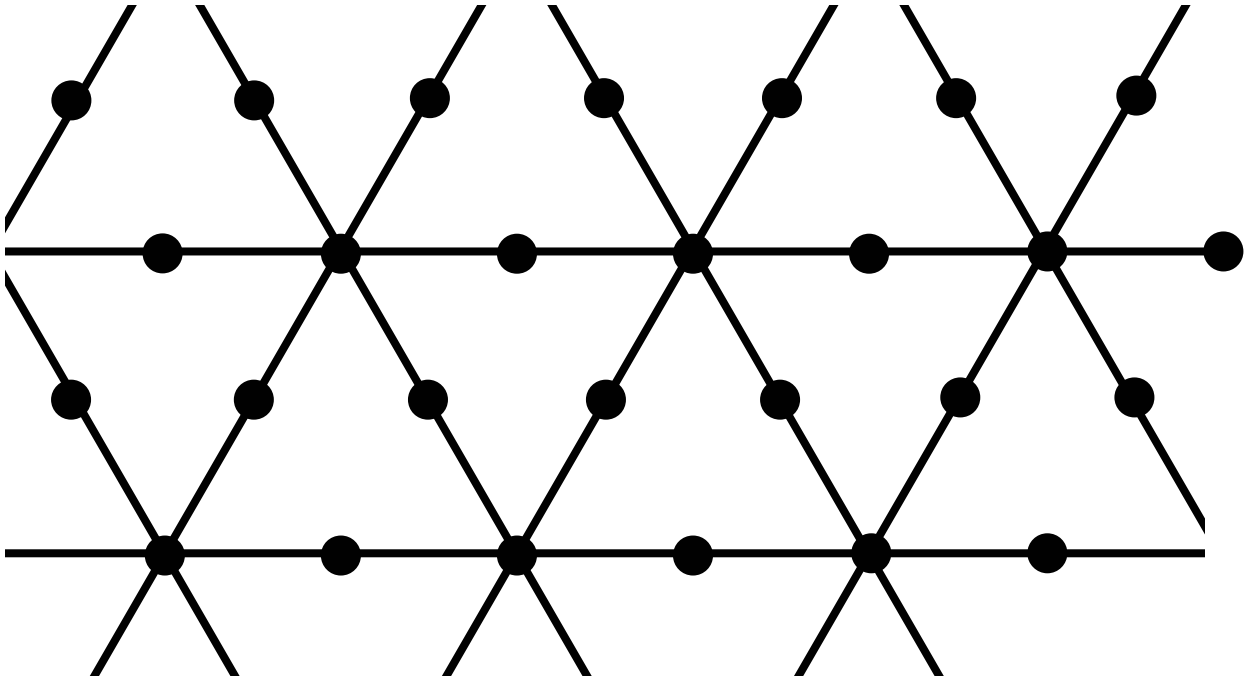
Dual / Quad



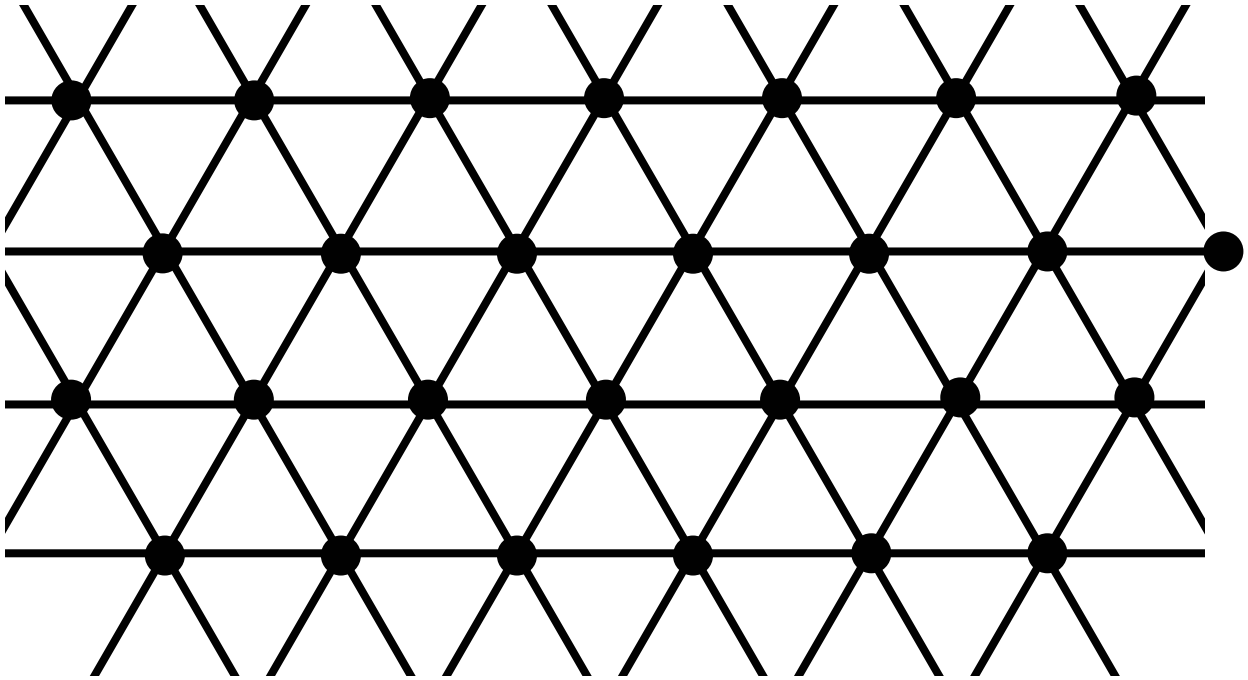
Primal / Triangle



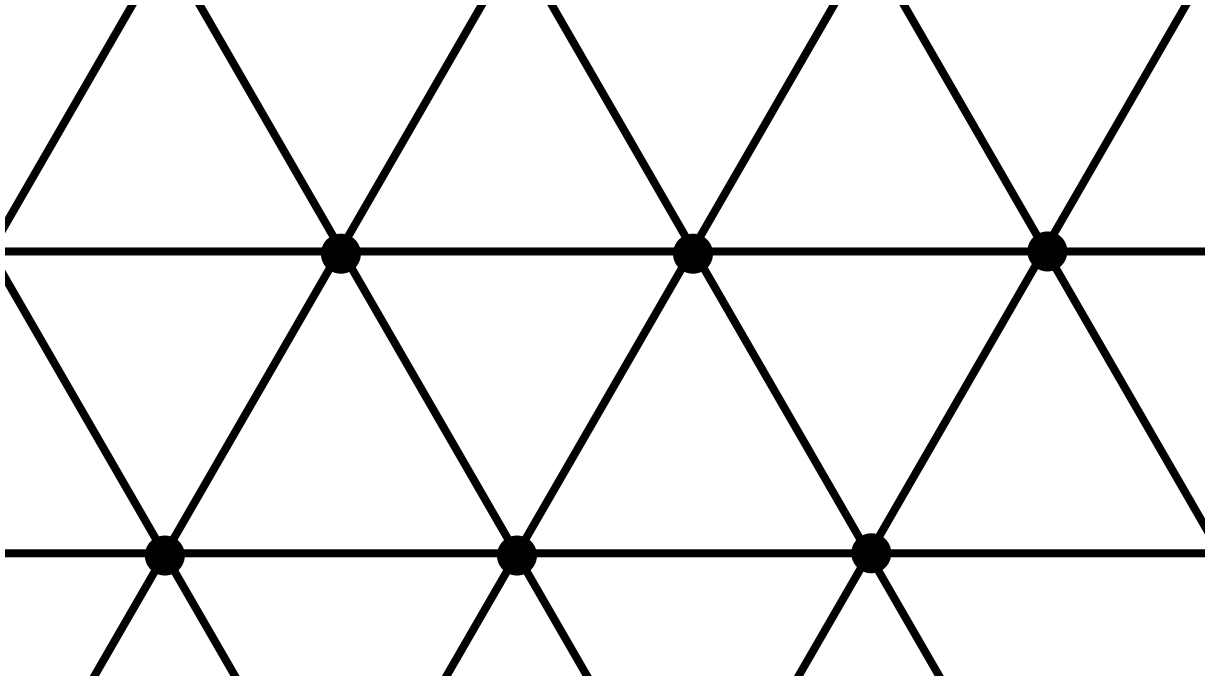
Primal / Triangle



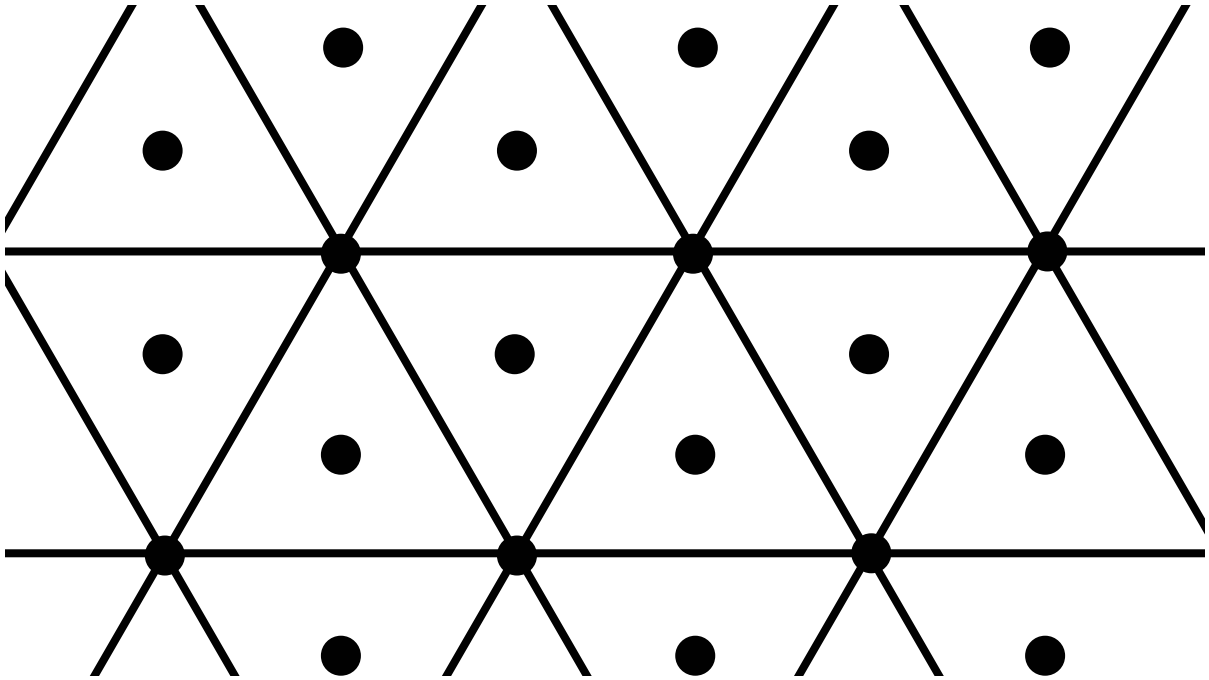
Primal / Triangle



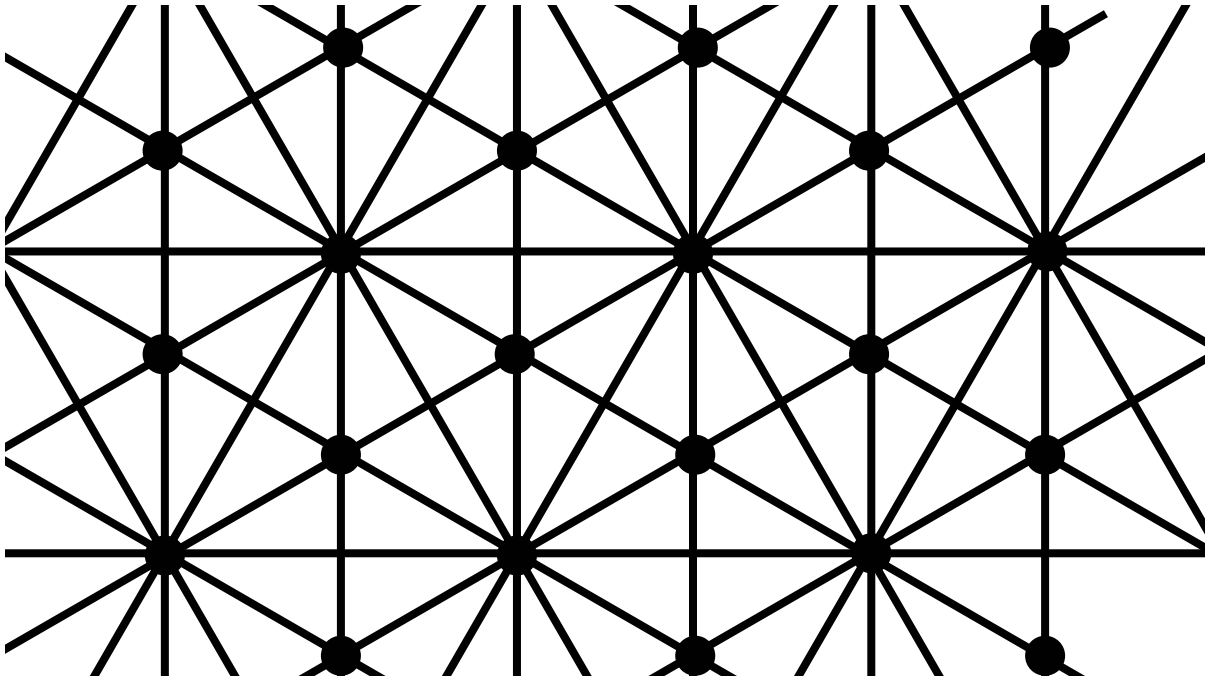
„Dual“ / Triangle



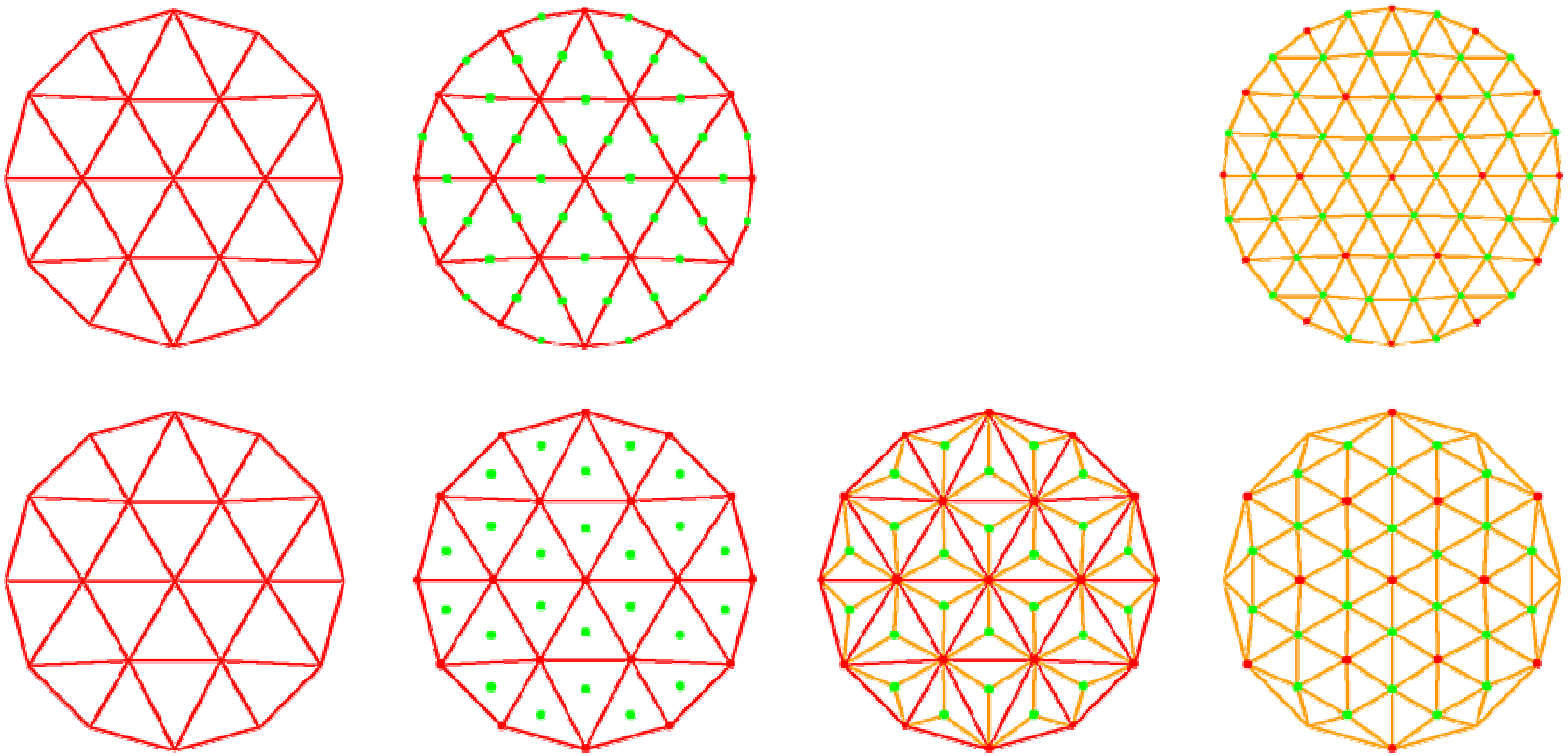
„Dual“ / Triangle



„Dual“ / Triangle



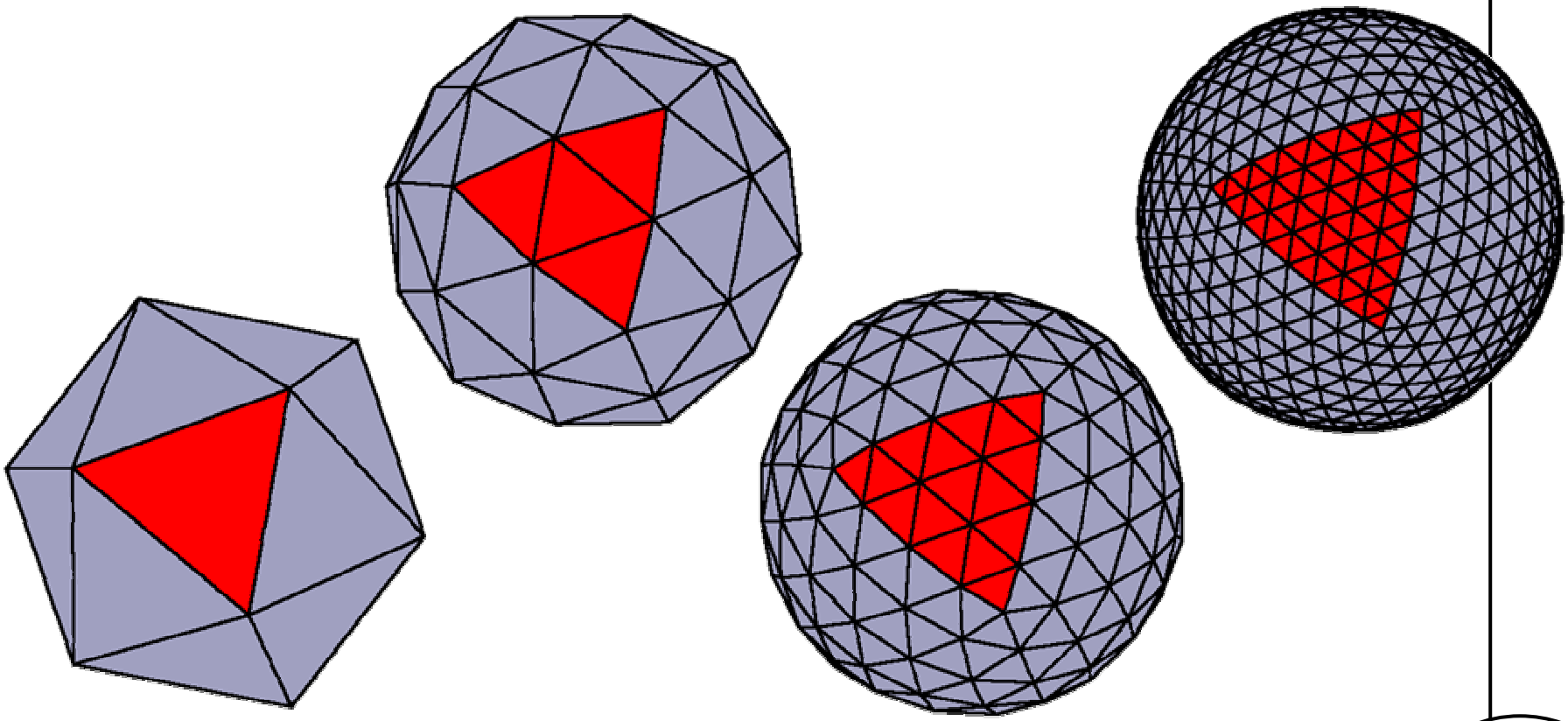
Triangle Refinement



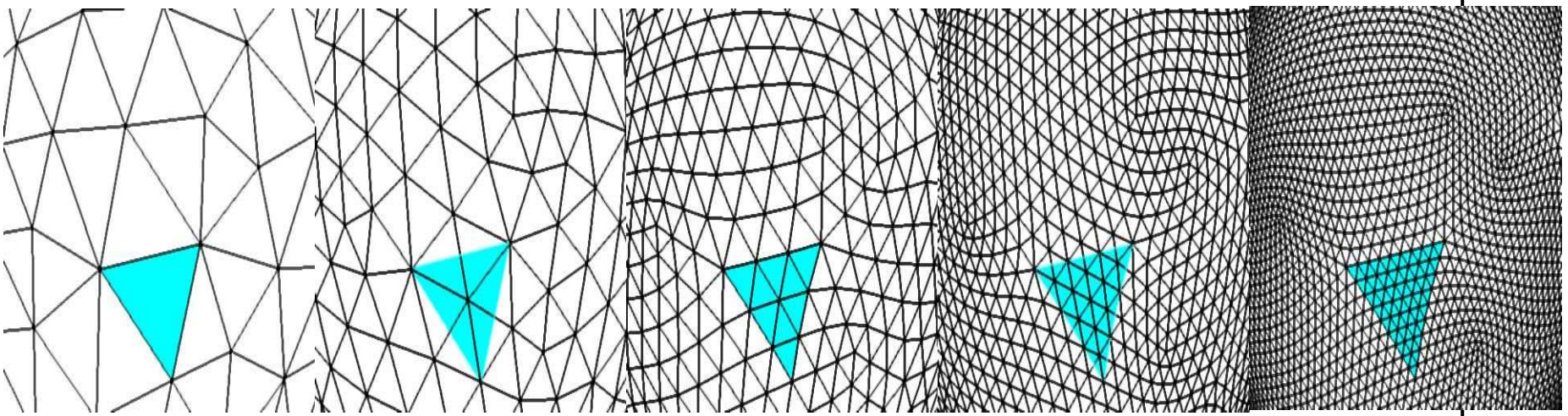
Triangle Refinement

- Splitting operators can be generalized to arbitrary (non-regular) meshes
  - Define face-splitting procedure
  - Apply to all faces in the mesh
- Only regular vertices are generated
- Number of irregular vertices is constant
- Semi-regular meshes

Triangle Refinement

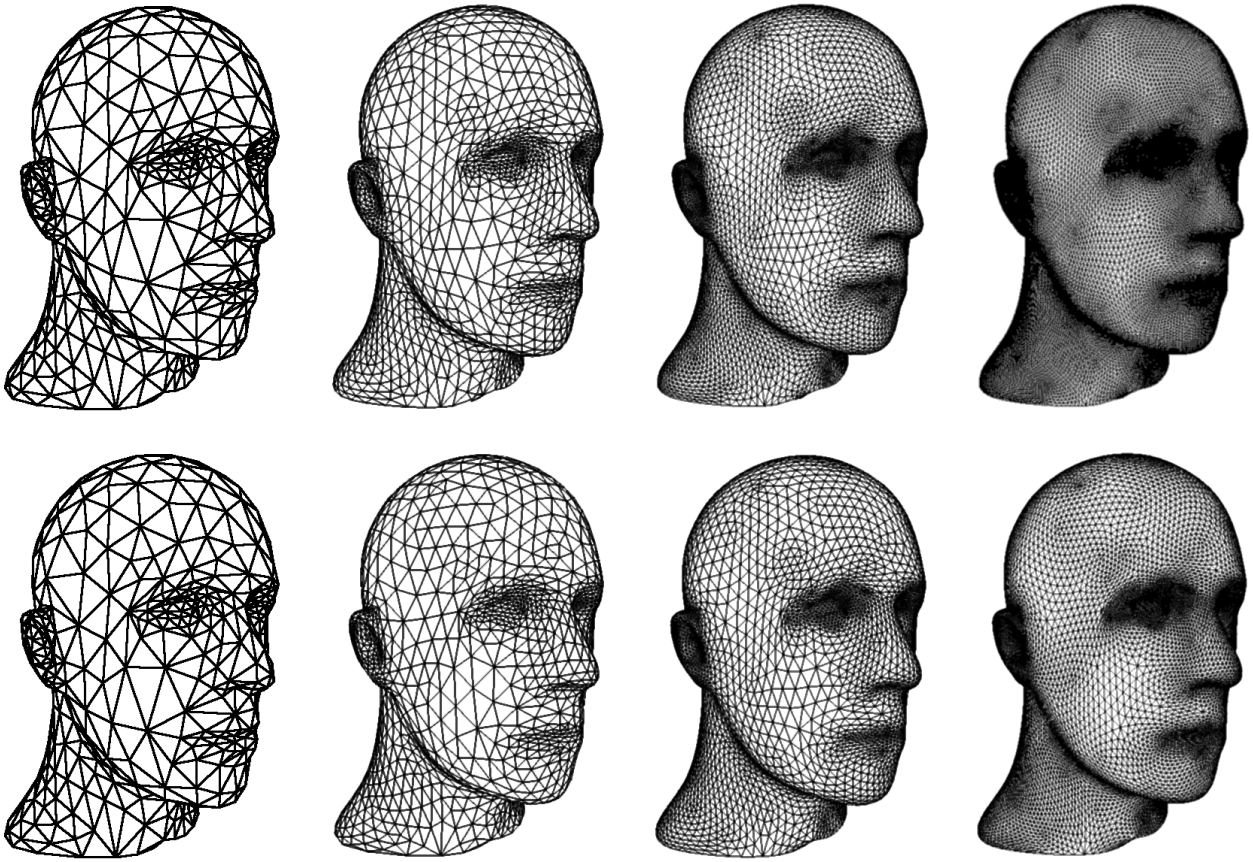


Triangle Refinement

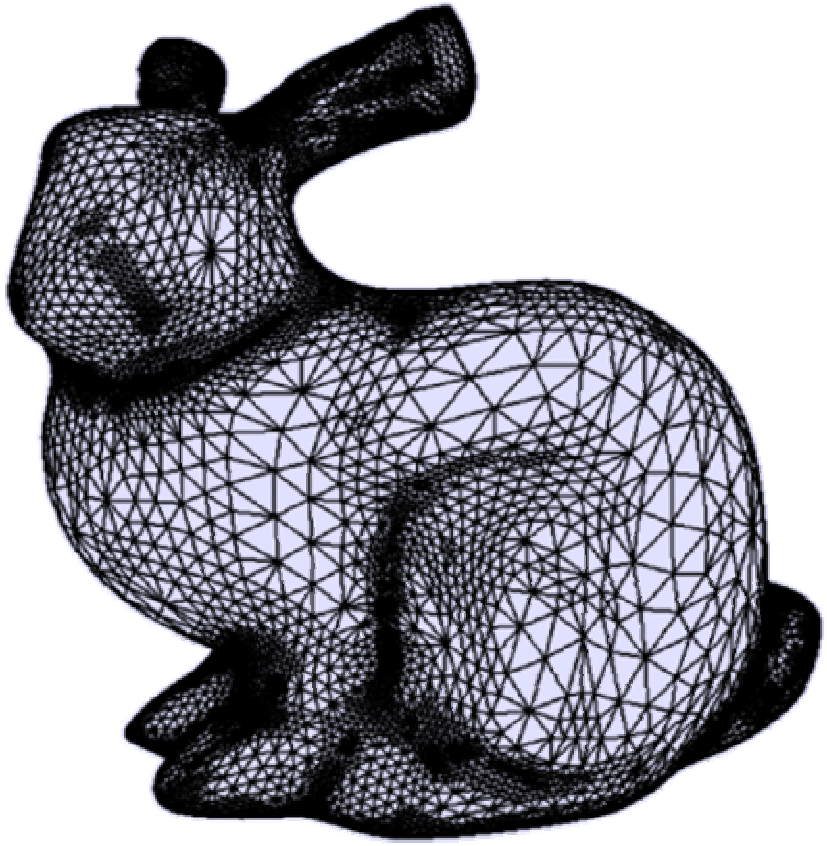


Sqrt(3) - refinement

Exponential Growth



Adaptive Refinement

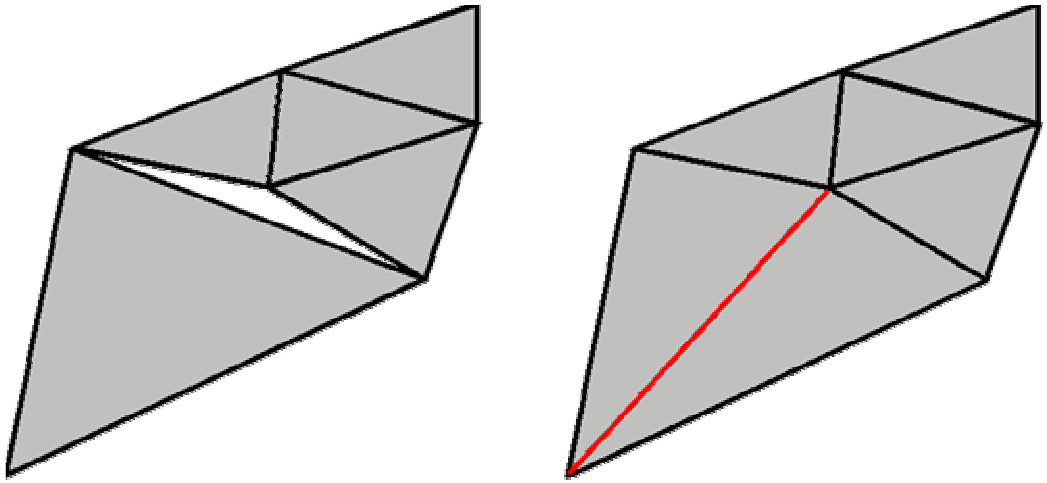


Adaptive Refinement

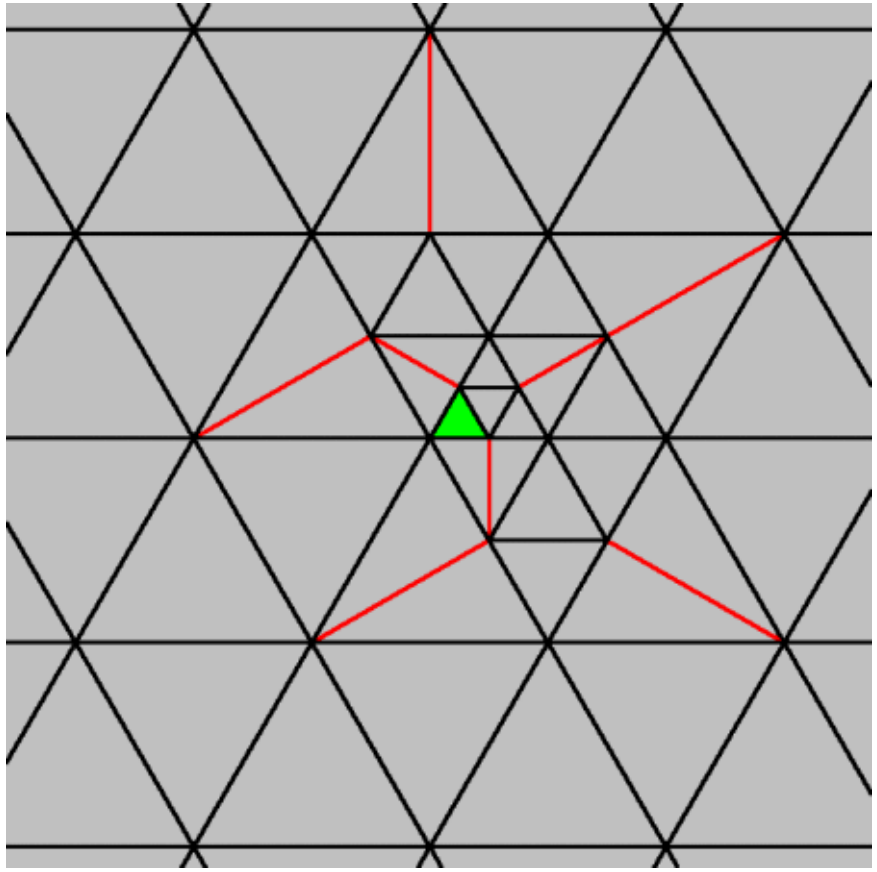
- Dyadic refinement (1-to-4 split)
  - Store quad-tree of triangles
  - Red-green-triangulation

- $\sqrt{3}$ - refinement (1-to-3 split)
  - Store mesh in flat data structure
  - Built-in consistency

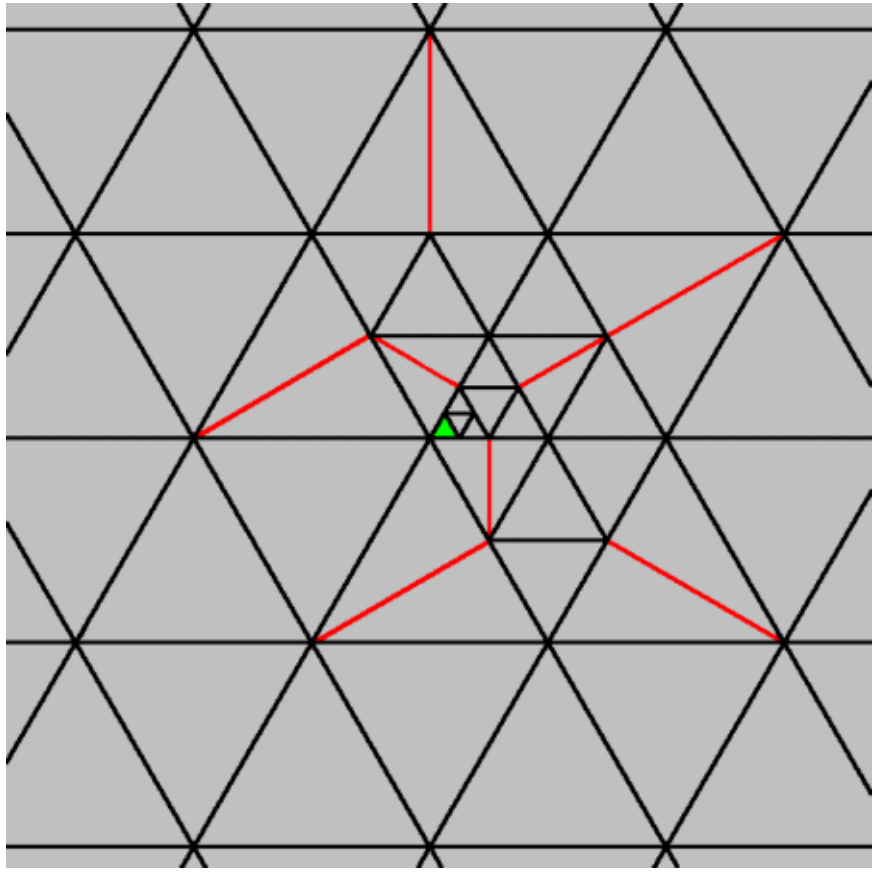
Red-Green Triangulation



Red-Green Triangulation

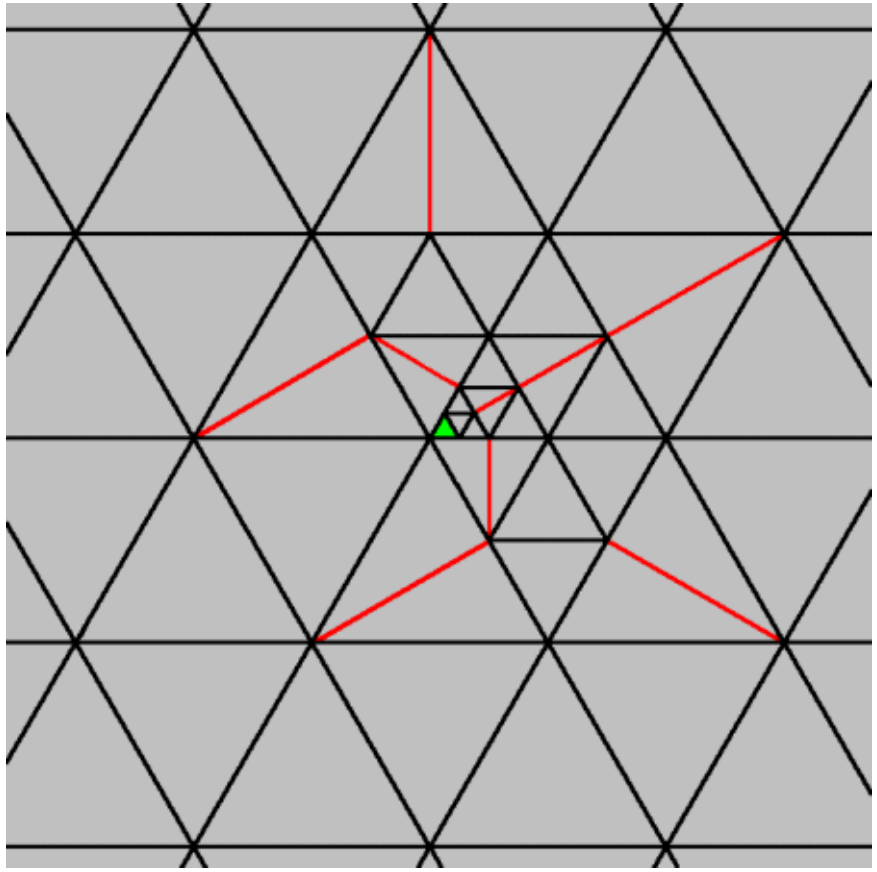


Red-Green Triangulation



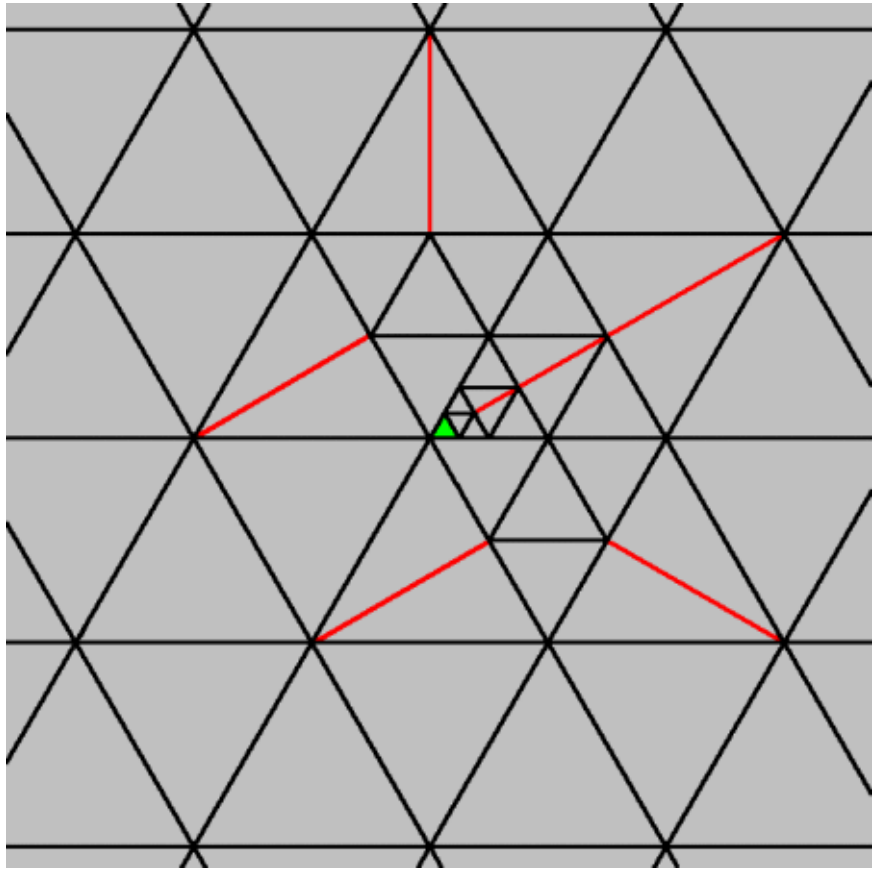
Green

Red-Green Triangulation



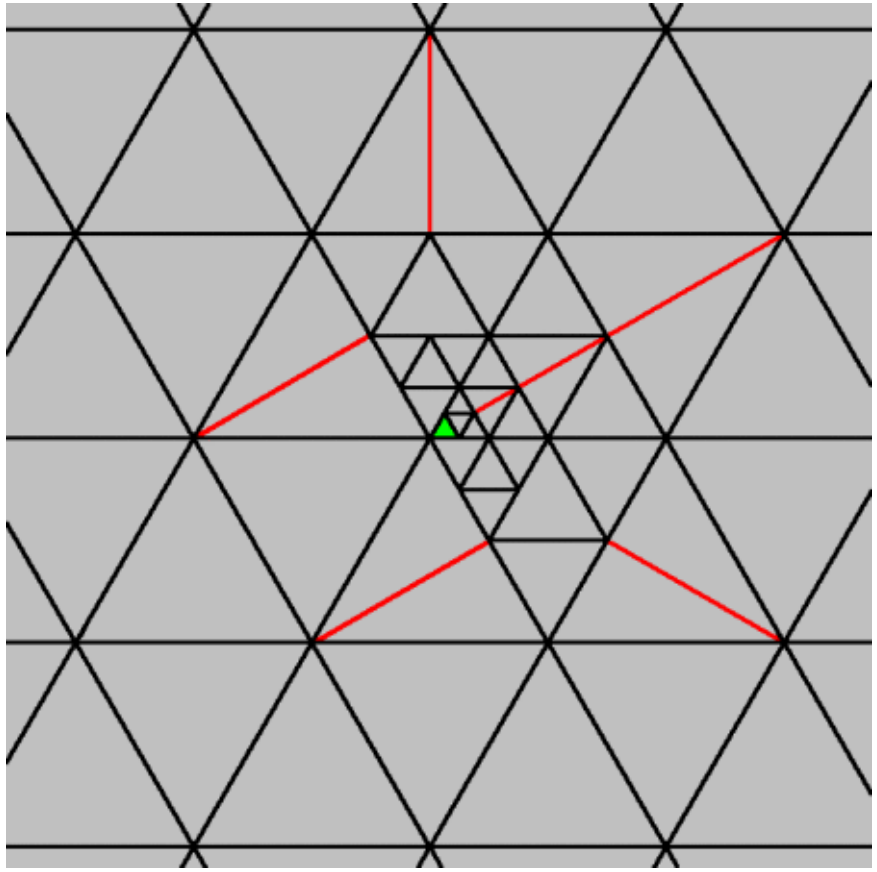
Green
Red

Red-Green Triangulation



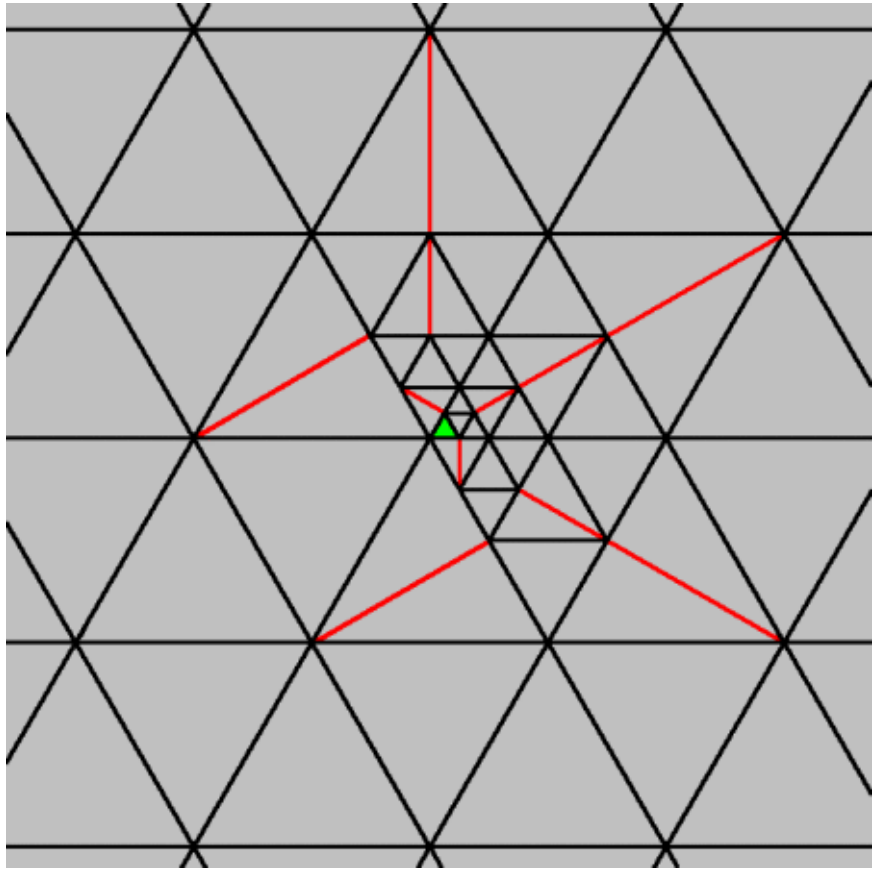
Green
Red
2x Undo

Red-Green Triangulation



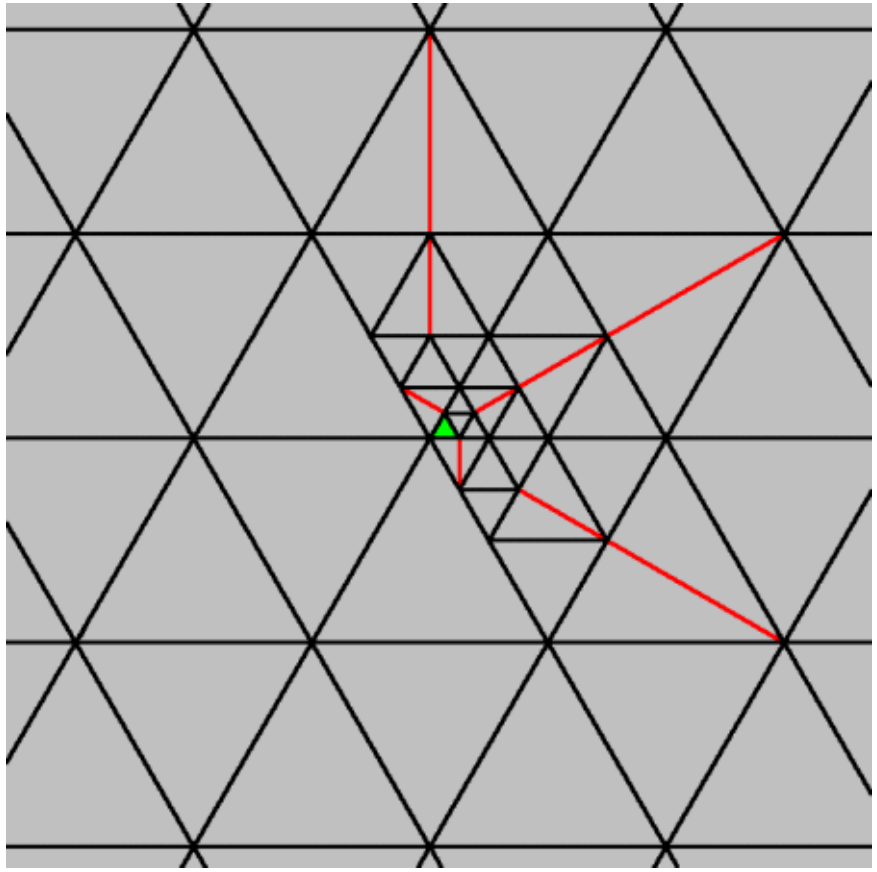
Green
Red
2x Undo
2x Green

Red-Green Triangulation



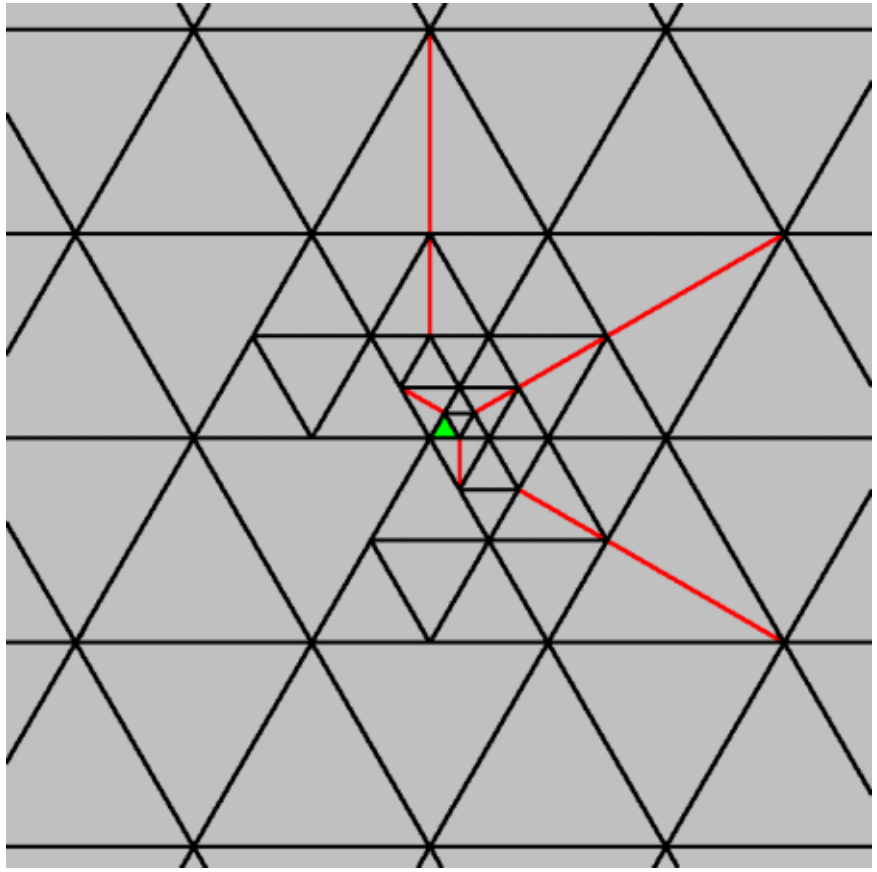
Green
Red
2x Undo
2x Green
4x Red

Red-Green Triangulation



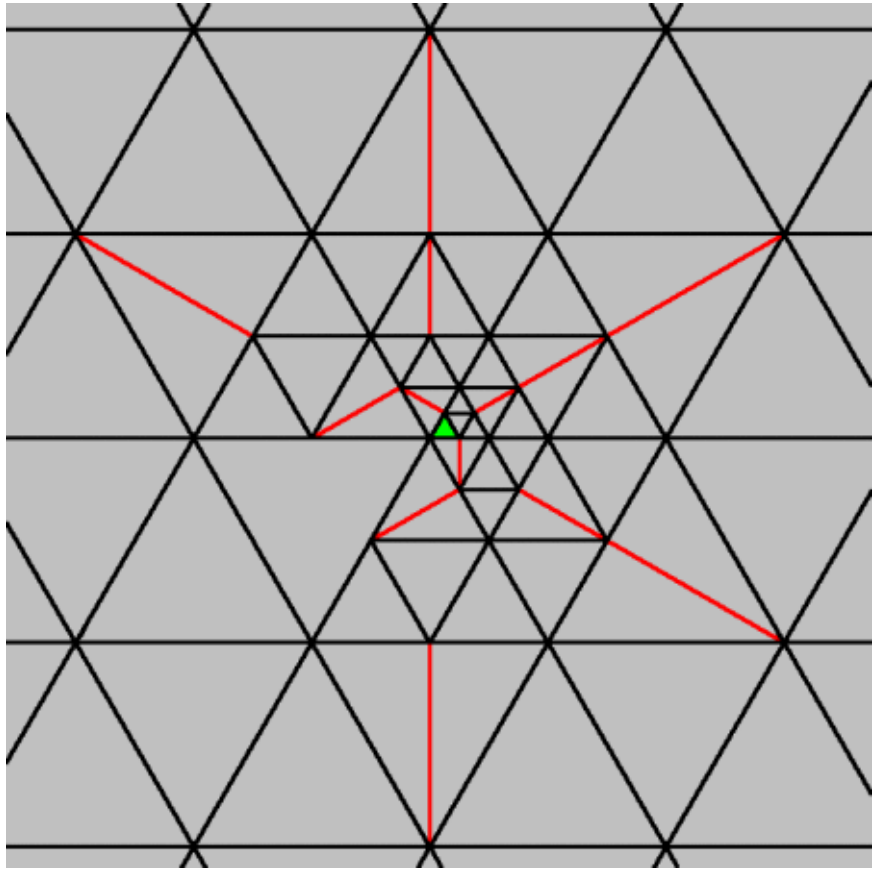
Green
Red
2x Undo
2x Green
4x Red
2x Undo

Red-Green Triangulation



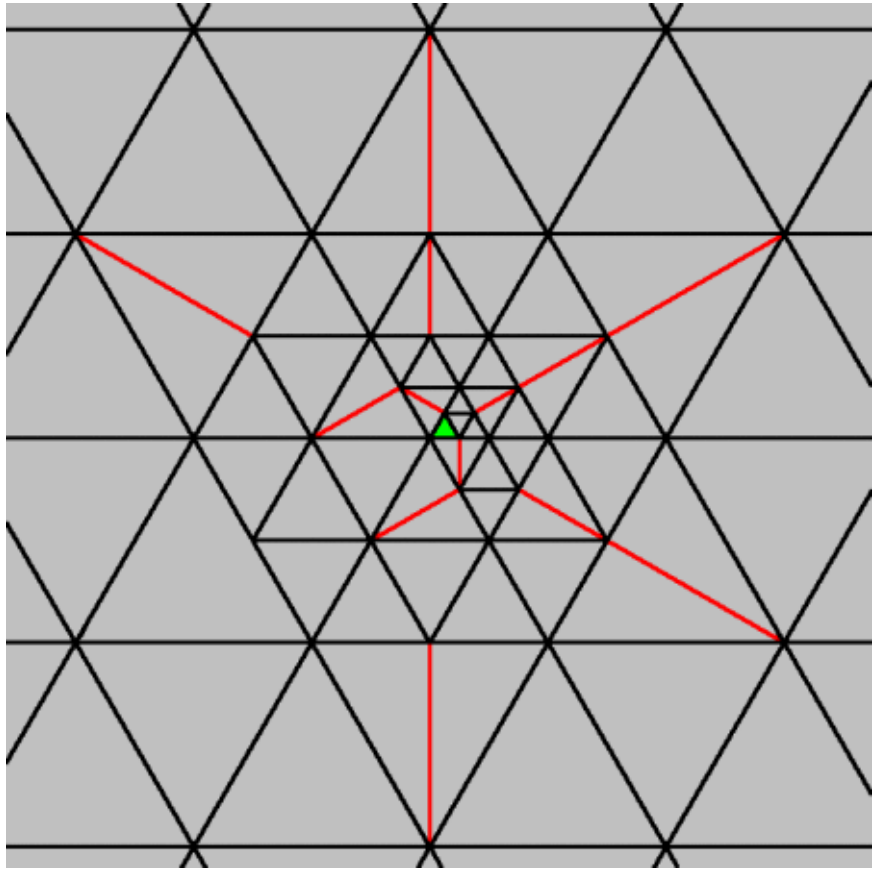
Green
Red
2x Undo
2x Green
4x Red
2x Undo
2x Green

Red-Green Triangulation



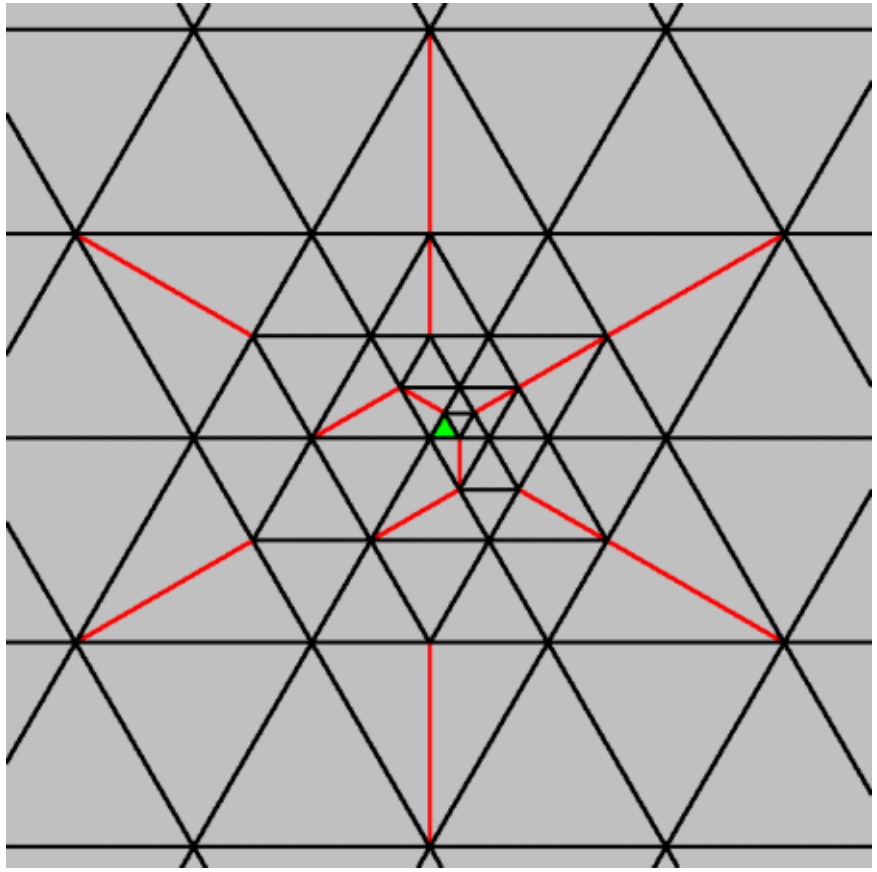
Green
Red
2x Undo
2x Green
4x Red
2x Undo
2x Green
4x Red

Red-Green Triangulation



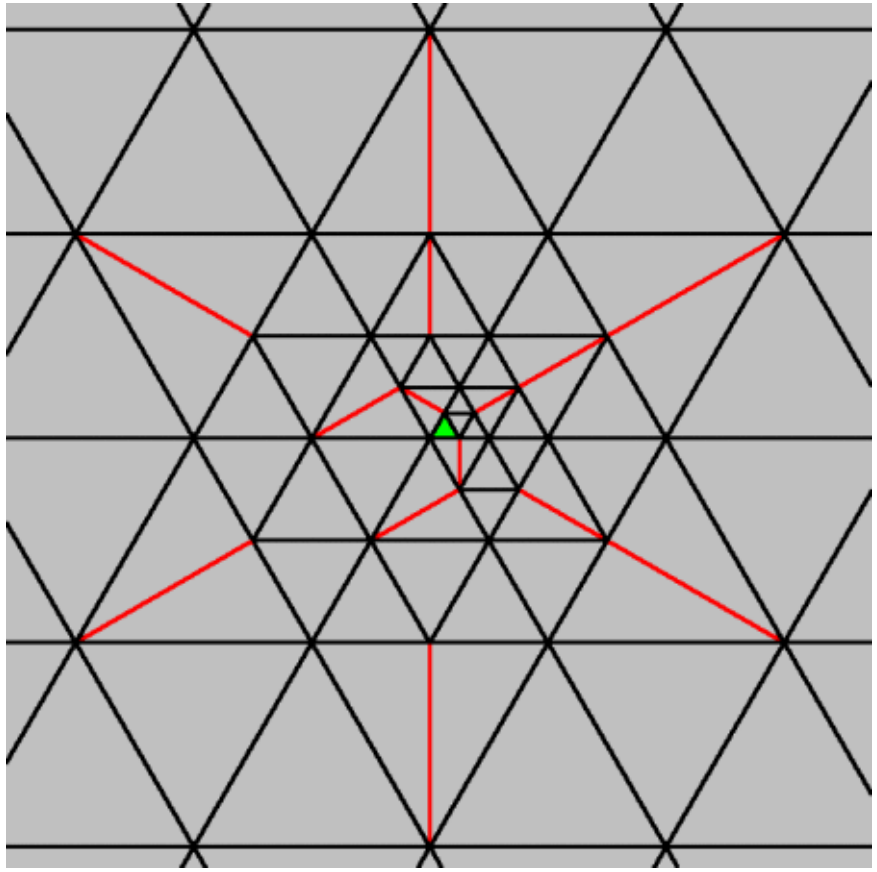
Green
Red
2x Undo
2x Green
4x Red
2x Undo
2x Green
4x Red
Green

Red-Green Triangulation



Green
Red
2x Undo
2x Green
4x Red
2x Undo
2x Green
4x Red
Green
Red

Red-Green Triangulation



Green

2x Green

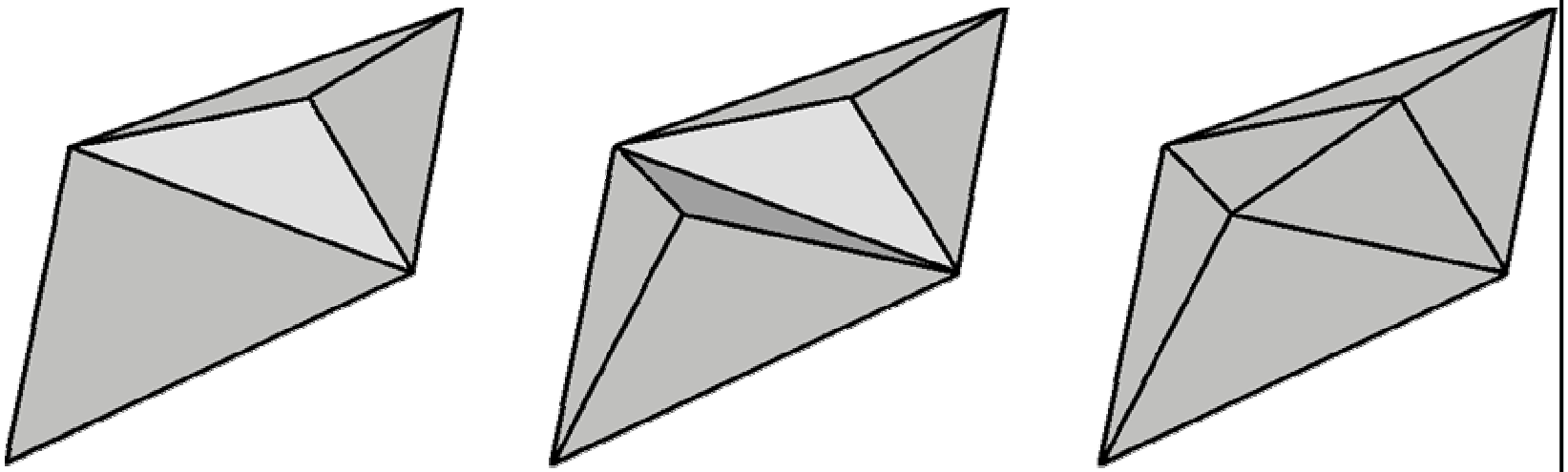
2x Green

Green
14x Red

Red-Green Triangulation

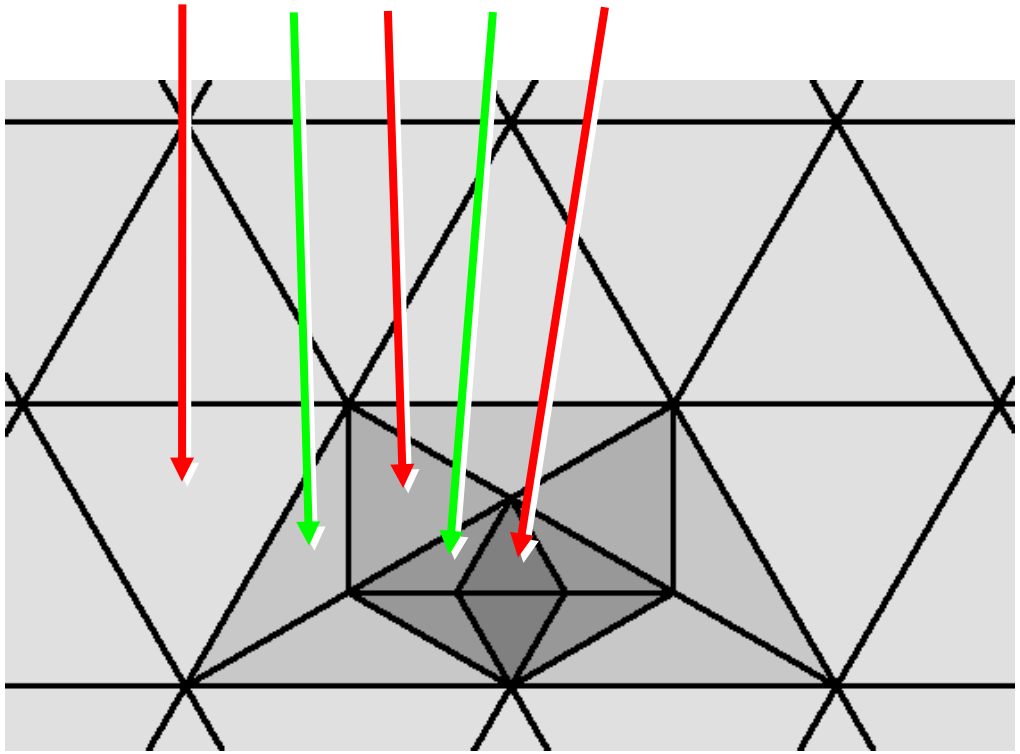
```
Split(T)
    for i = 1,2,3
        if level(T->neighbor[i]) < level(T)
            Split(T->neighbor[i])
    Quadrisect(T)
    Crack fixing ...
```


$\sqrt{3}$ - Refinement

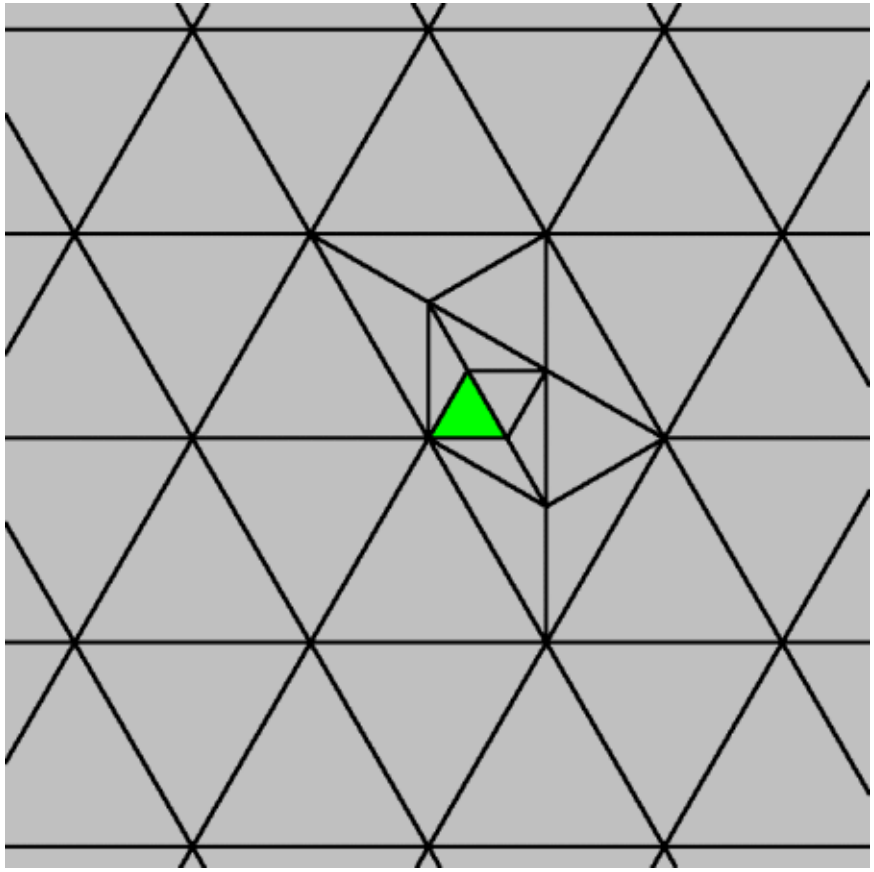


$\sqrt{3}$ - Refinement

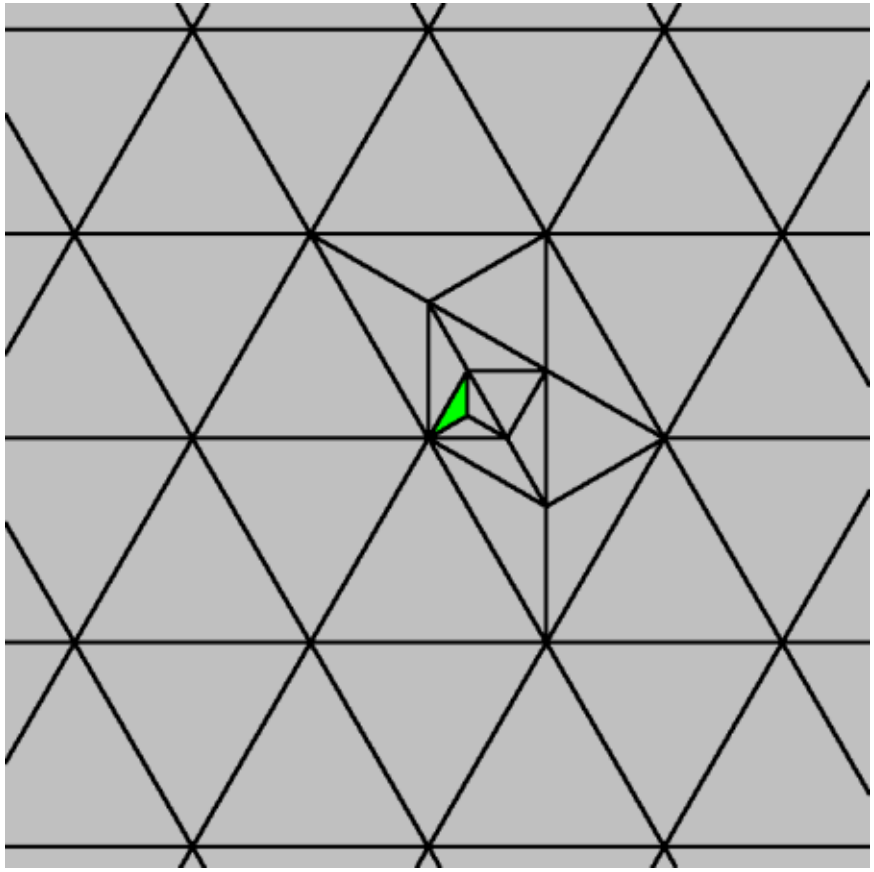
Face generation: 0, 1, 2, 3, 4, … even / odd



$\sqrt{3}$ - Refinement

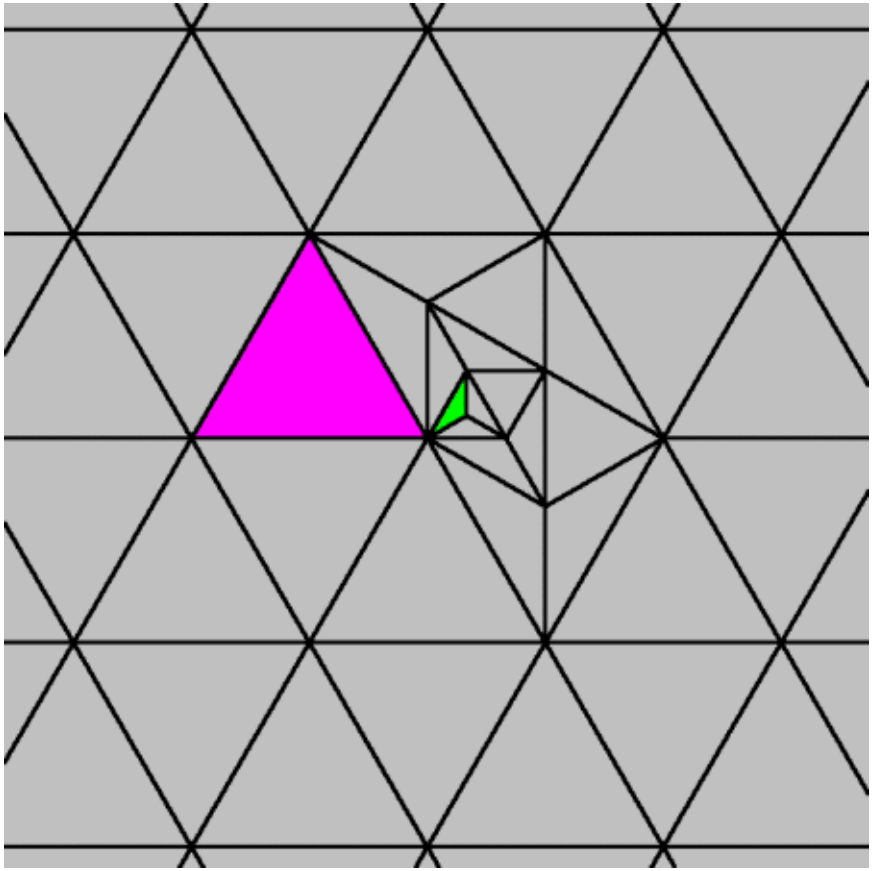


$\sqrt{3}$ - Refinement

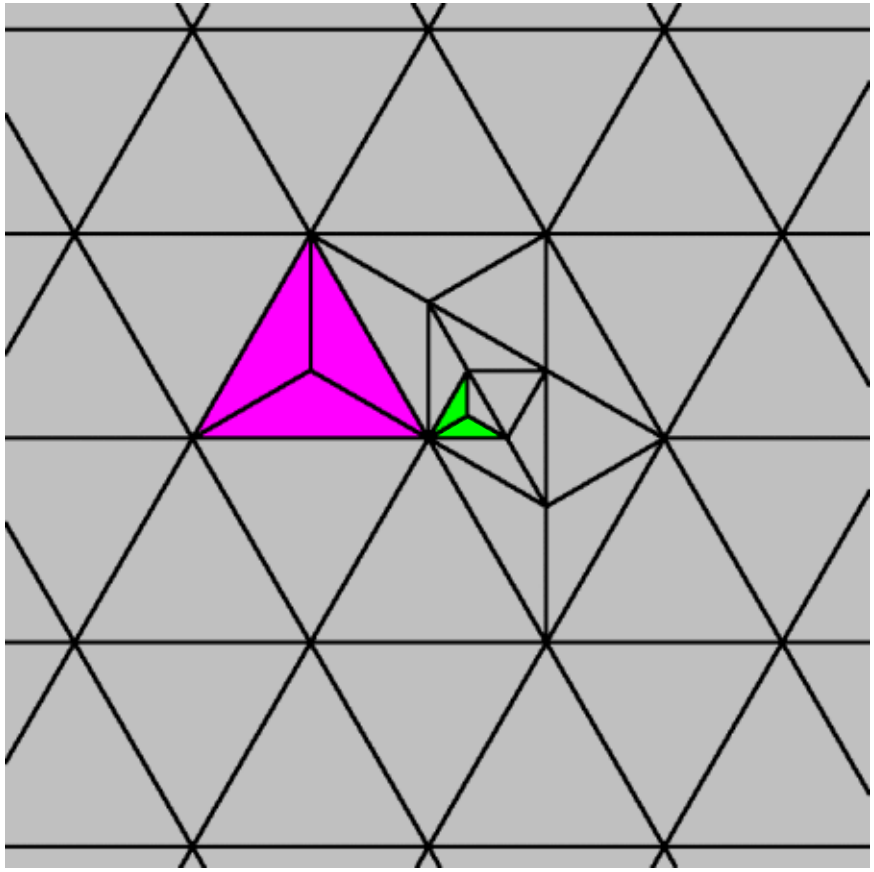


Center

$\sqrt{3}$ - Refinement

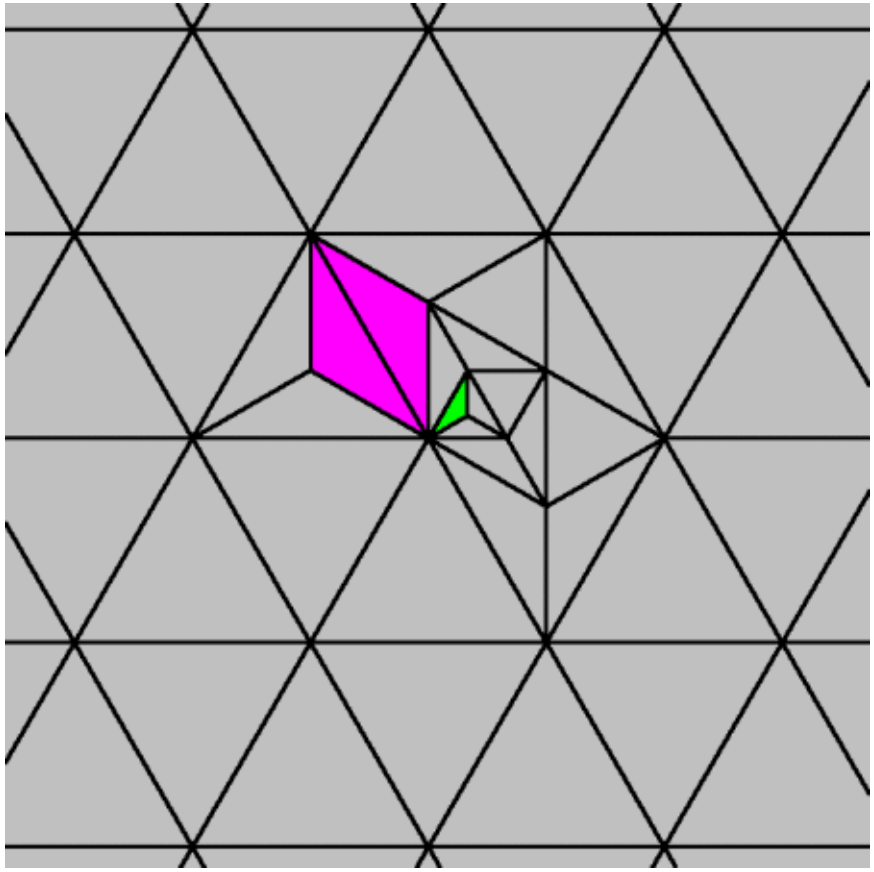


$\sqrt{3}$ - Refinement



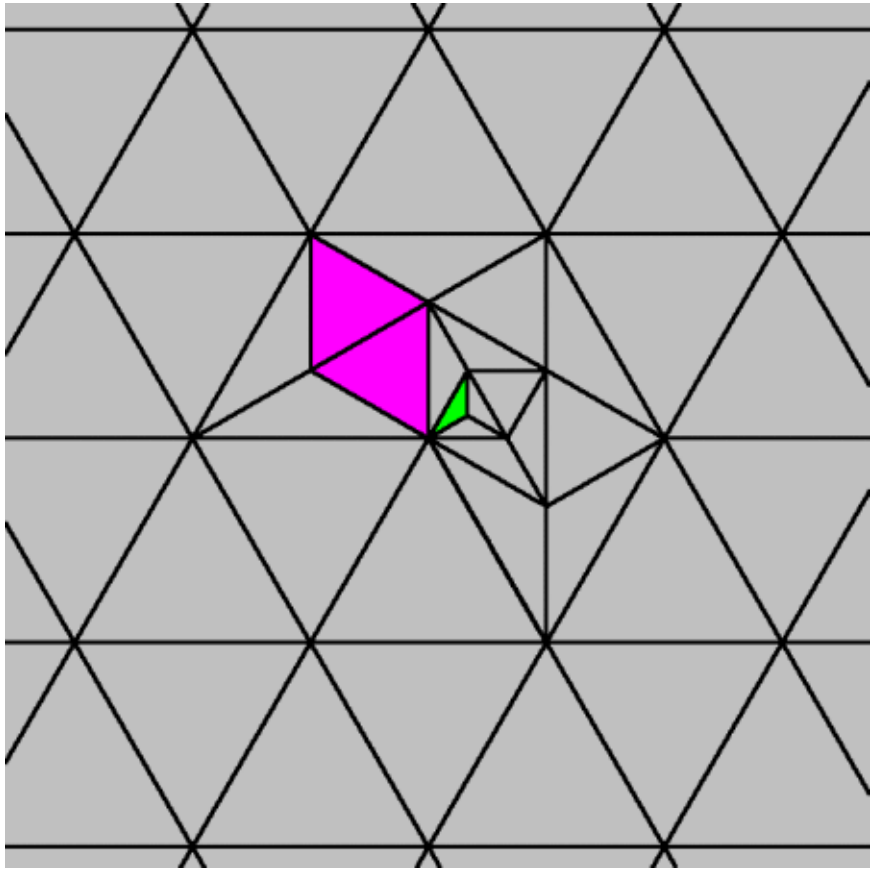
Center

$\sqrt{3}$ - Refinement



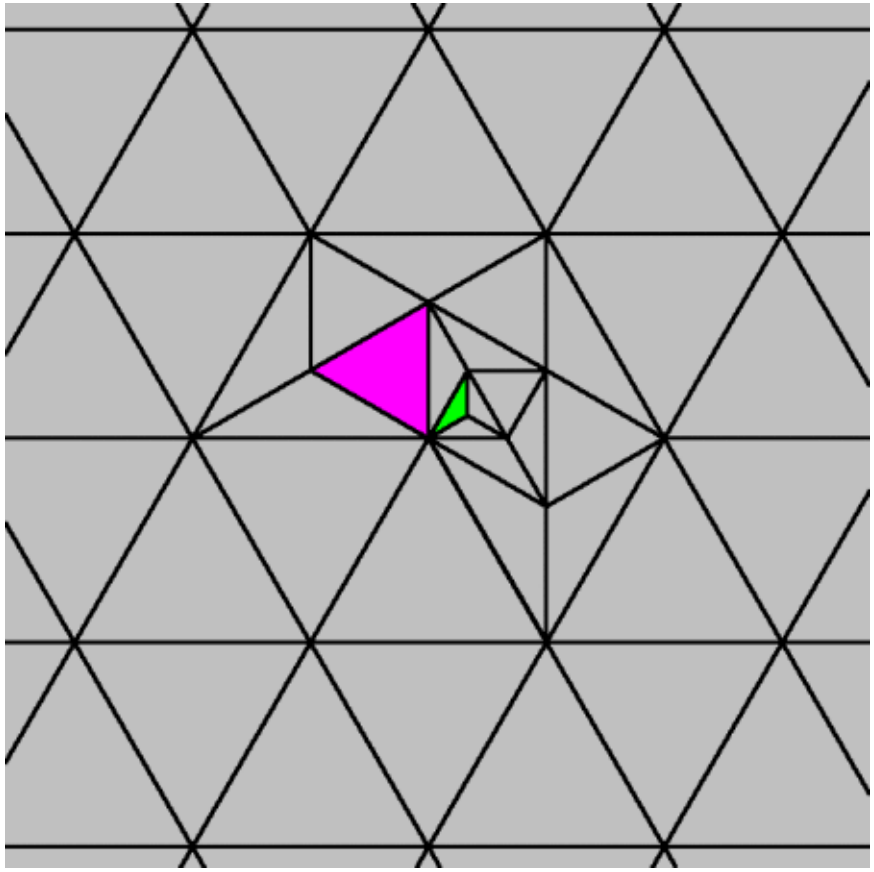
Center

$\sqrt{3}$ - Refinement



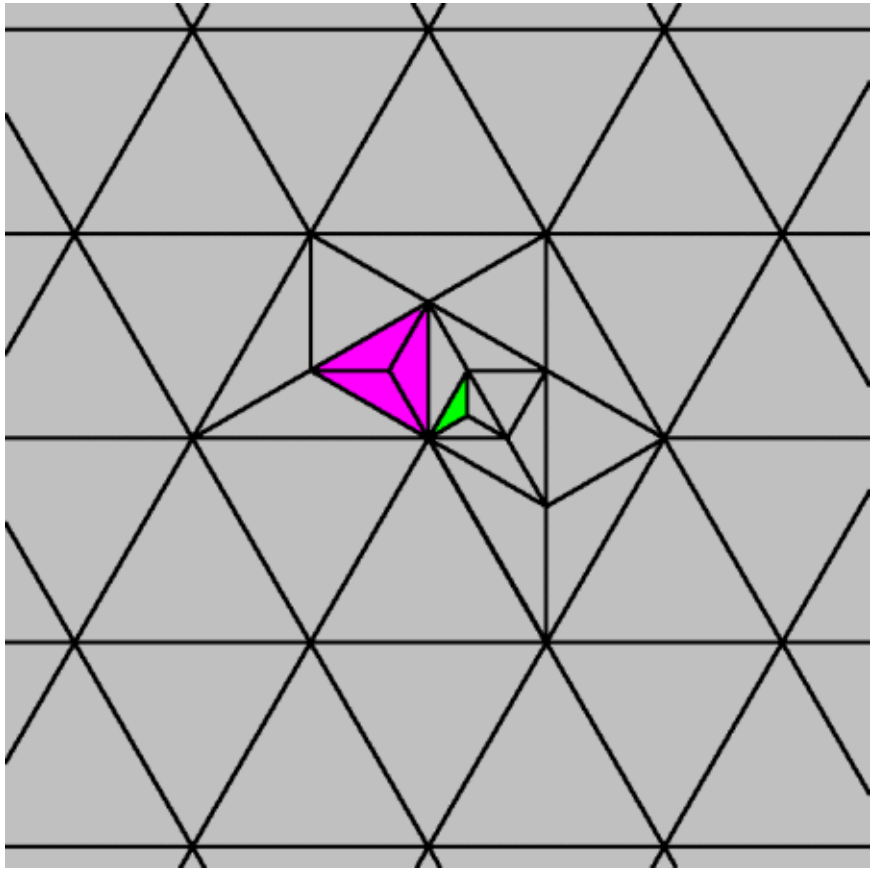
Center
Flip

$\sqrt{3}$ - Refinement



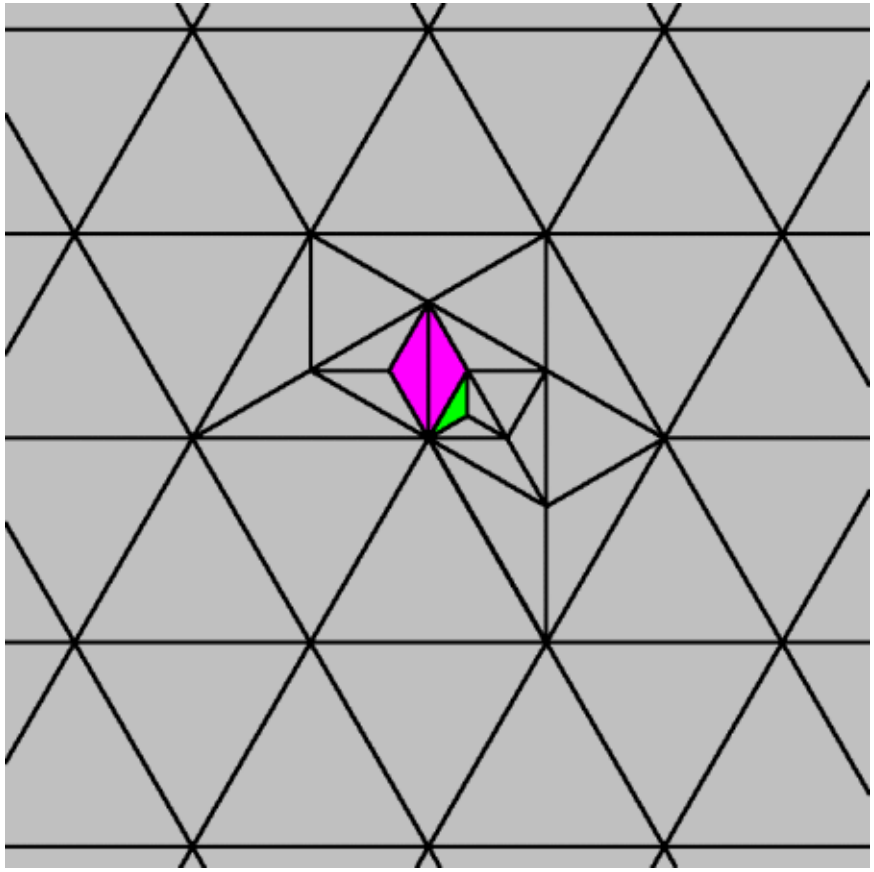
Center
Flip

$\sqrt{3}$ - Refinement



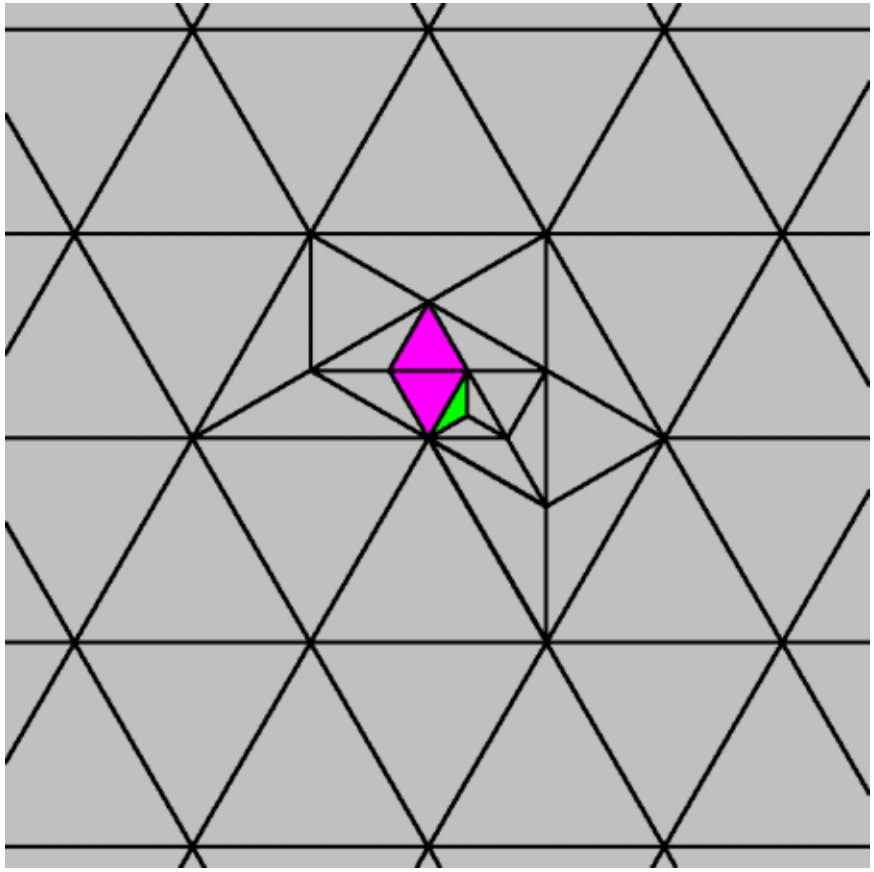
Center
Flip
Center

$\sqrt{3}$ - Refinement



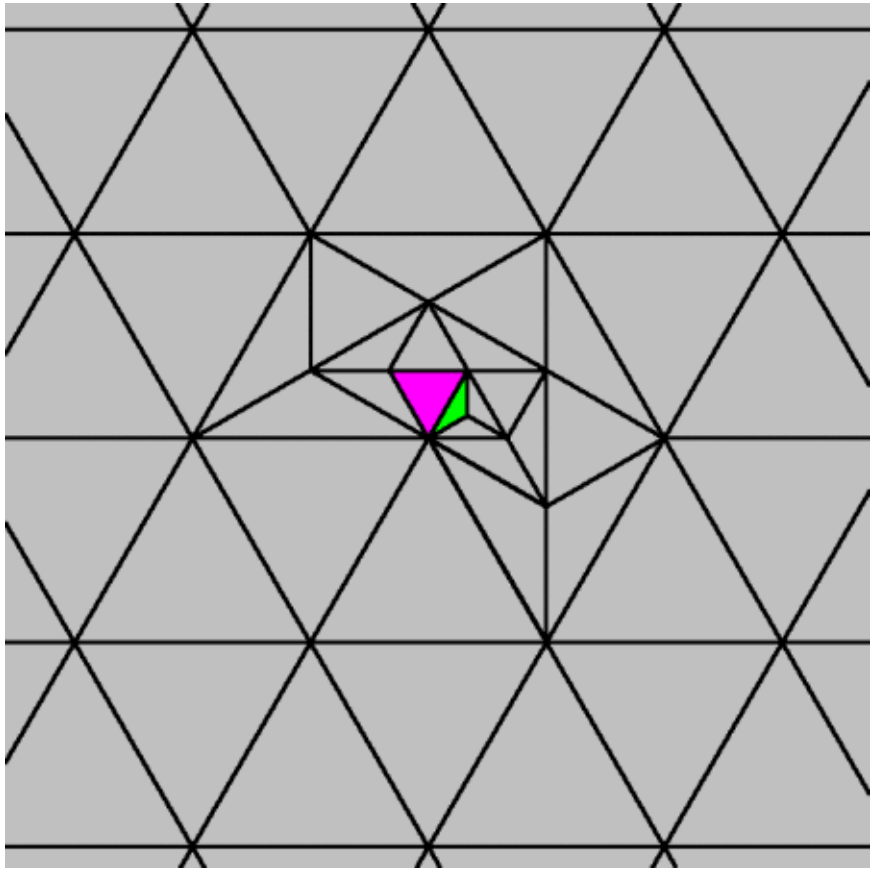
Center
Flip
Center

$\sqrt{3}$ - Refinement



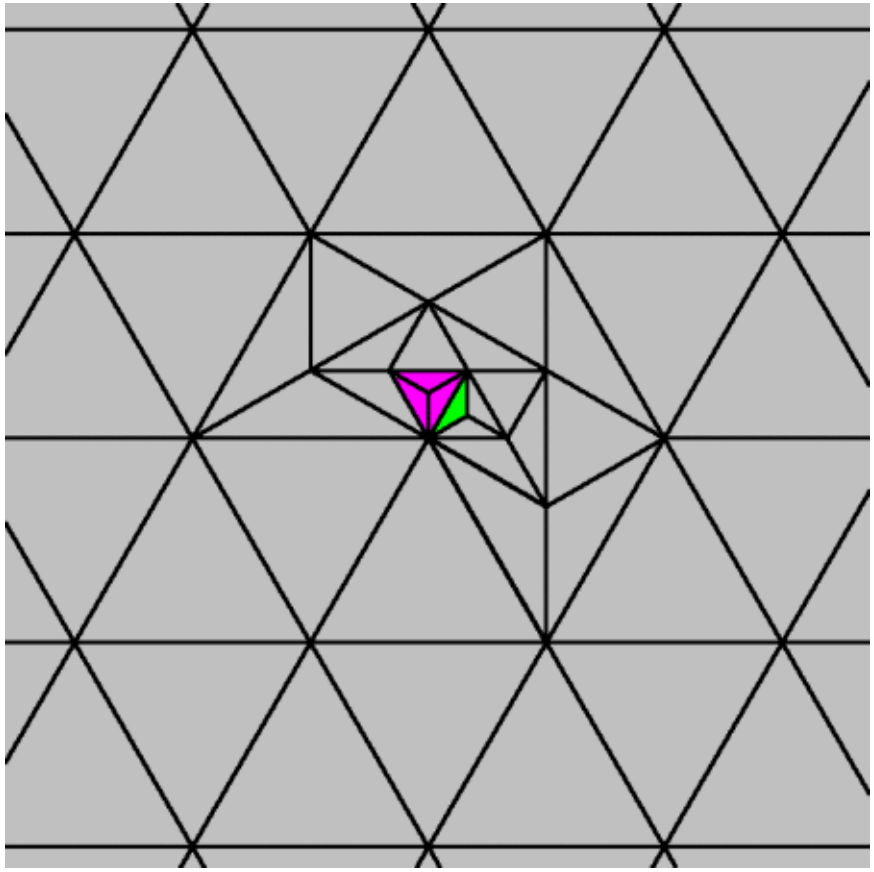
Center
Flip
Center
Flip

$\sqrt{3}$ - Refinement



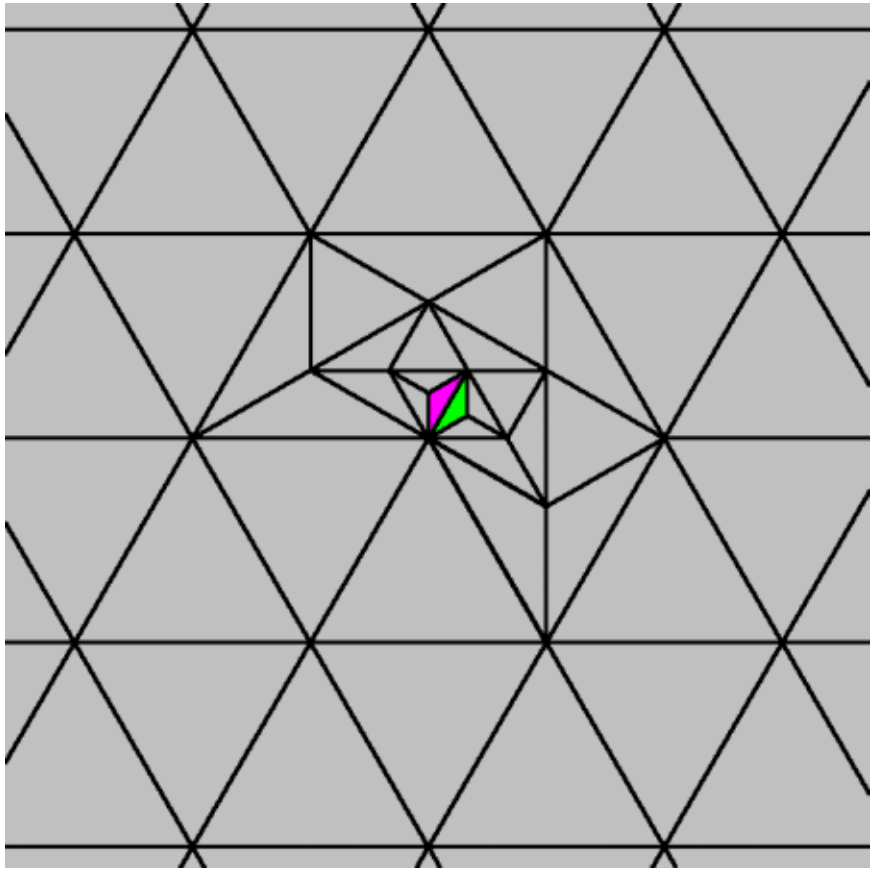
Center
Flip
Center
Flip

$\sqrt{3}$ - Refinement



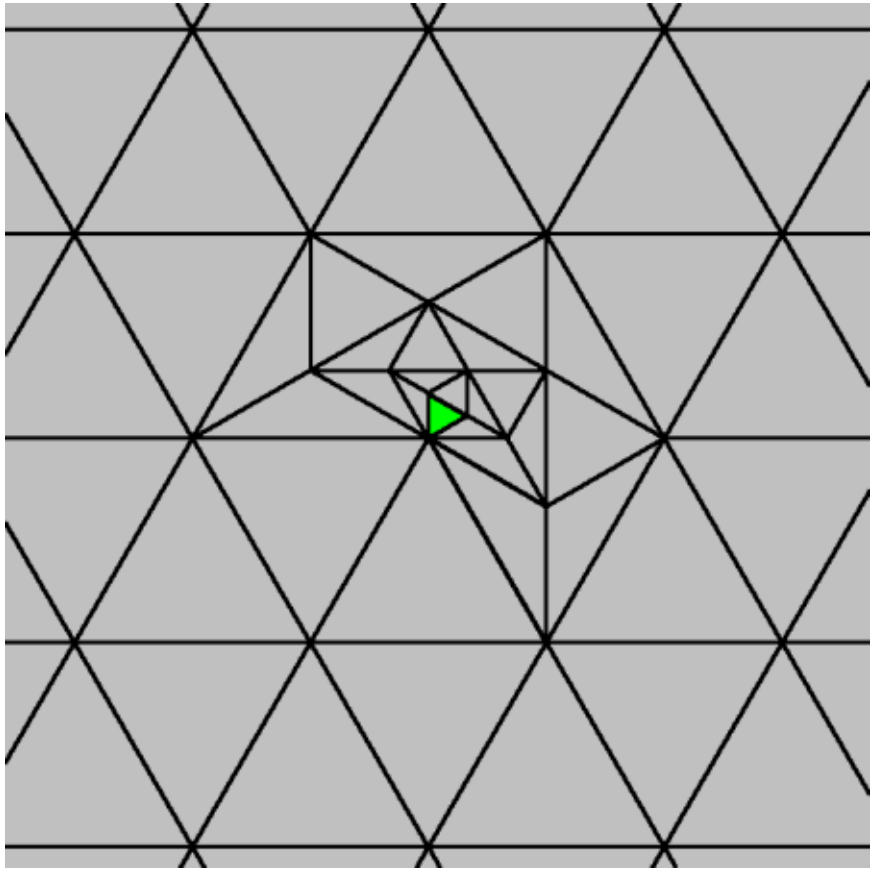
Center
Flip
Center
Flip
Center

$\sqrt{3}$ - Refinement



Center
Flip
Center
Flip
Center

$\sqrt{3}$ - Refinement

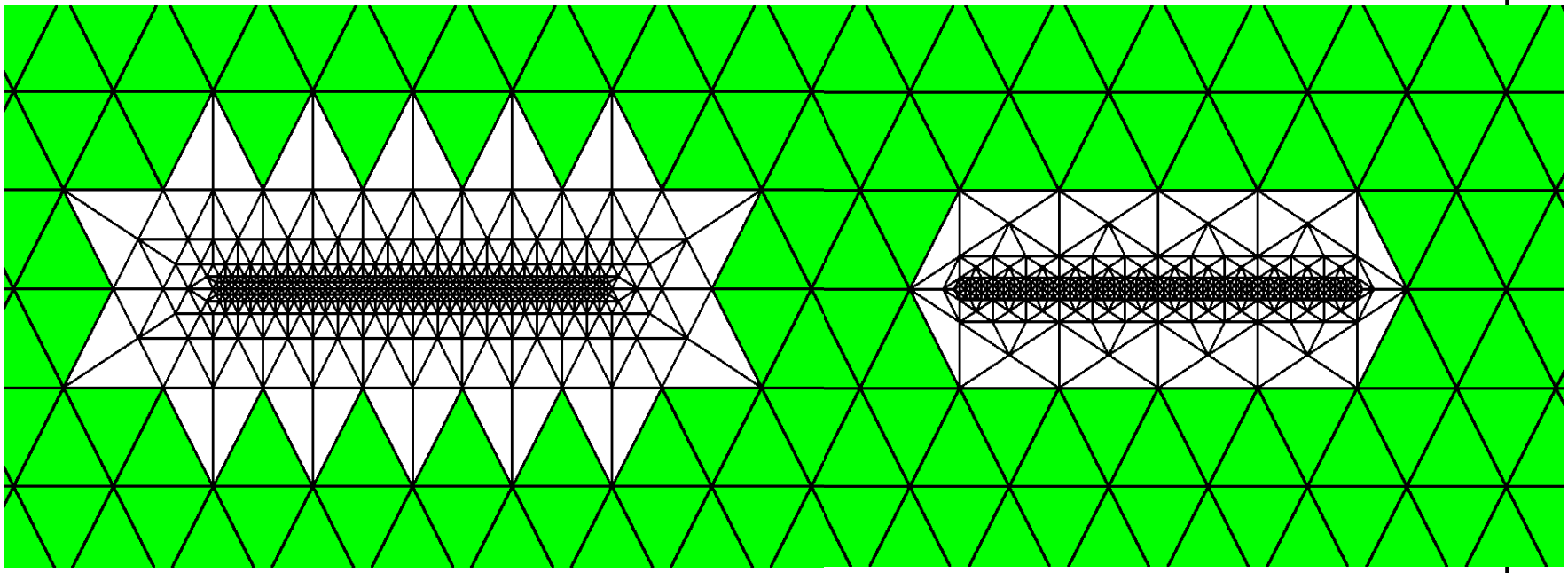


Center
Flip
Center
Flip
Center
Flip

$\sqrt{3}$ - Refinement

```
Split(T)
    if generation(T) = even
        center-split(T)
        flip edges (if possible)
    else
        if generation(T->opposite) = odd
            Split(T->opposite)
        Split(T->opposite)
```


$\sqrt{3}$ - Refinement



Surface Subdivision

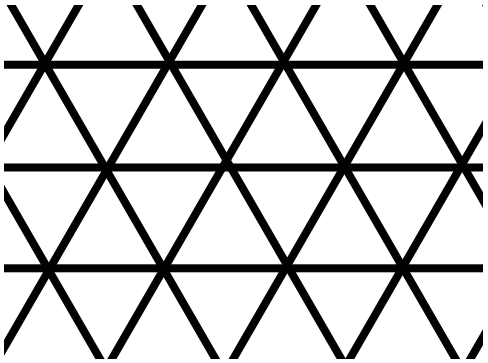
- Splitting
  - Primal / Dual
  - Triangle / Quad - based
- Smoothing
  - Polynomial
  - Non-polynomial
  - Interpolatory ...

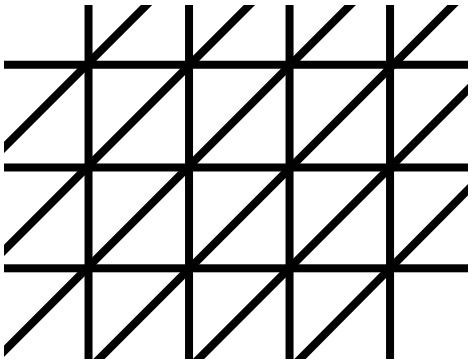
Polynomial Subdivision

- Univariate: uniform B-splines
  - $[\alpha_j] = [1,1]$
  - $[\alpha_j] = [1,2,1] / 2$
  - $[\alpha_j] = [1,3,3,1] / 4$
  - $[\alpha_j] = [1,4,6,4,1] / 8$
  - ...
- Masks obtained by ***averaging***

Polynomial Subdivision

- Bivariate: uniform box-splines
- Masks obtained by averaging
  - Many possible directions ...





Polynomial Subdivision

- Bivariate: uniform box-splines
- Masks obtained by averaging
  - Many possible directions ...

$$[\alpha_{ij}] = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Polynomial Subdivision

- Bivariate: uniform box-splines
- Masks obtained by averaging
  - Many possible directions ...

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix} / 2$$

Polynomial Subdivision

- Bivariate: uniform box-splines
- Masks obtained by averaging
  - Many possible directions ...

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 1 & 1 \end{pmatrix} / 2$$

Polynomial Subdivision

- Bivariate: uniform box-splines
- Masks obtained by averaging
  - Many possible directions ...

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix} / 2$$

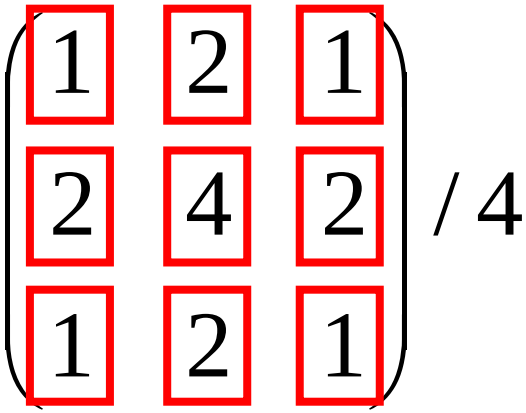
Polynomial Subdivision

- Bivariate: uniform box-splines
- Masks obtained by averaging
  - Many possible directions ...

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} / 4$$

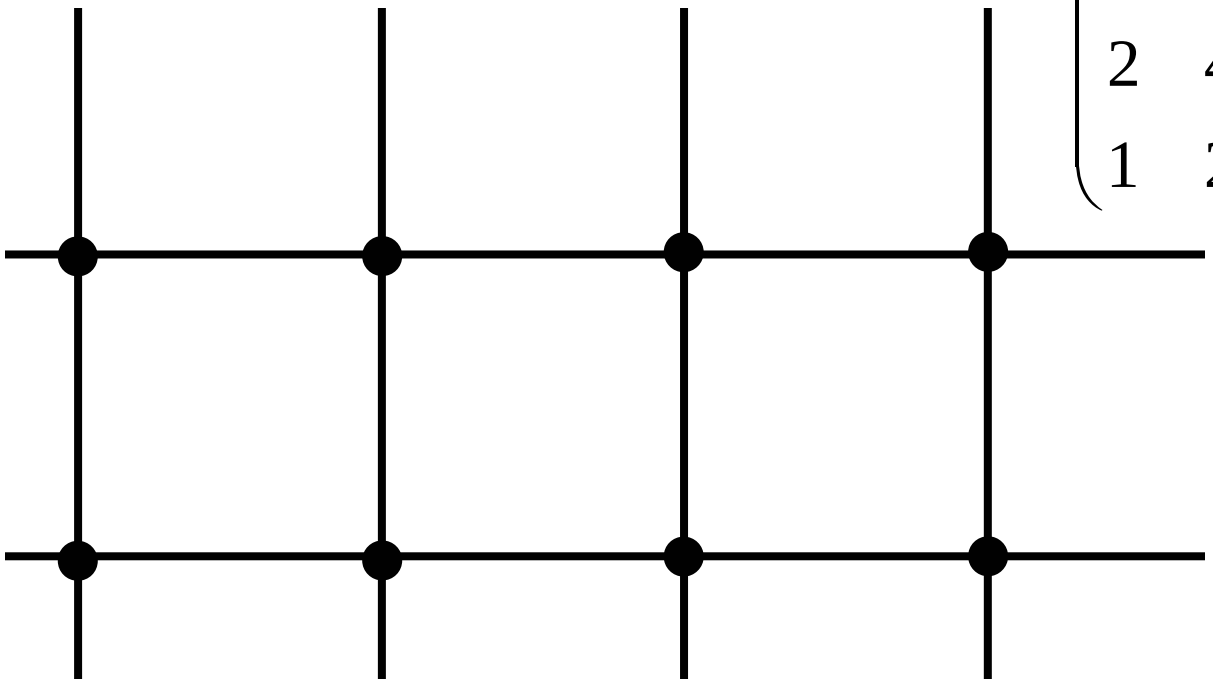
Smoothing Rules

- Subdivision masks encodes 4 rules (depending on index parities)

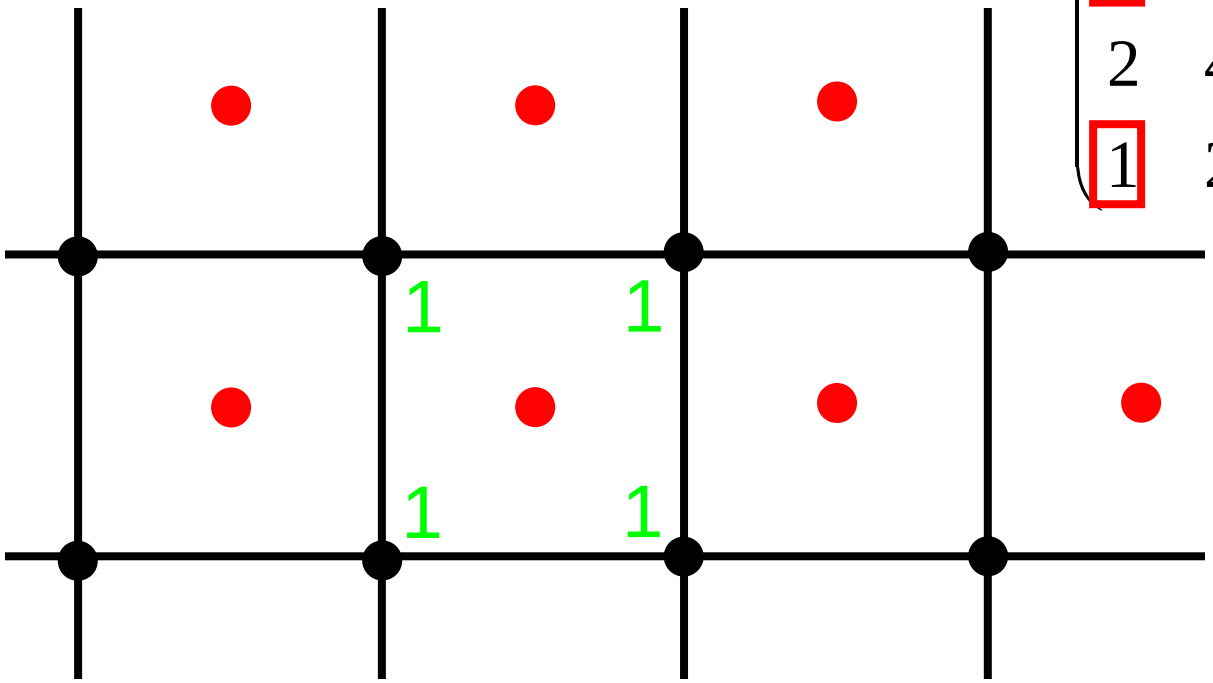


Smoothing Rules

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

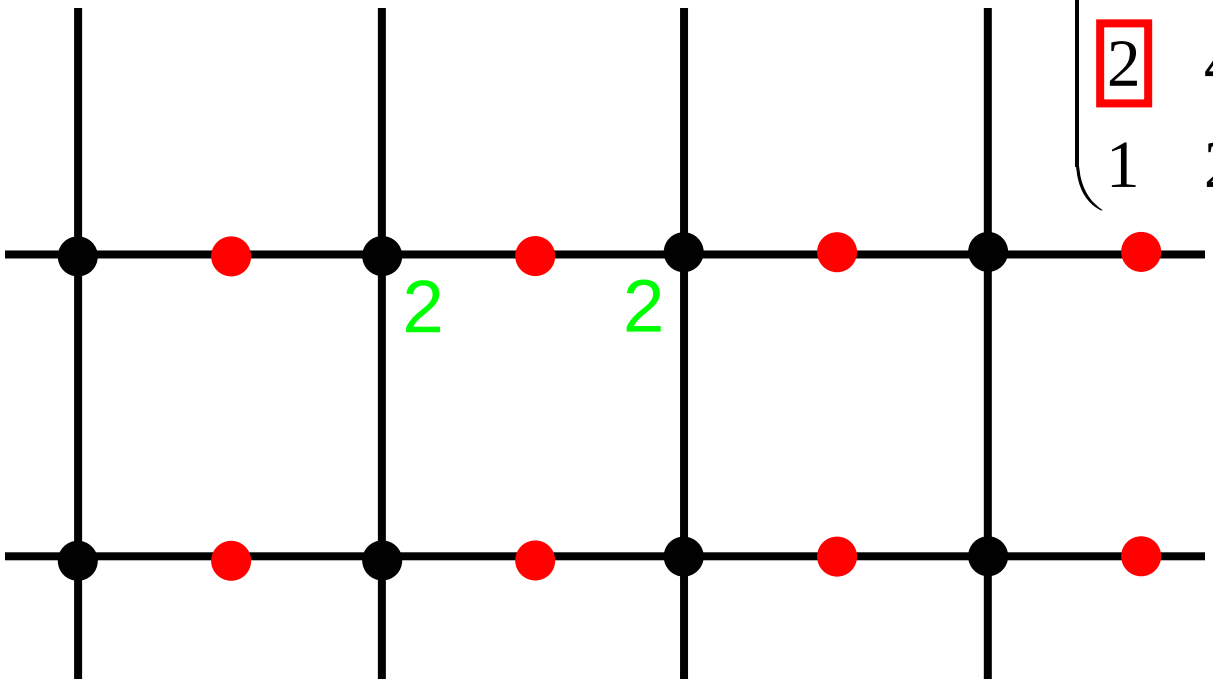


Smoothing Rules



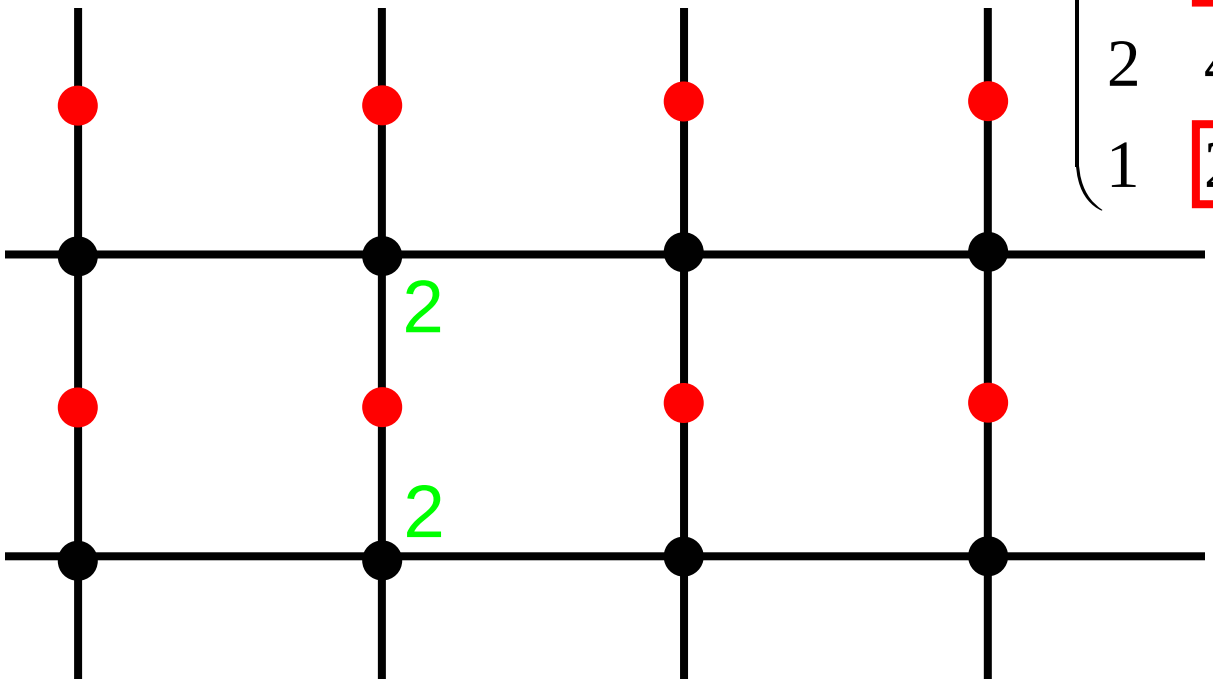
$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Smoothing Rules



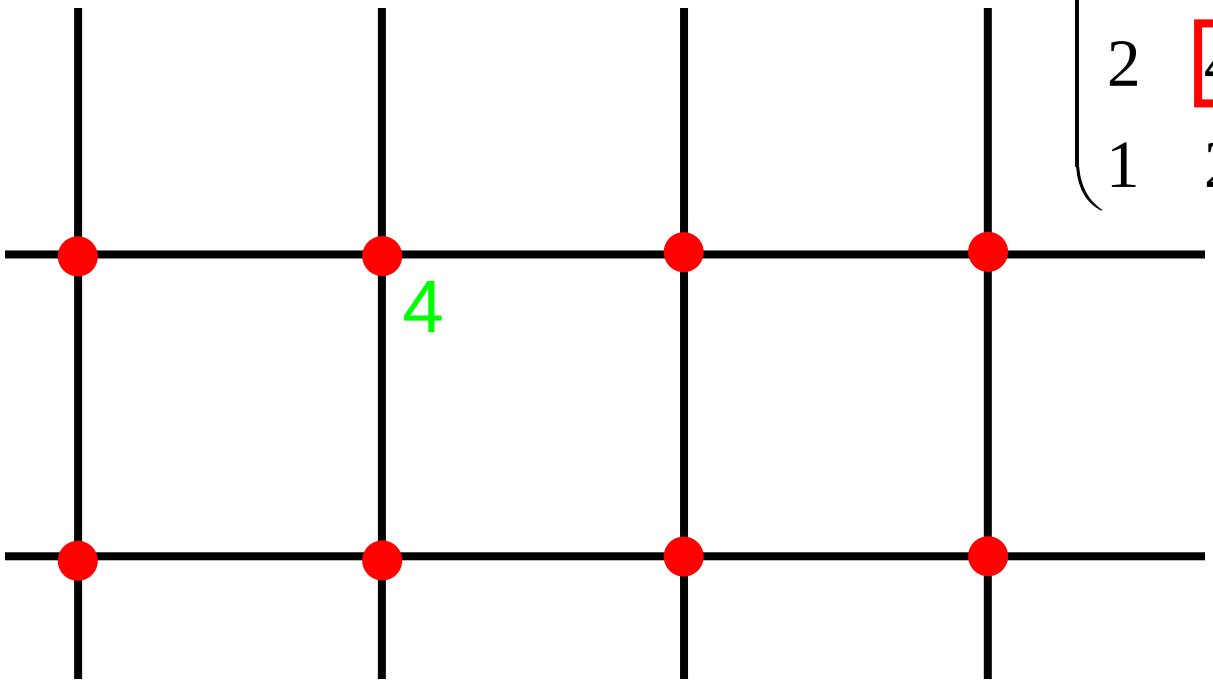
$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Smoothing Rules



$$\begin{pmatrix} 1 & \boxed{2} & 1 \\ 2 & 4 & 2 \\ 1 & \boxed{2} & 1 \end{pmatrix}$$

Smoothing Rules

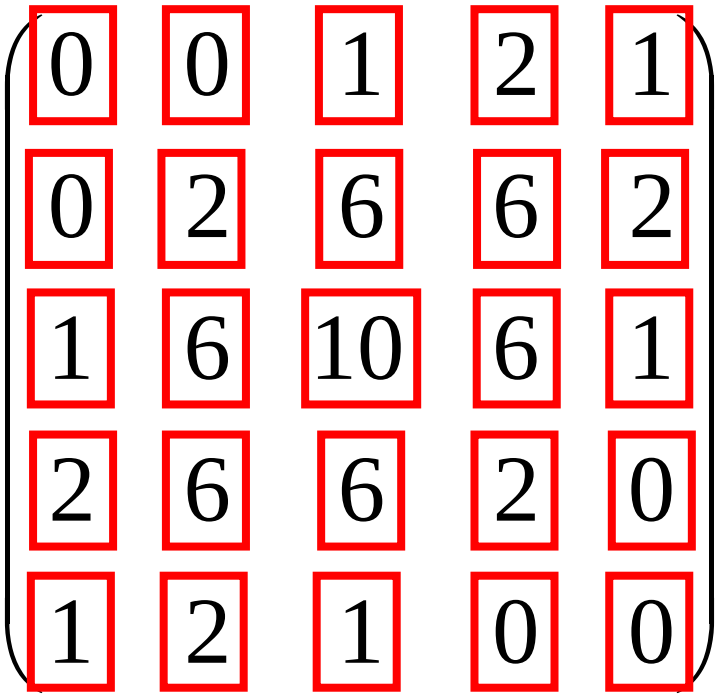


$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

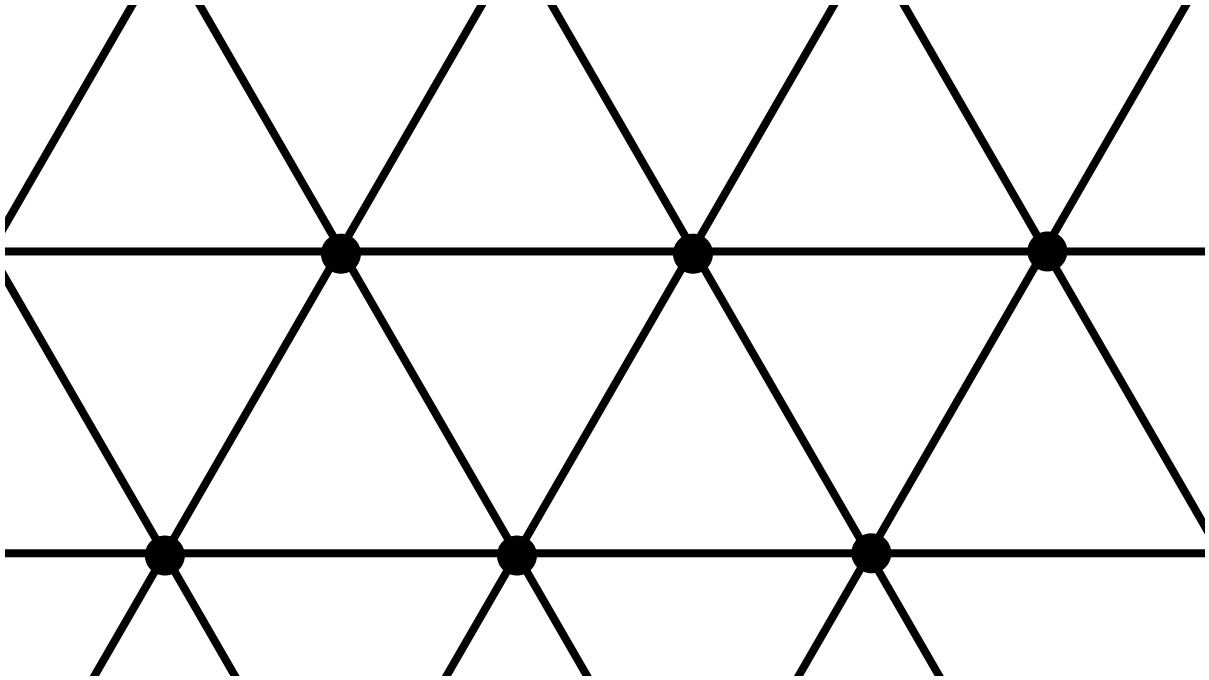
M_{222} Subdivision

$$(4) \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}^2 \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}^2 \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix}^2$$

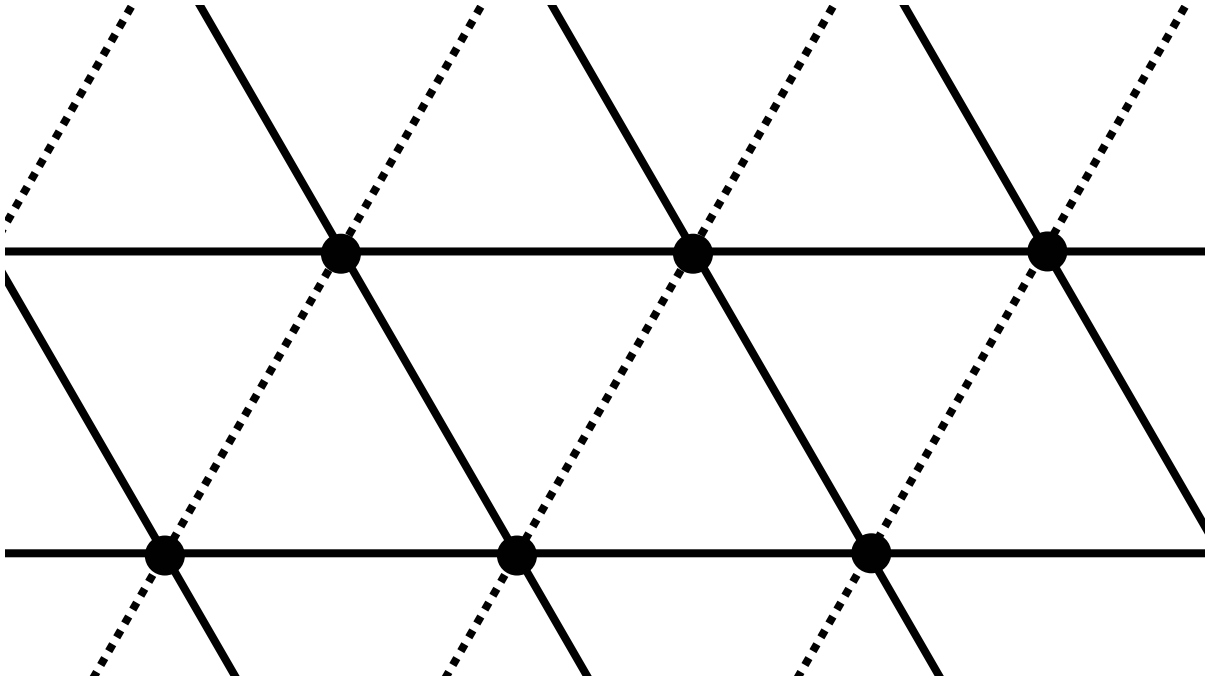
M_{222} Subdivision



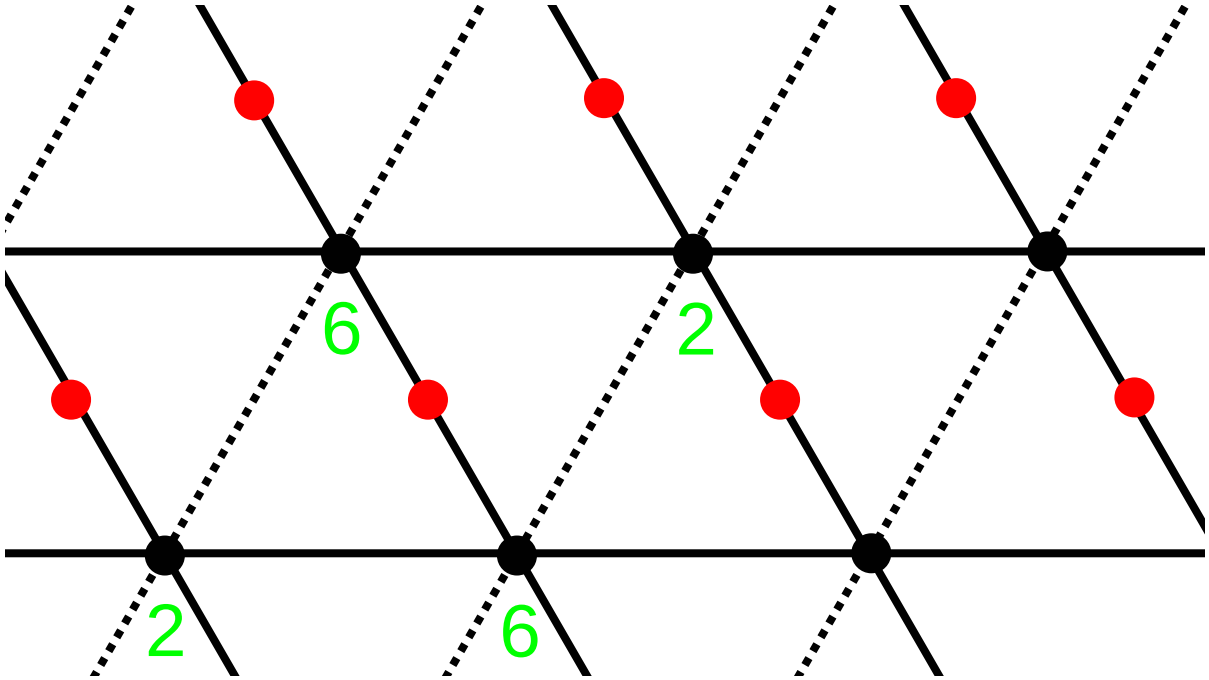
M_{222} Subdivision



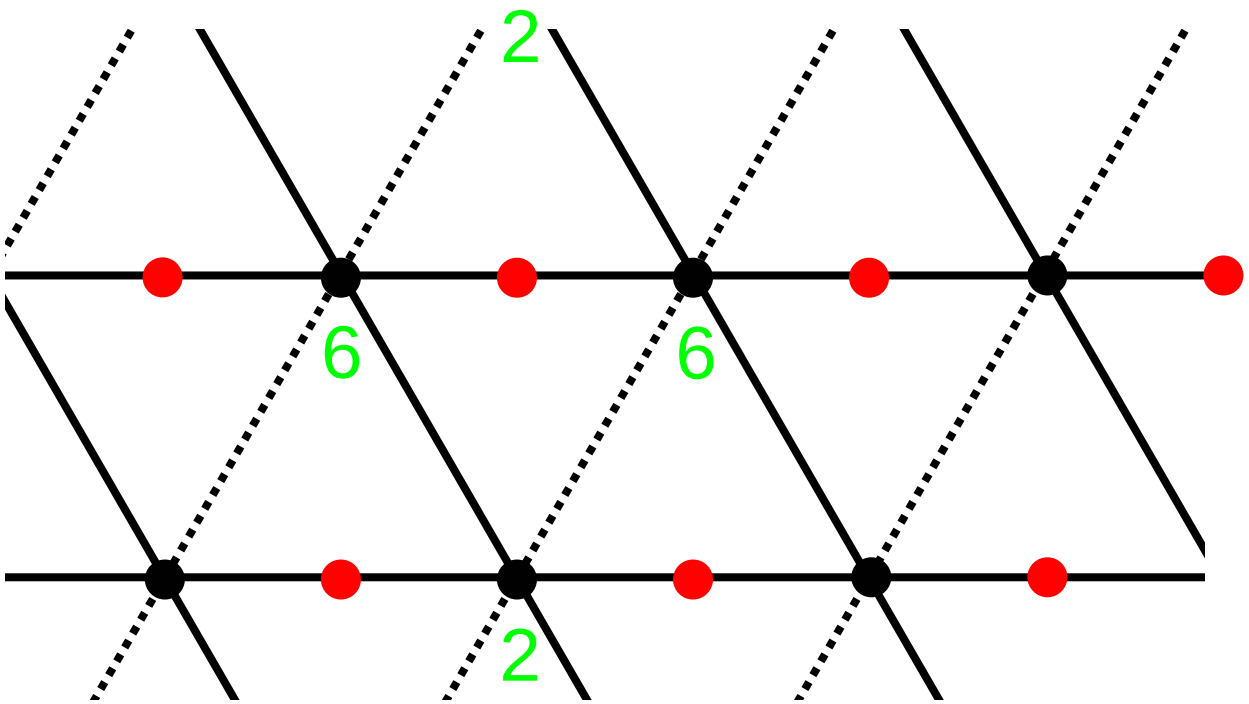
M_{222} Subdivision



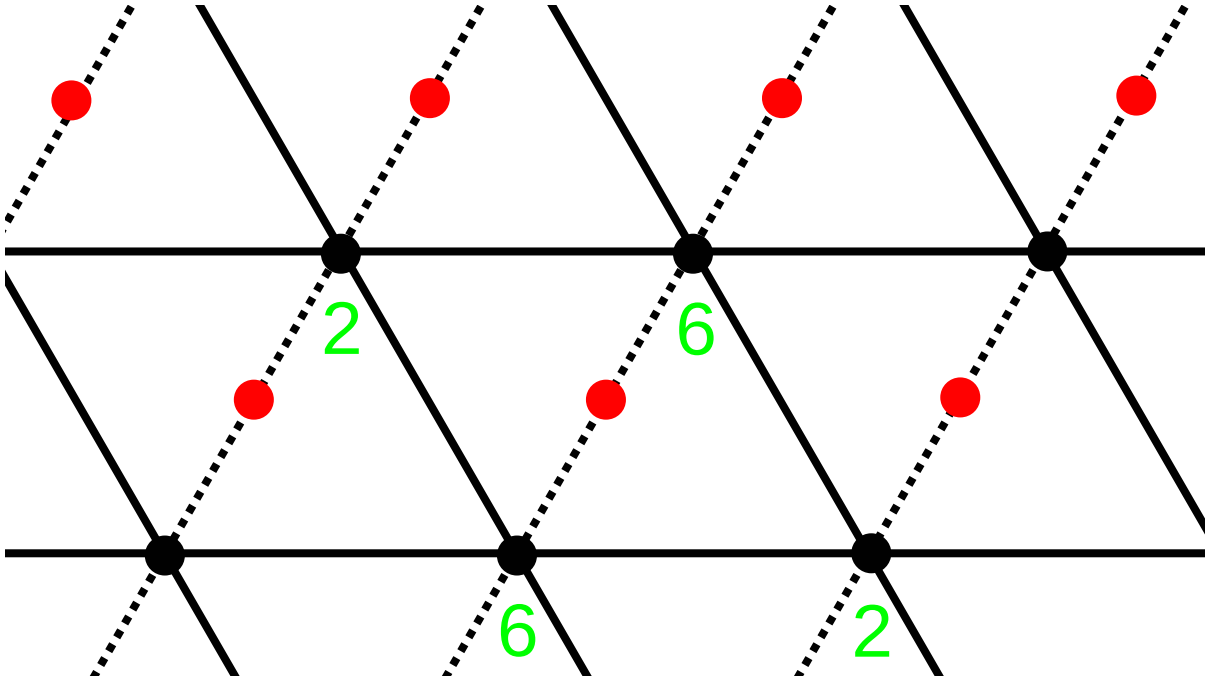
M_{222} Subdivision



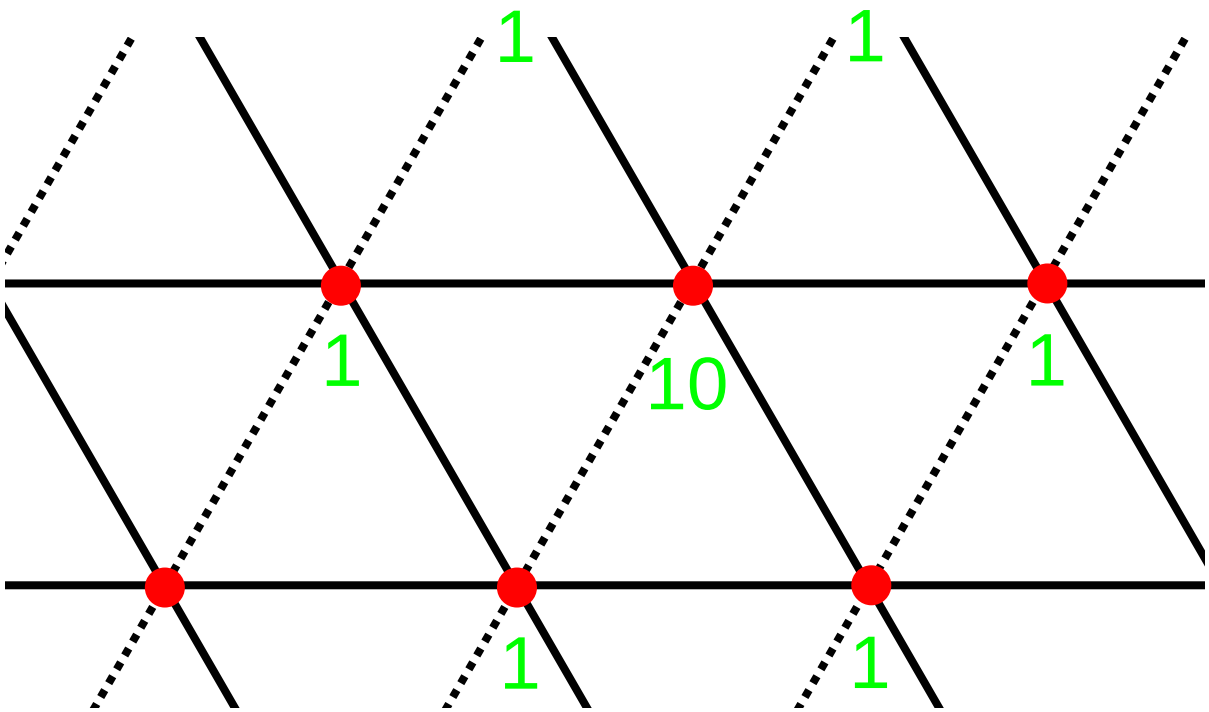
M_{222} Subdivision



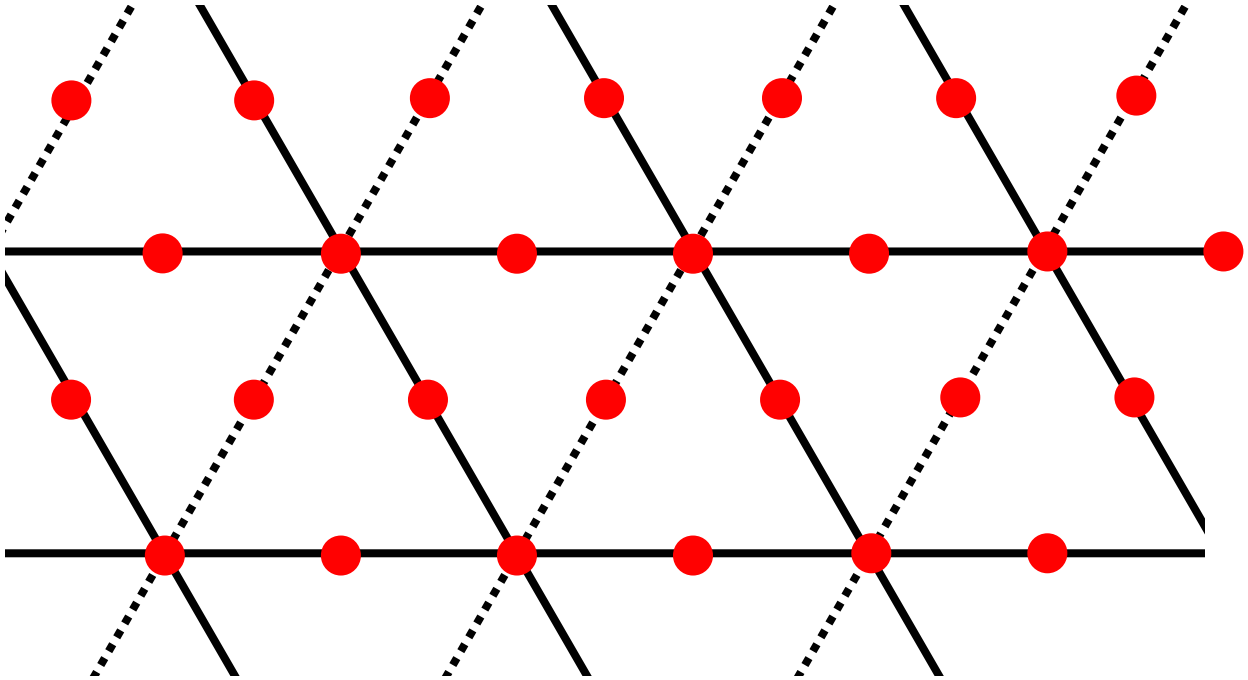
M_{222} Subdivision



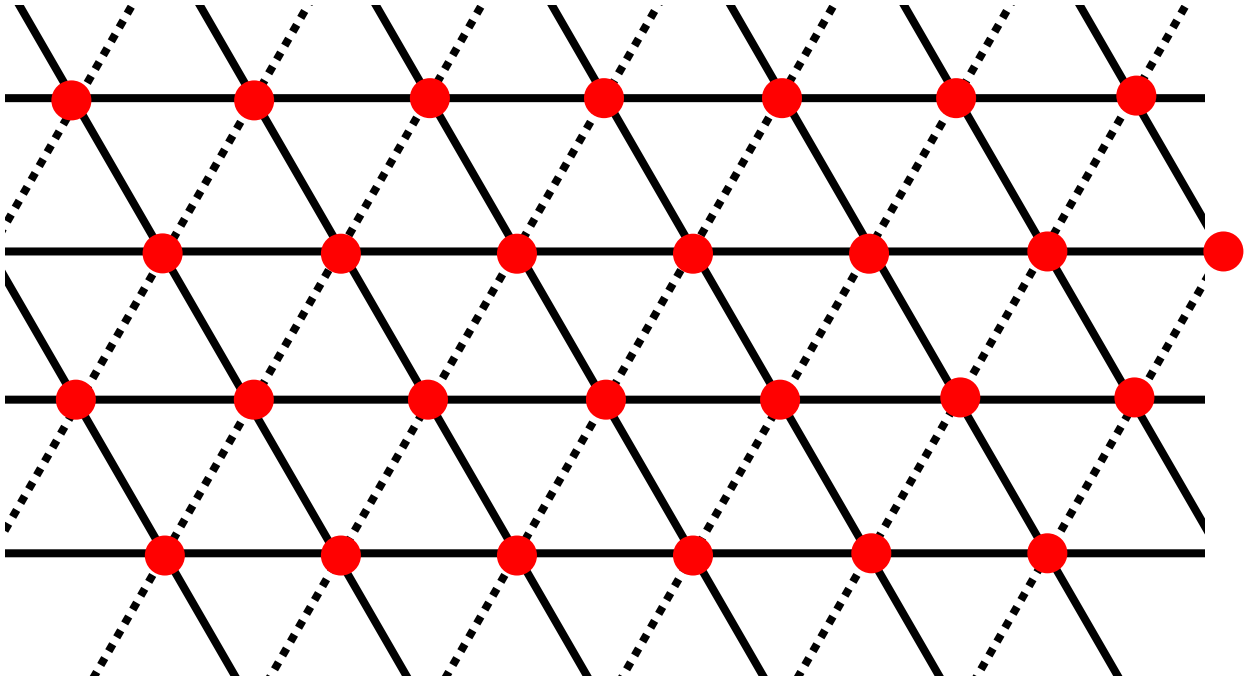
M_{222} Subdivision



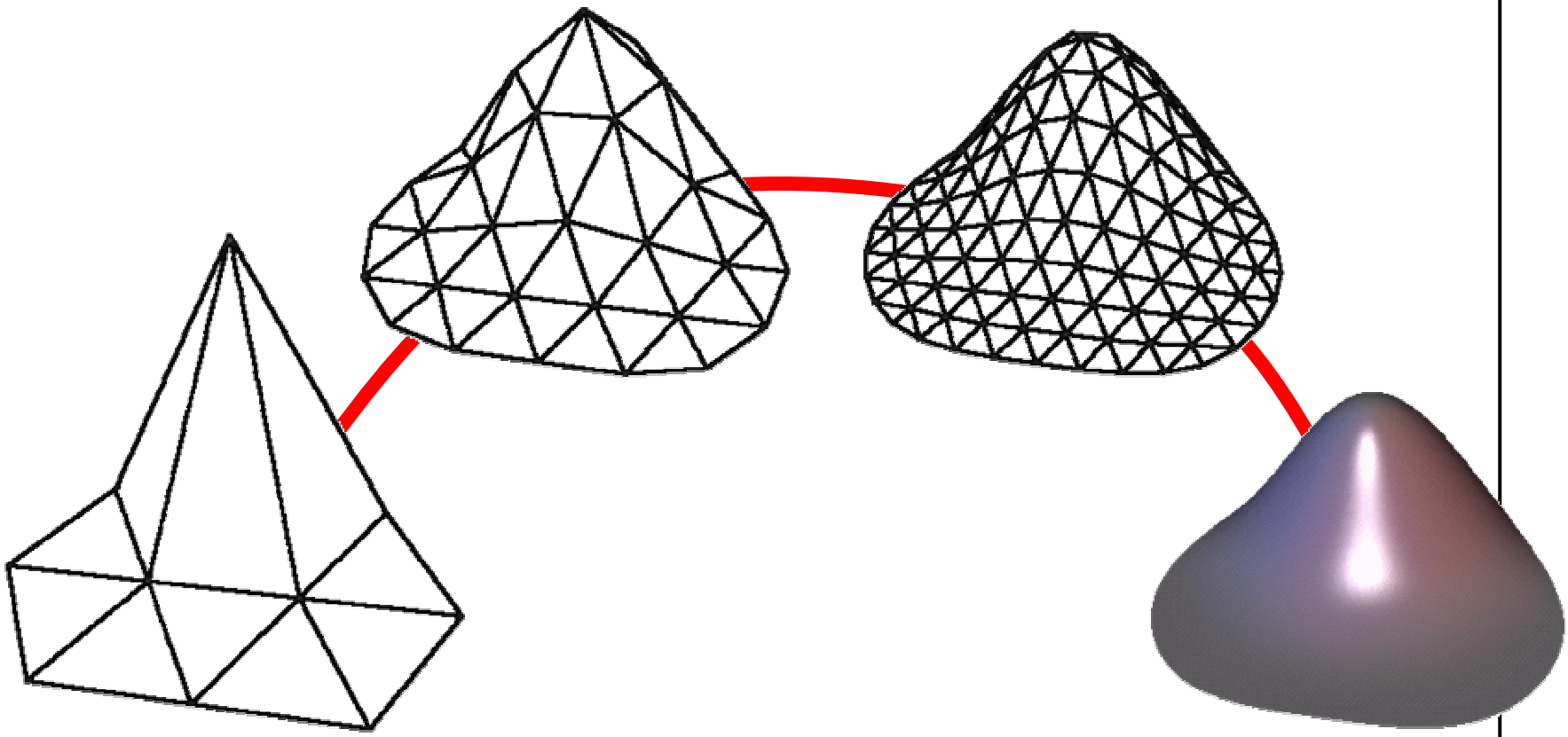
M_{222} Subdivision



M_{222} Subdivision



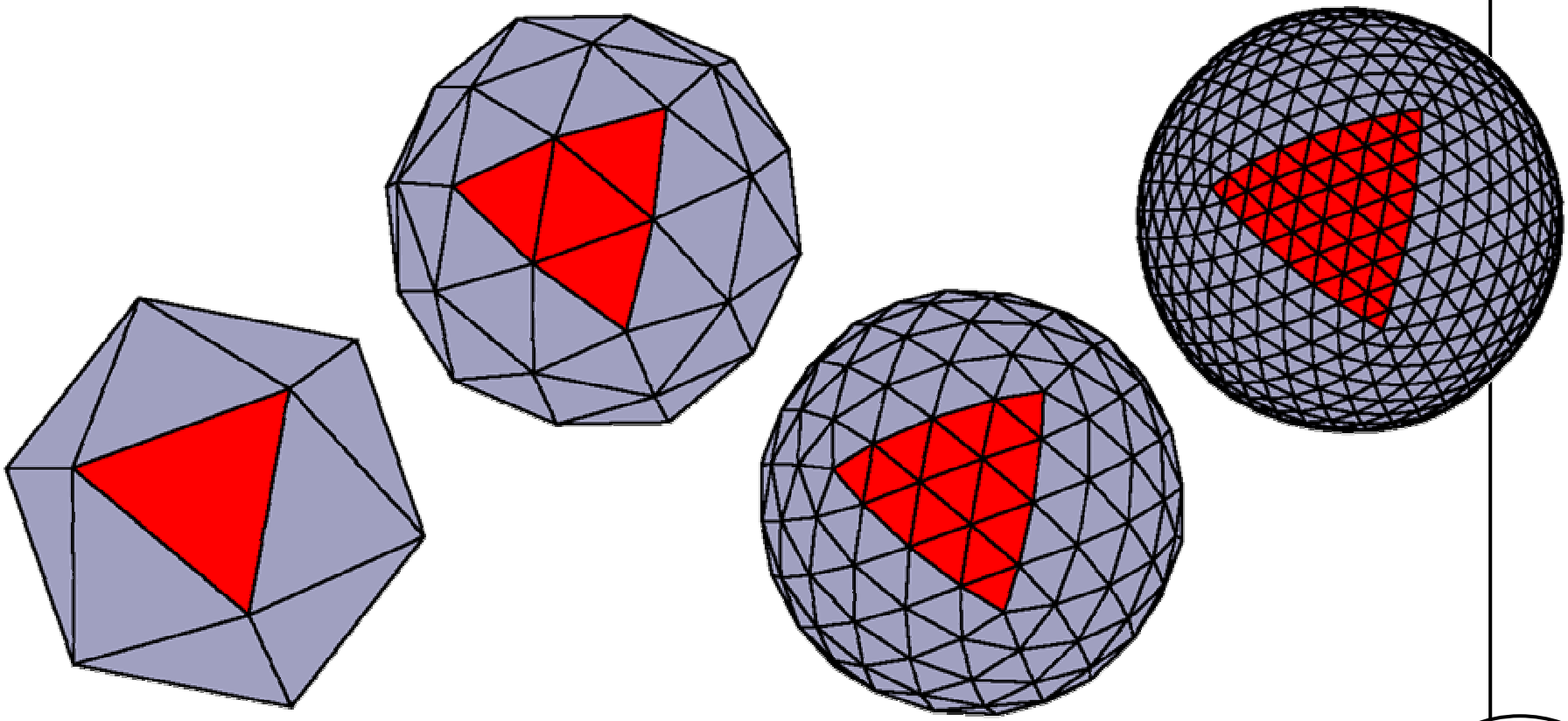
M_{222} Subdivision



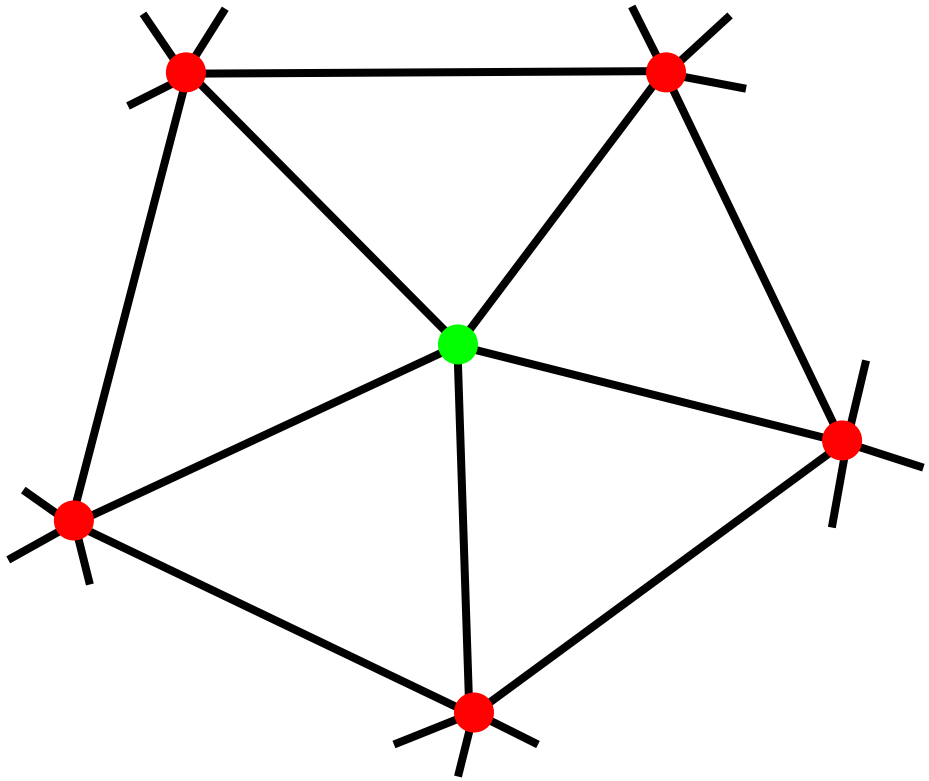
Subdivision Algorithms

- So far: regular meshes only
- Generalization to arbitrary meshes
  - Splitting operator
  - Smoothing operator
- Remember: uniform splitting generates *semi-regular* meshes

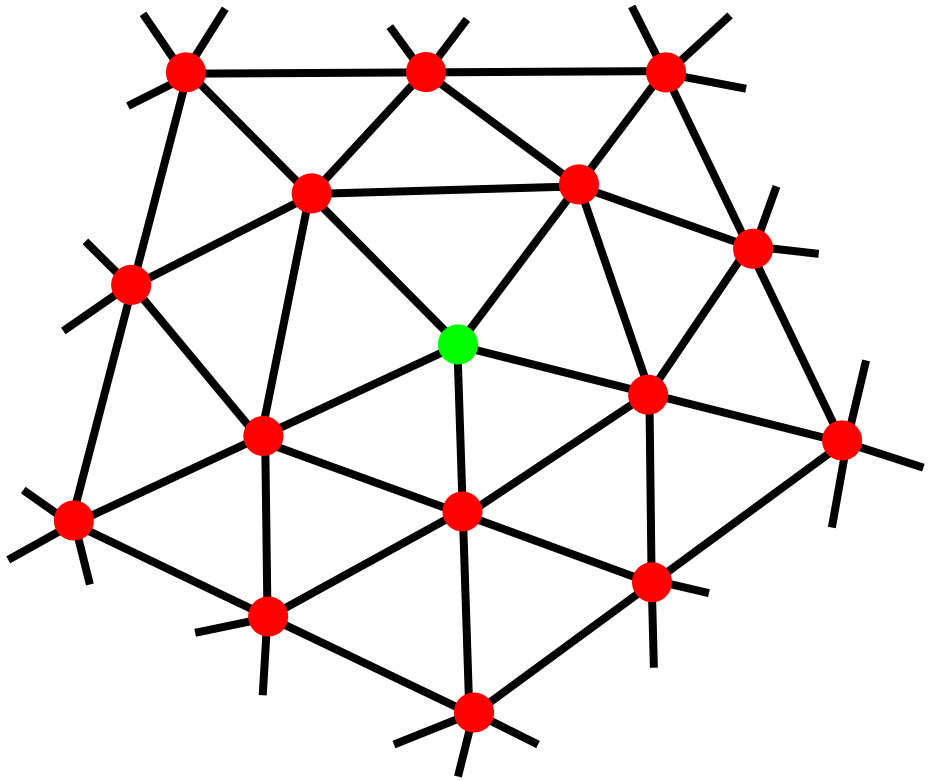
Subdivision Algorithms



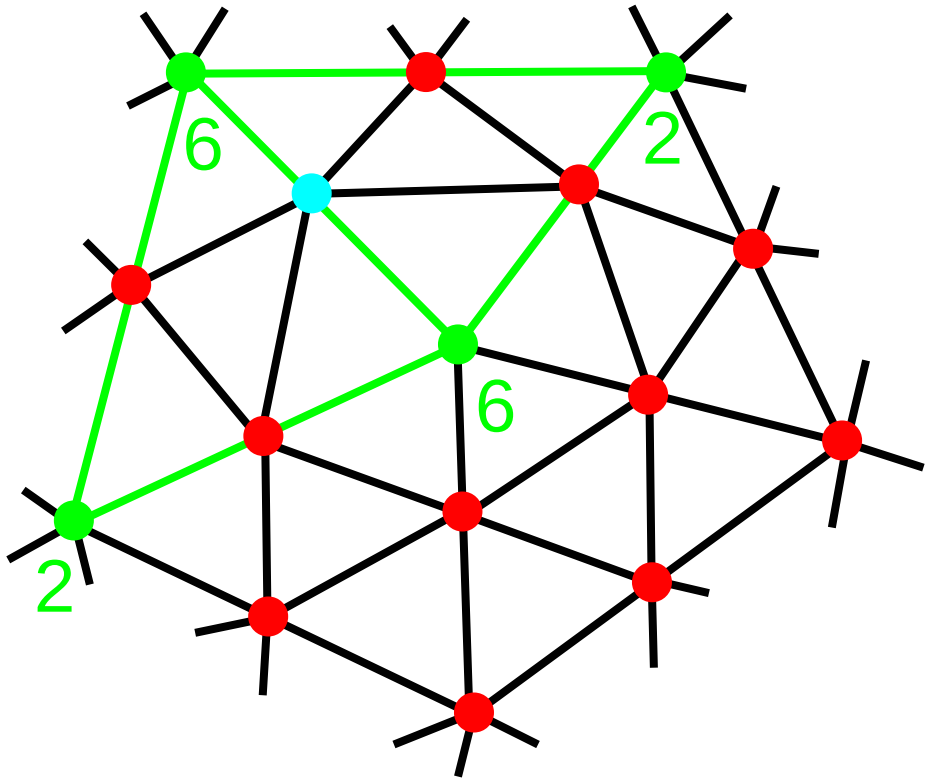
Extraordinary Vertices



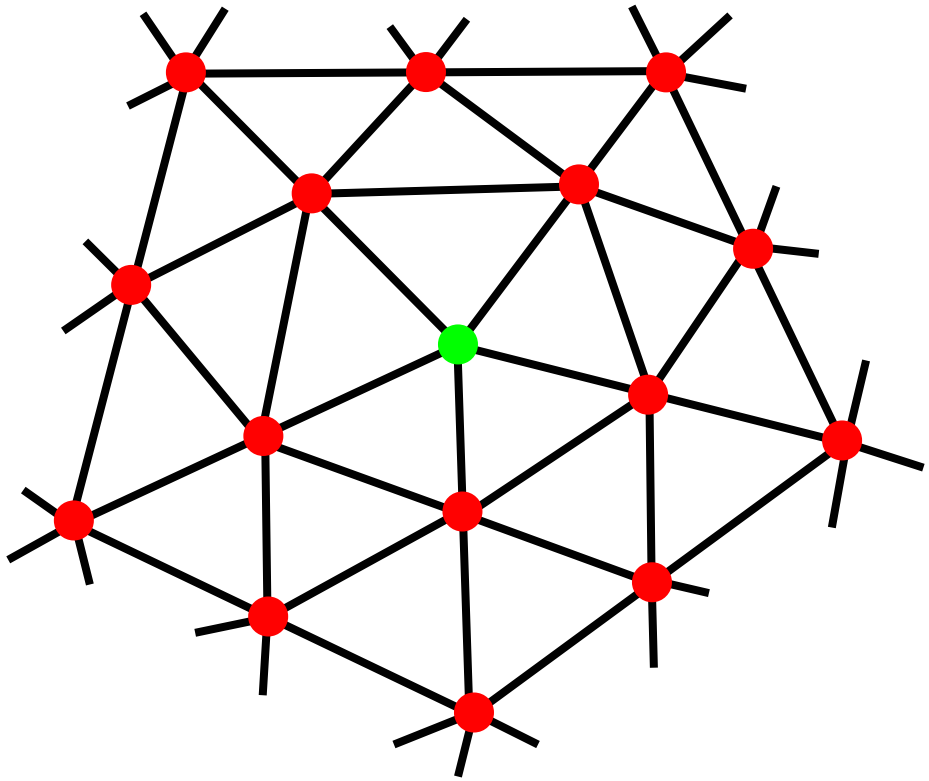
Extraordinary Vertices



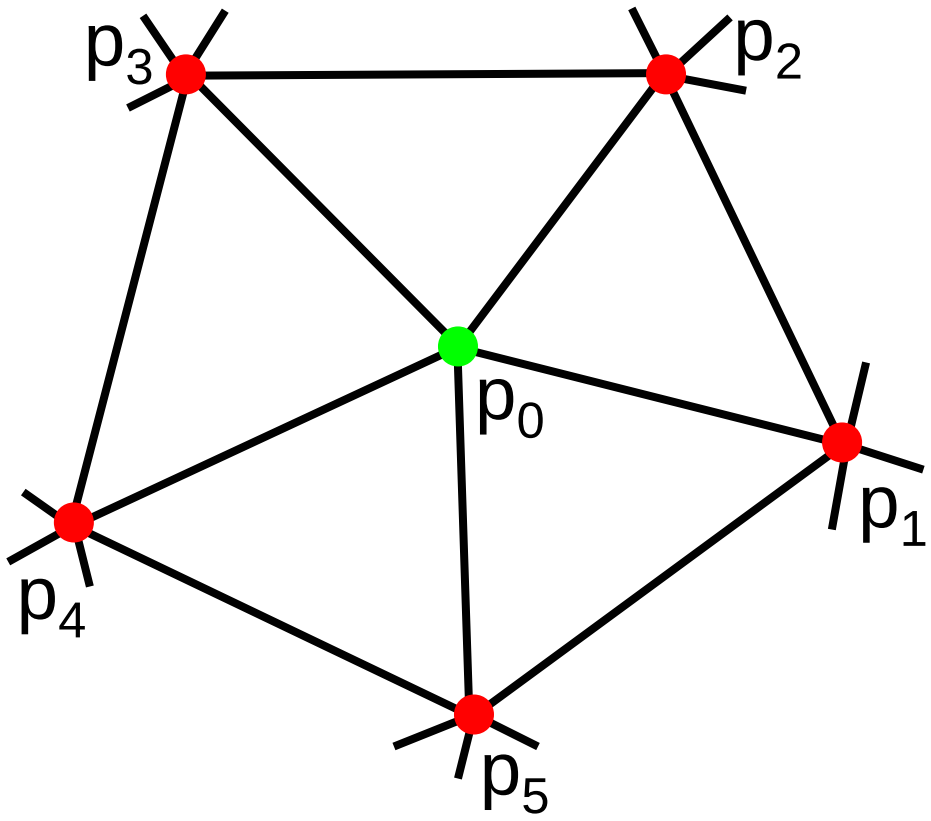
Extraordinary Vertices



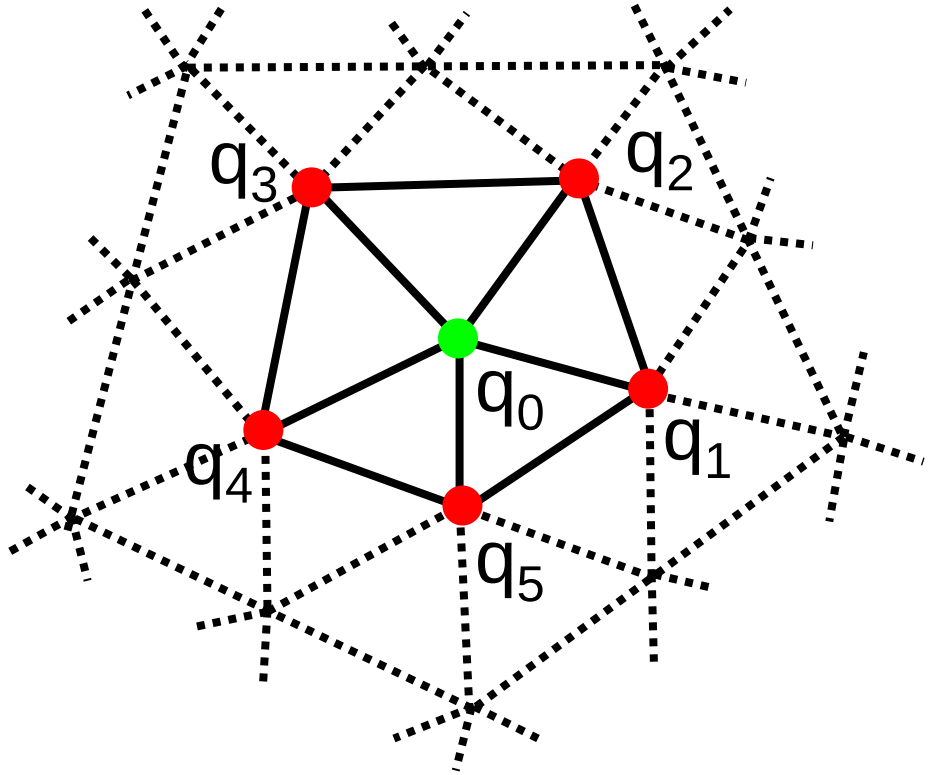
Extraordinary Vertices



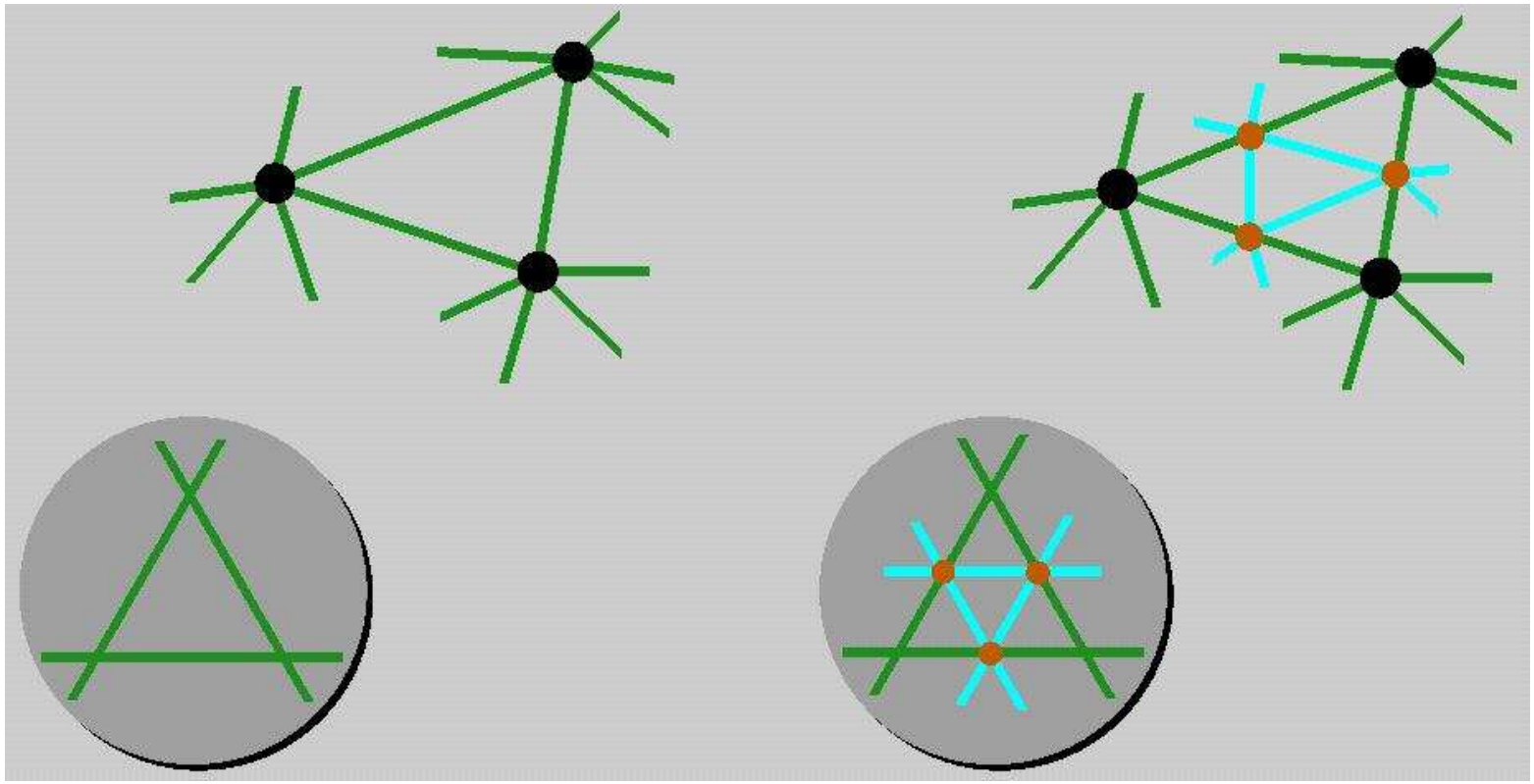
Extraordinary Vertices



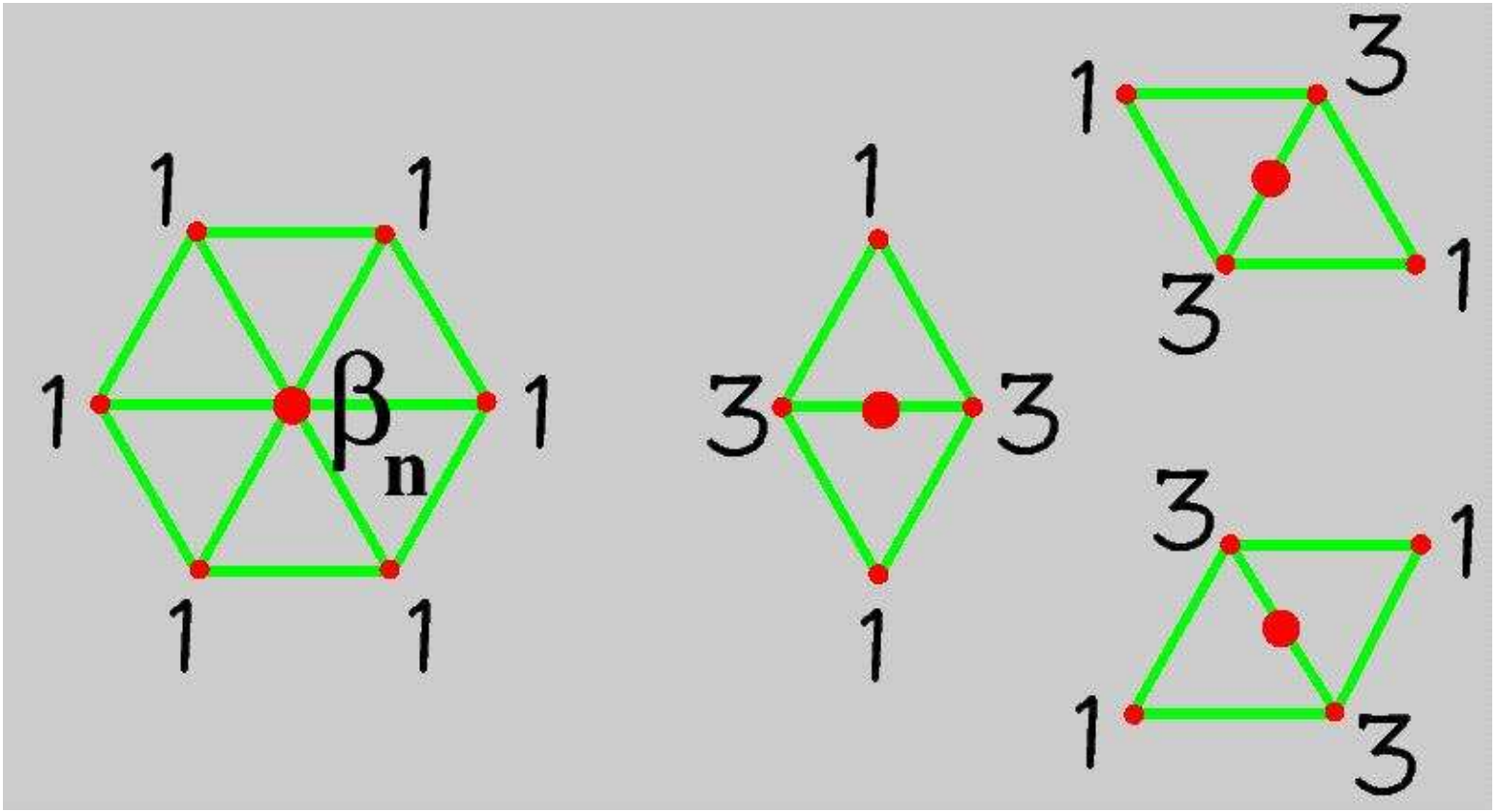
Extraordinary Vertices



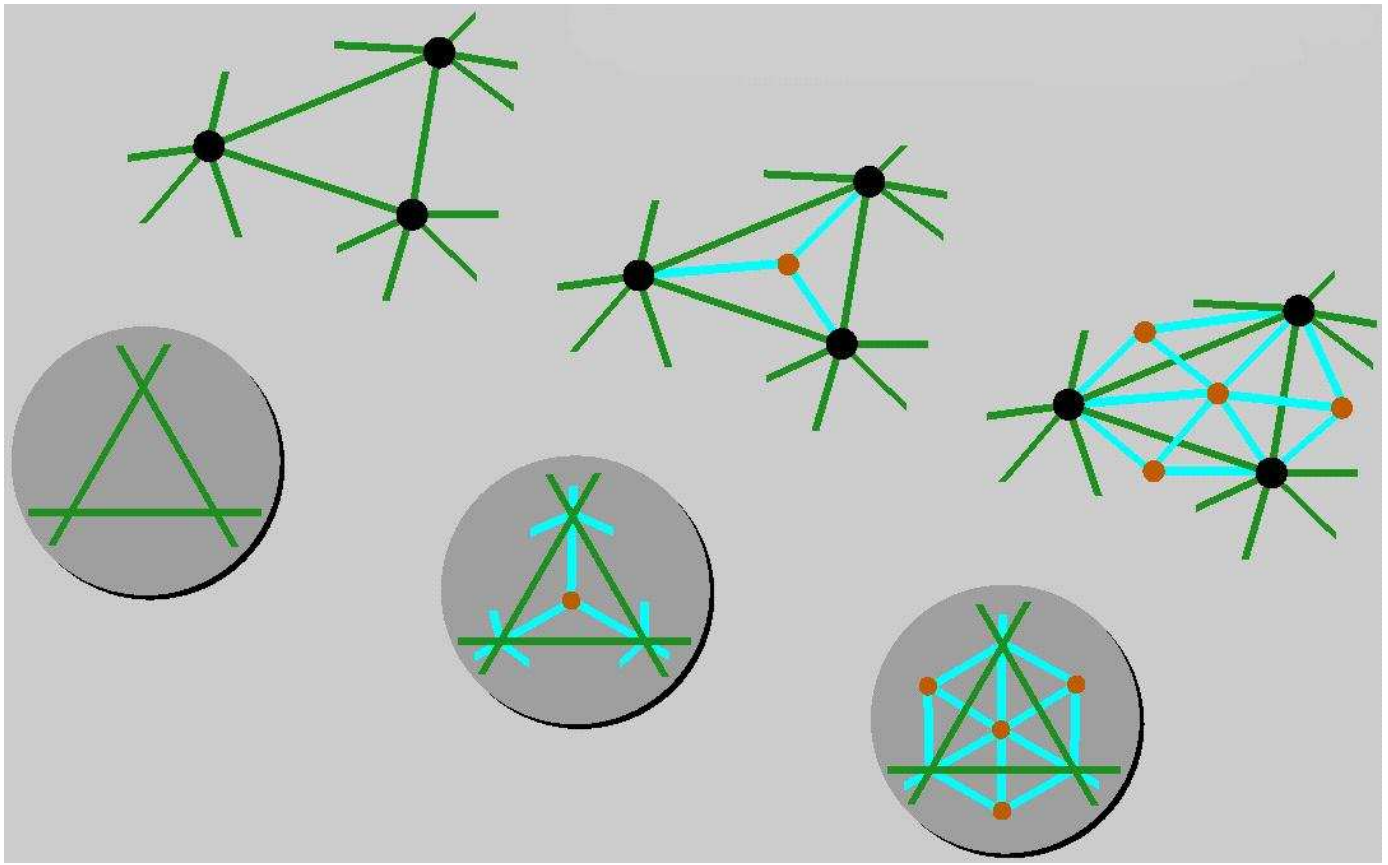
Loop Subdivision Scheme



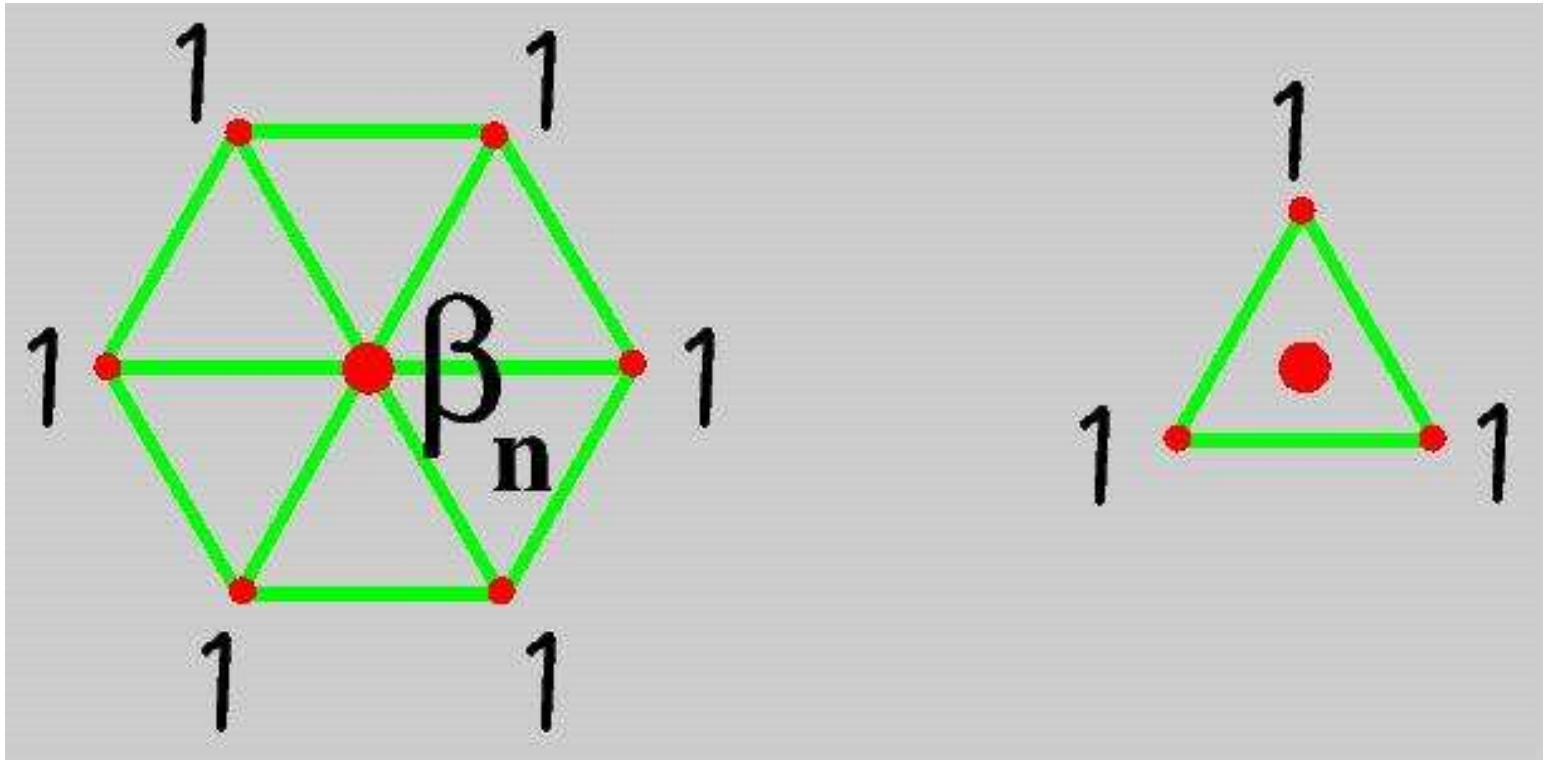
Loop Subdivision Scheme



$\sqrt{3}$ Subdivision Scheme



$\sqrt{3}$ Subdivision Scheme



Summary

- Subdivision schemes consist of
  - Splitting operator
  - Smoothing rule

Summary

- Subdivision schemes consist of
  - Splitting operator
    - Derived from uniform refinement and duality
    - Applied for each face (of an irregular mesh)
    - Generates semi-regular meshes
    - Special operators, e.g., for adaptive refinement
  - Smoothing rule

Summary

- Subdivision schemes consist of
  - Splitting operator
  - Smoothing rule
    - Rules for regular regions derived from box-splines
    - Subdivision masks obtained by averaging
    - Subdivision masks encode 4 different rules
    - Generalization to extraordinary vertices by analyzing the subdivision matrix