Subdivision Curves and Surfaces

Subdivision Curves

- Repeatedly refine the control polygon:

$$P^0 \rightarrow P^1 \rightarrow P^2 \rightarrow \cdots$$

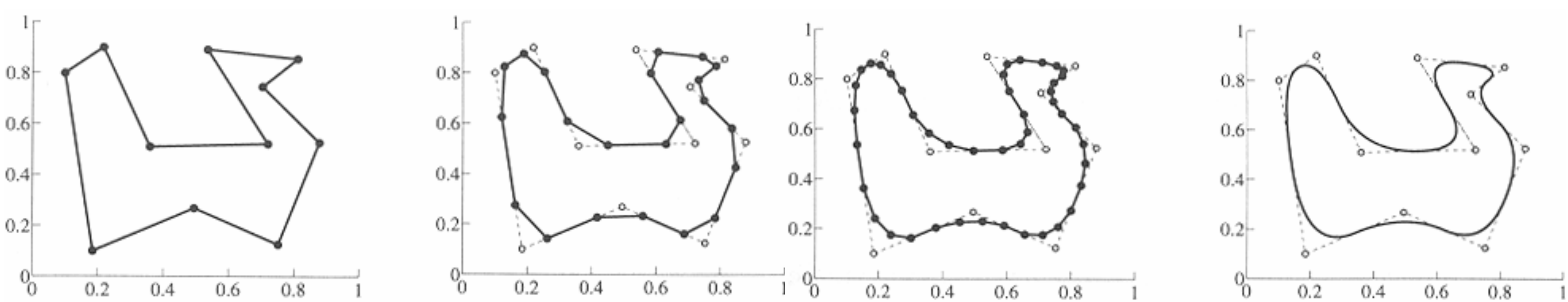
- The curve is the limit of this process, $C = \lim_{j \rightarrow \infty} P^j$

Subdivision Curves

- Repeatedly refine the control polygon:

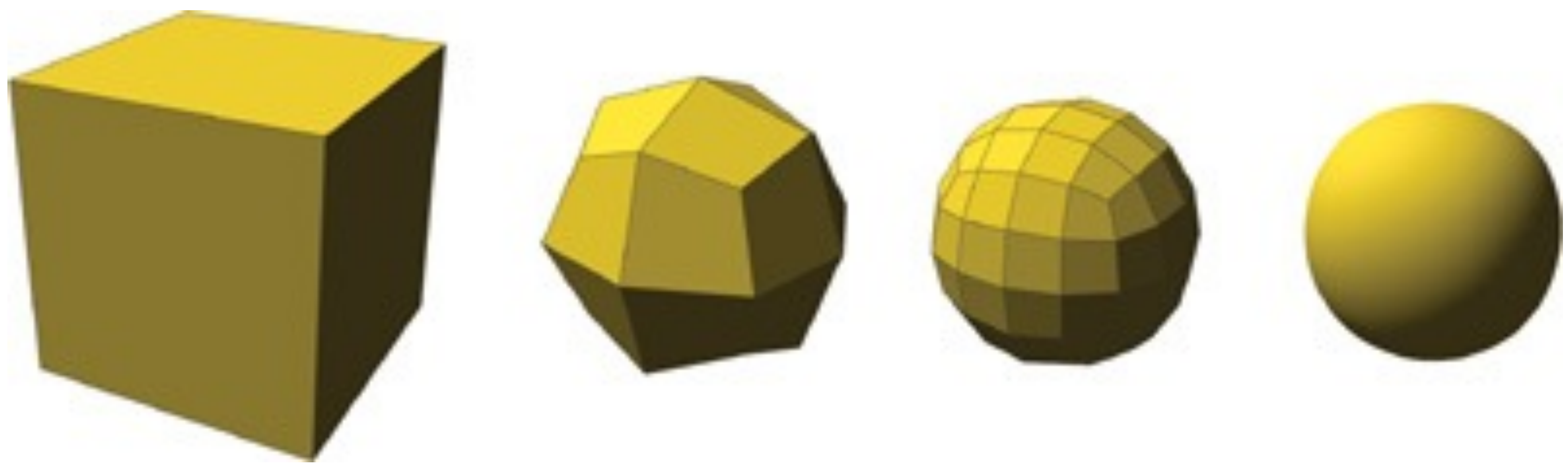
$$P^0 \to P^1 \to P^2 \to \cdots$$

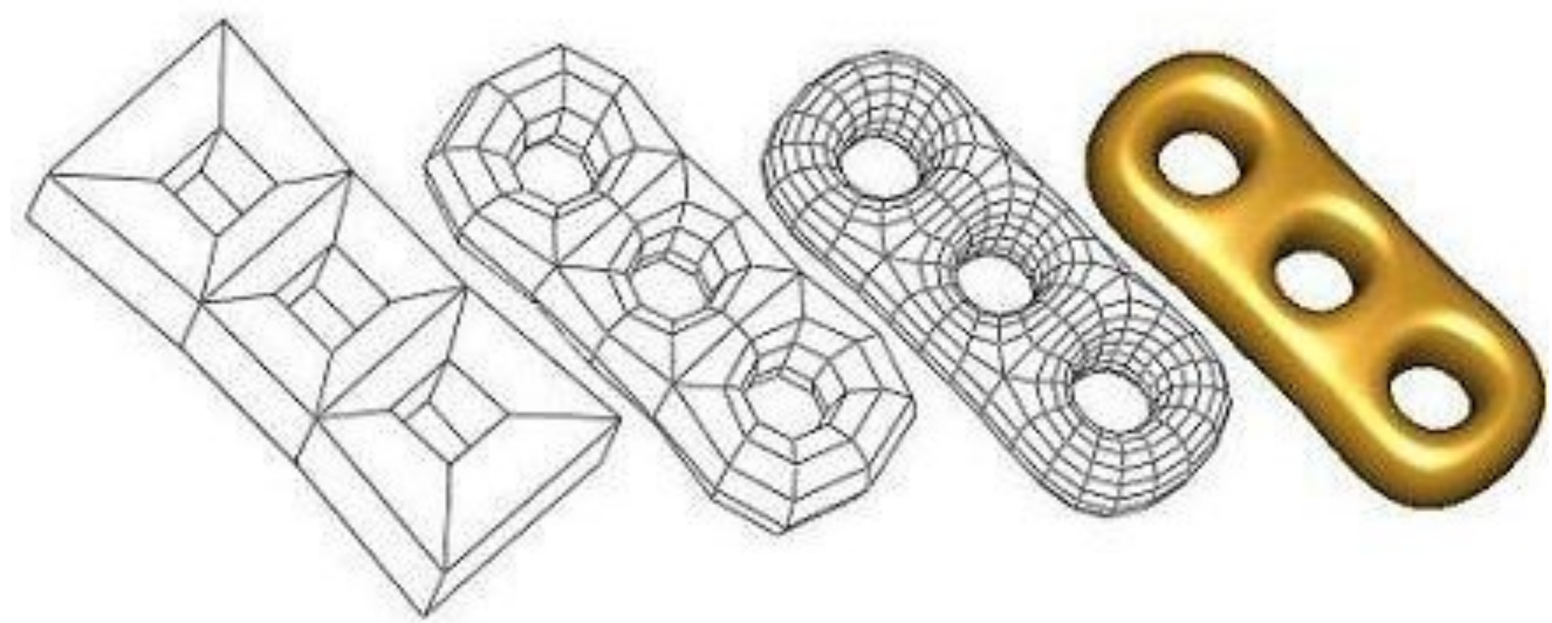
- The curve is the limit of this process, $C = \lim_{j \to \infty} P^j$



Subdivision Surfaces

$$M^0 \to M^1 \to M^2 \to \cdots \to M^\infty$$

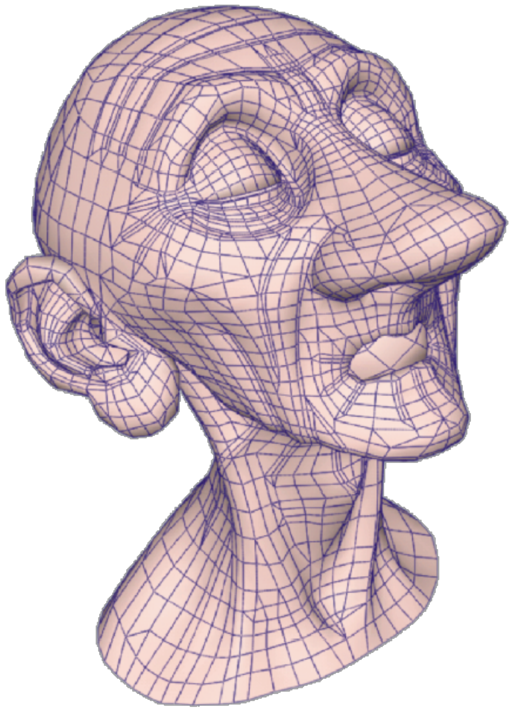




Why Use Subdivision?

- Generates smooth surfaces from polygonal meshes of arbitrary topology
- Convenient for animation
- Efficient rendering
- Built-in level-of-detail
- Compression
- Smoothing

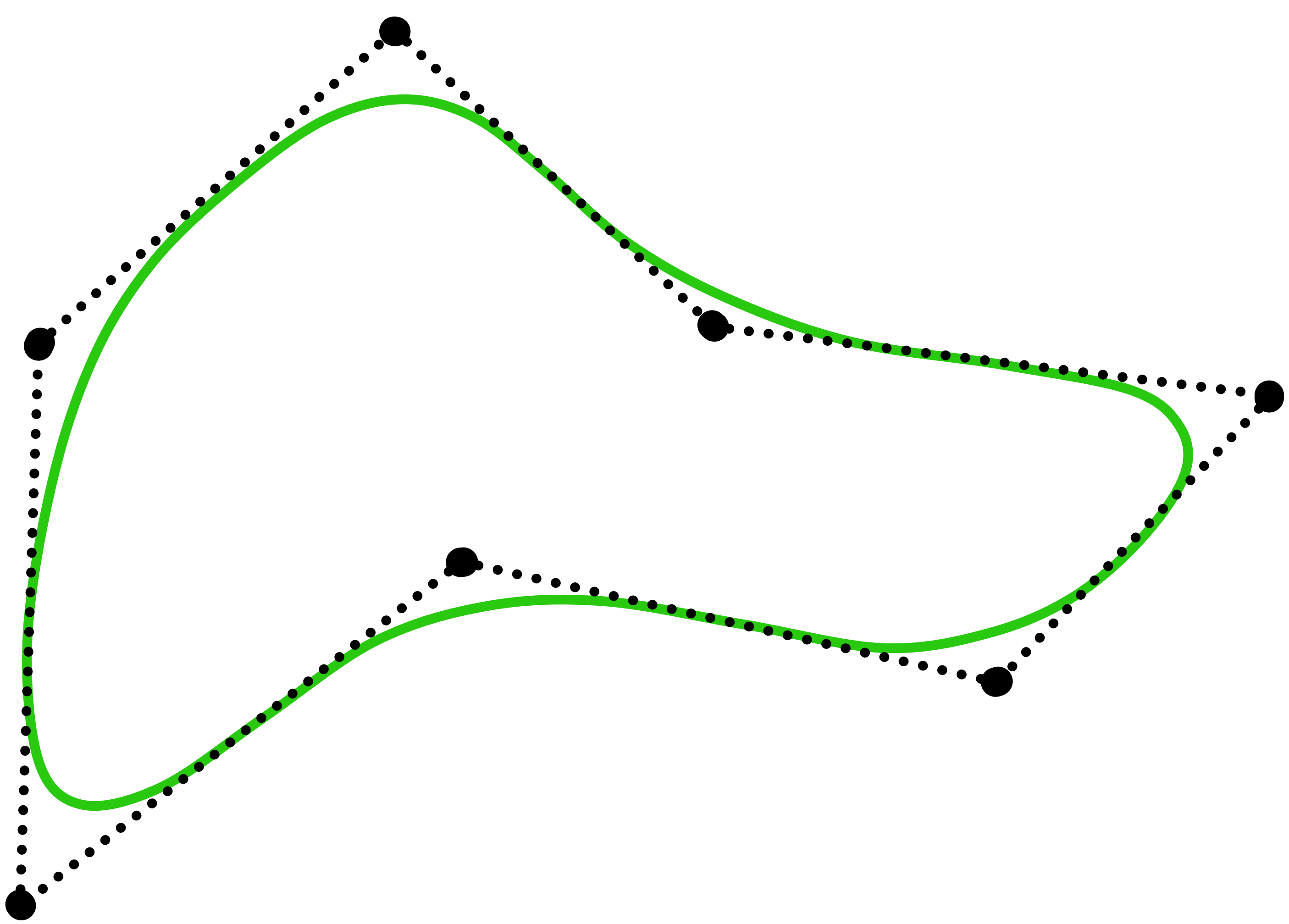
Examples



Subdivision Schemes:

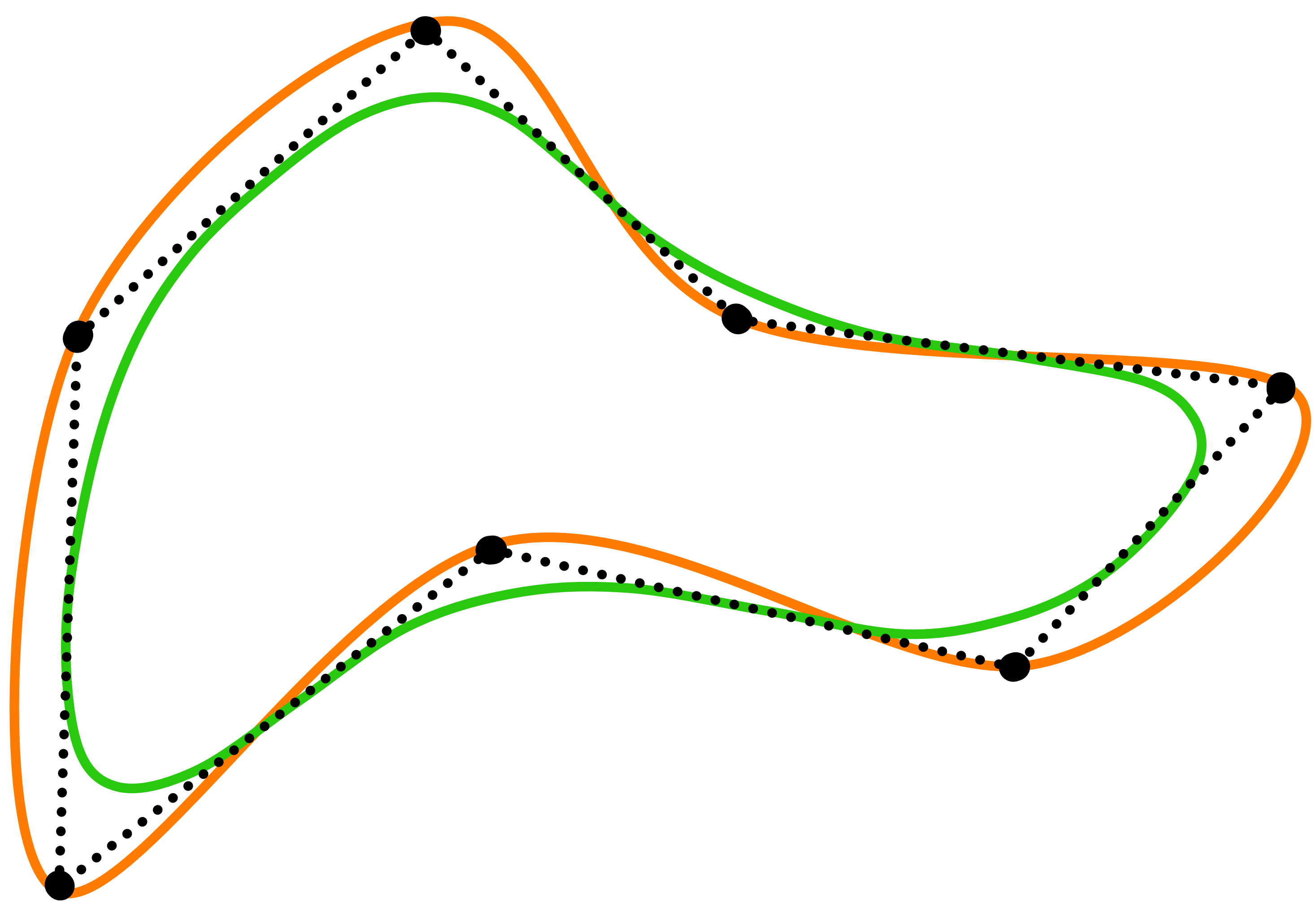
two main groups

- **Approximating:** limit curve/surface does not pass through the control vertices



Subdivision Schemes:
two main groups

- **Approximating:** limit curve/surface does not pass through the control vertices
- **Interpolating:** control vertices lie on the limit curve/surface



Control Polygon Refinement

- *Splitting step*: introduce new vertices into the control polygon, resulting in a refined polygon:

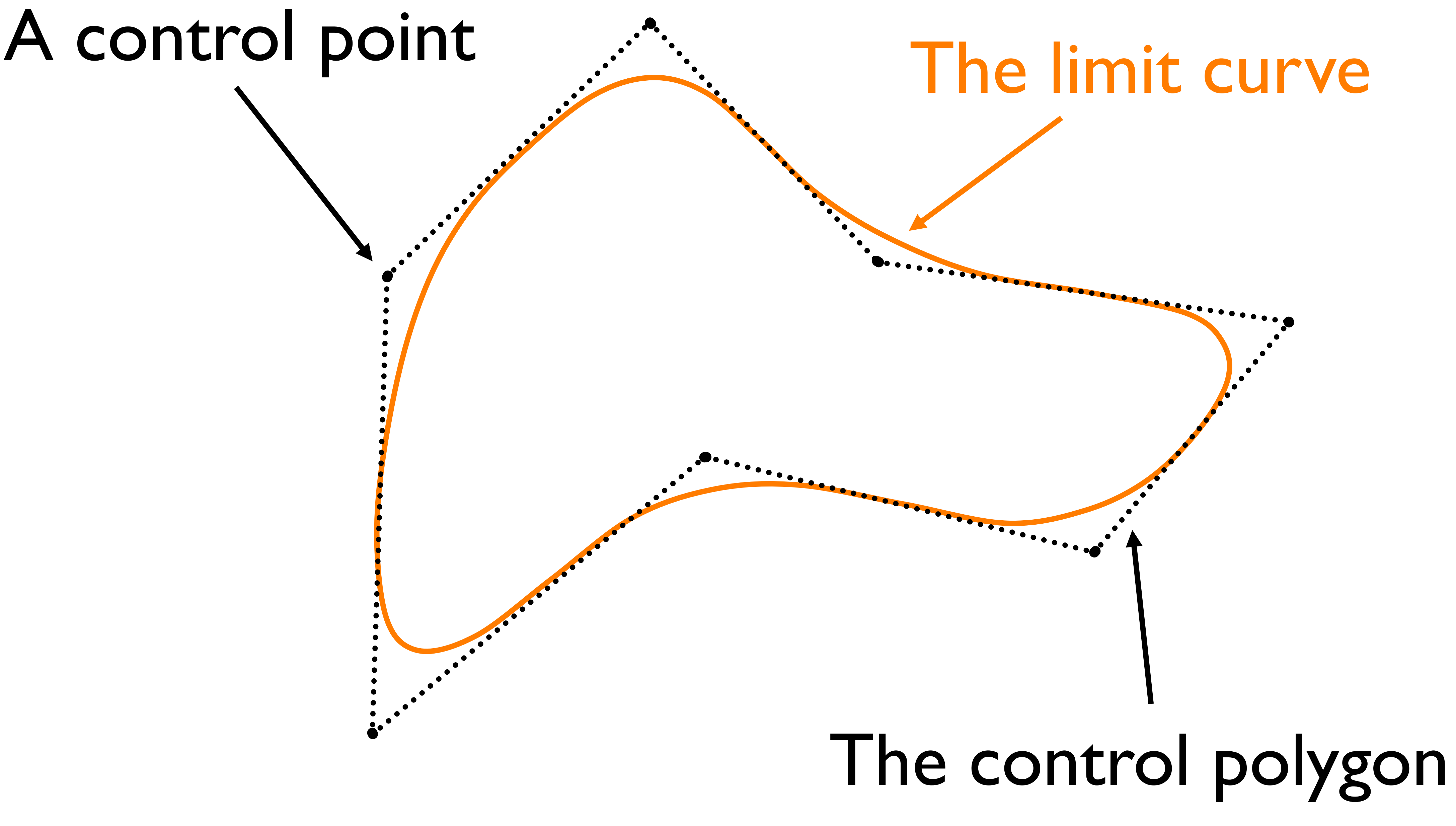
$$\hat{p}^{j}_{2i} = p^{j-1}_{i} \qquad \hat{p}^{j}_{2i+1} = 0.5\left(p^{j-1}_{i} + p^{j-1}_{i+1}\right)$$

- *Averaging step*: compute a new location for each vertex in the refined polygon by averaging its local neighborhood:

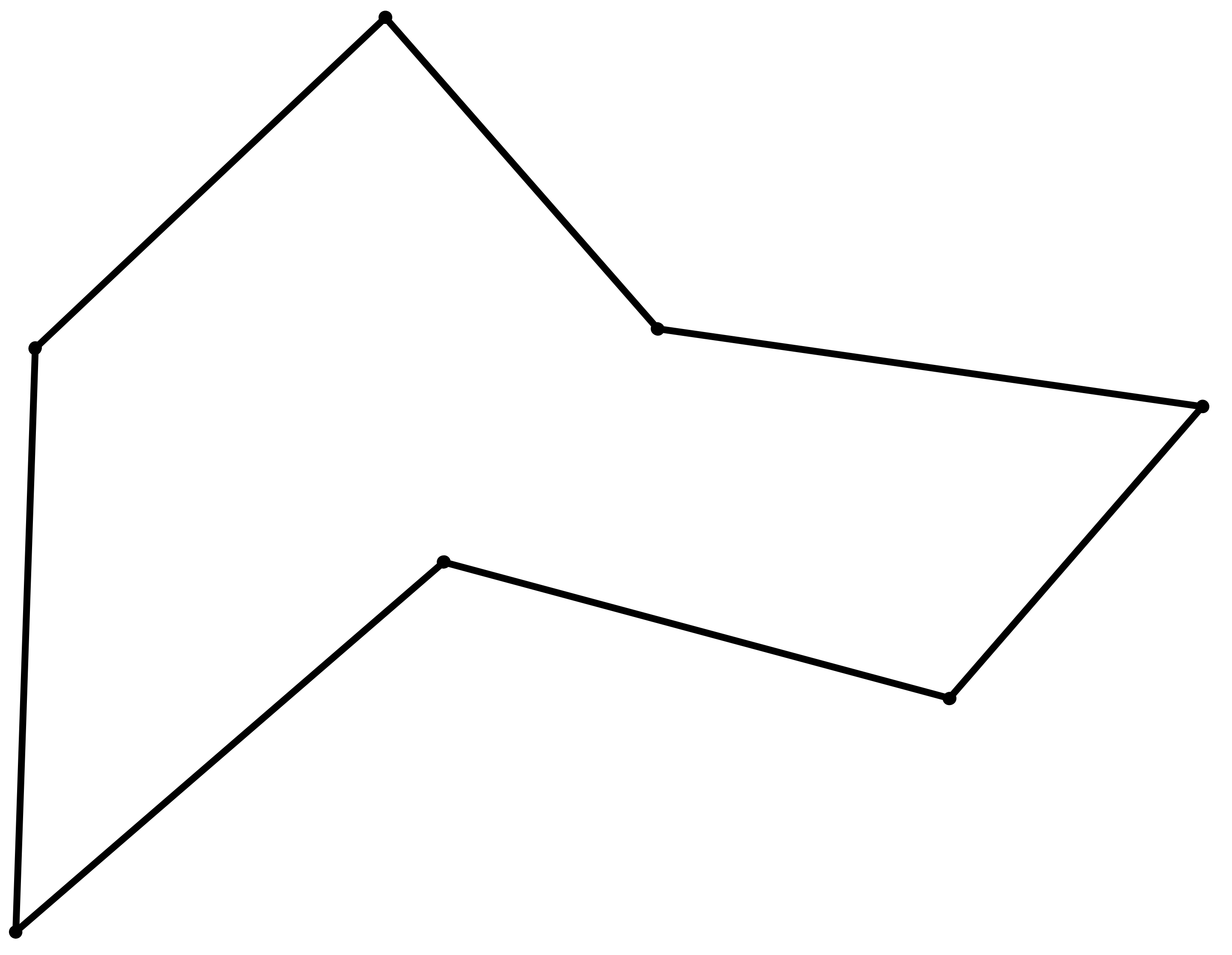
$$p^{j}_{i} = \sum_{k} r_k \hat{p}^{j}_{i+k}$$

- Different schemes differ in the way they perform the averaging step.

Example 1: Corner Cutting

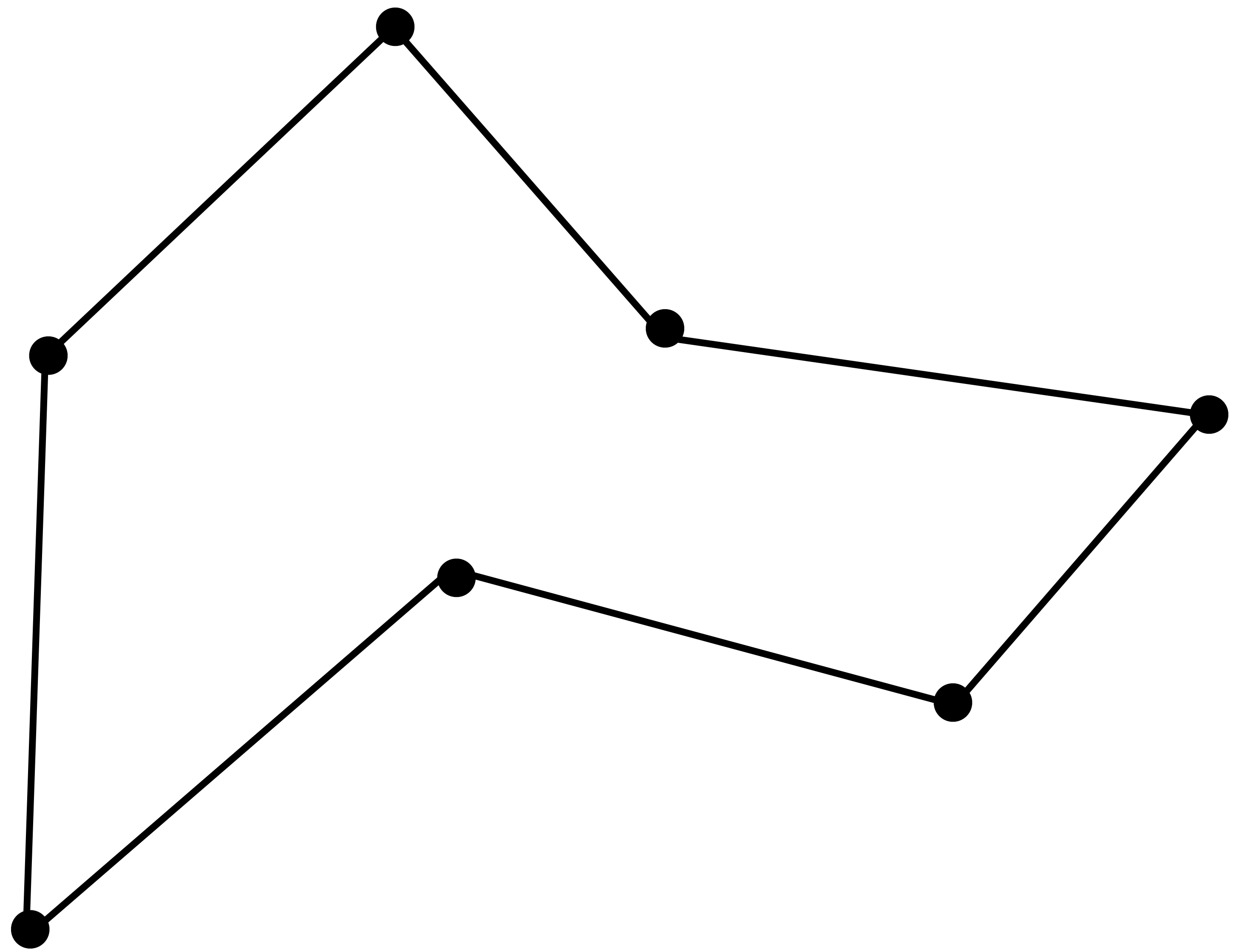


Corner Cutting

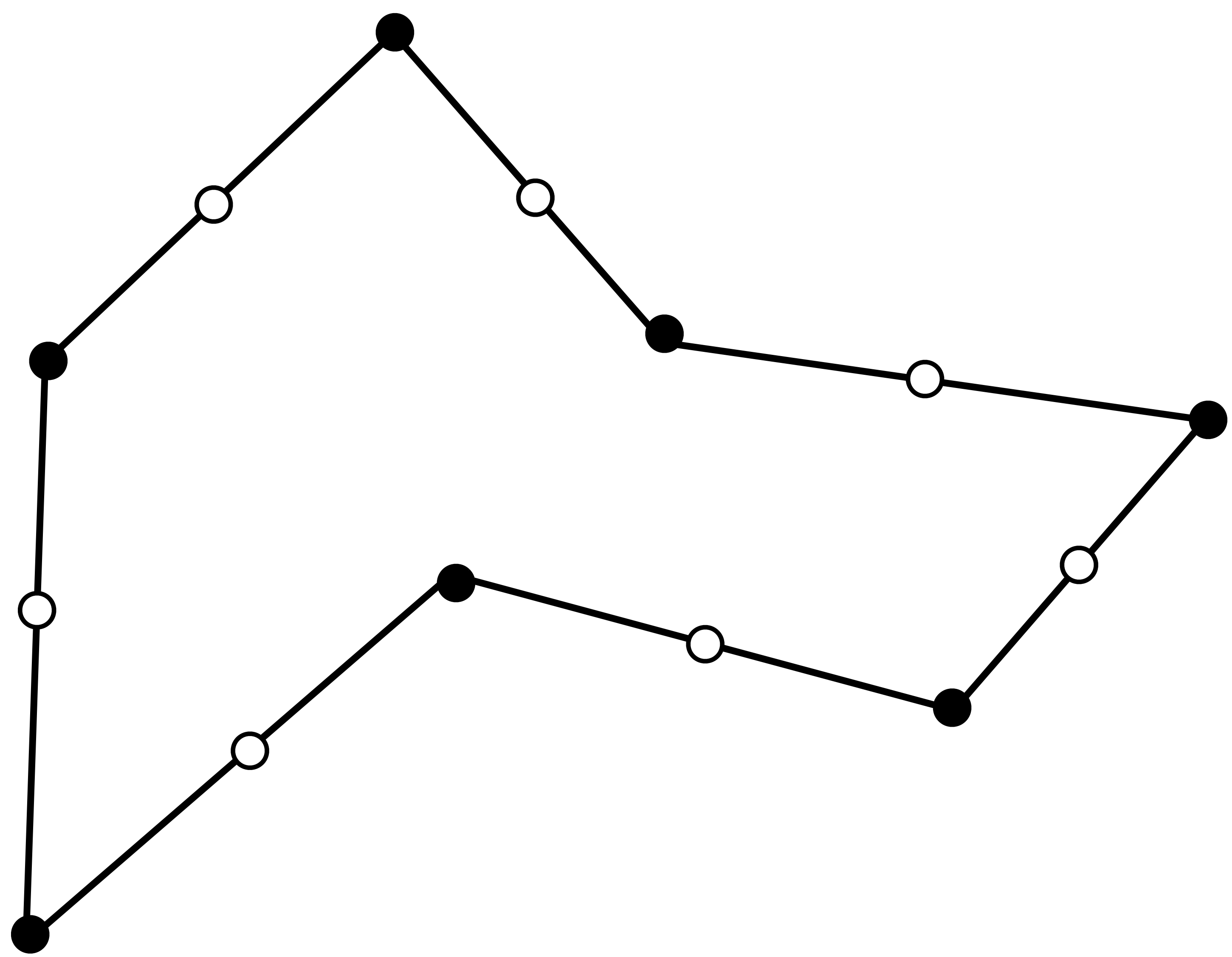


original control polygon

Corner Cutting

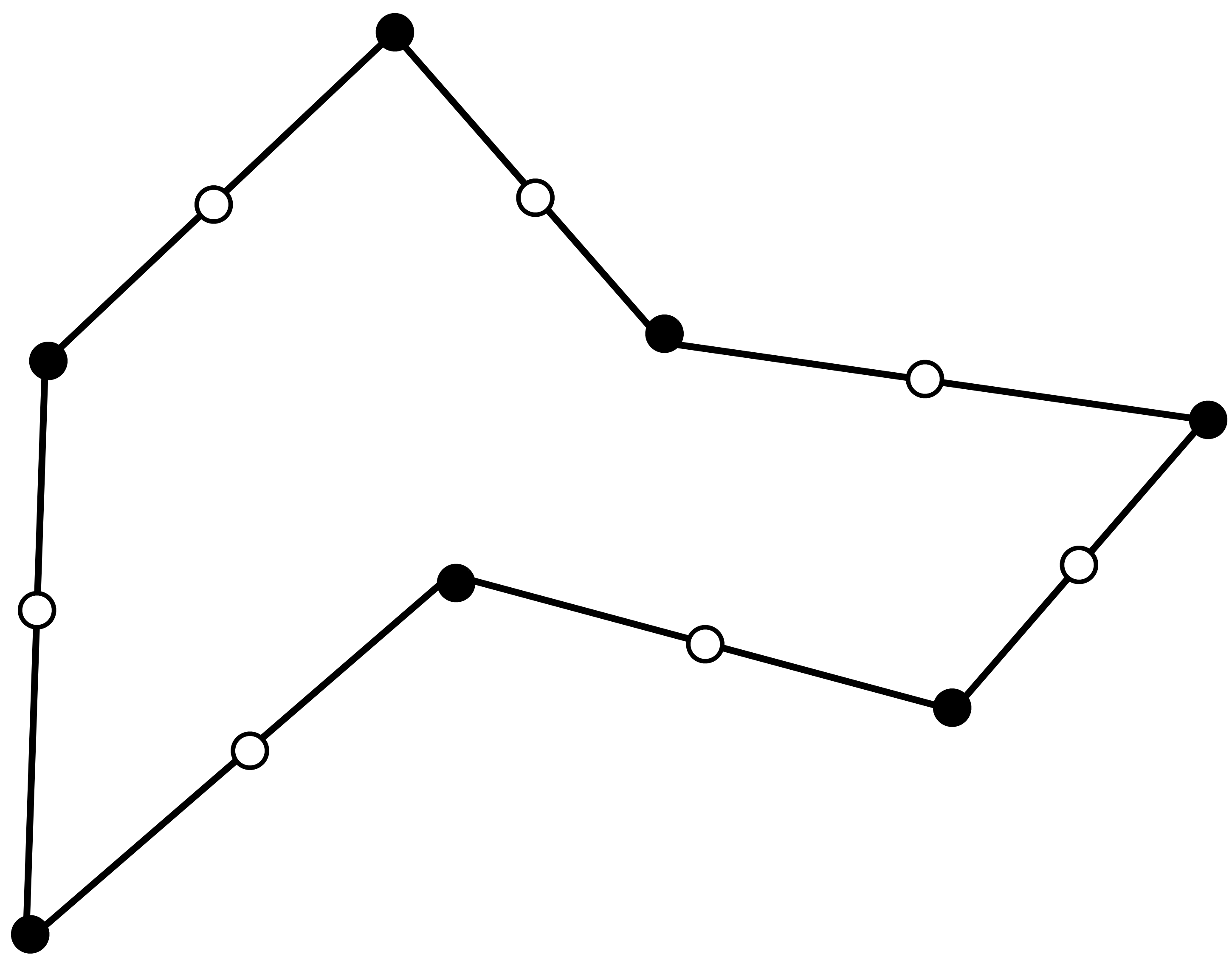


Corner Cutting



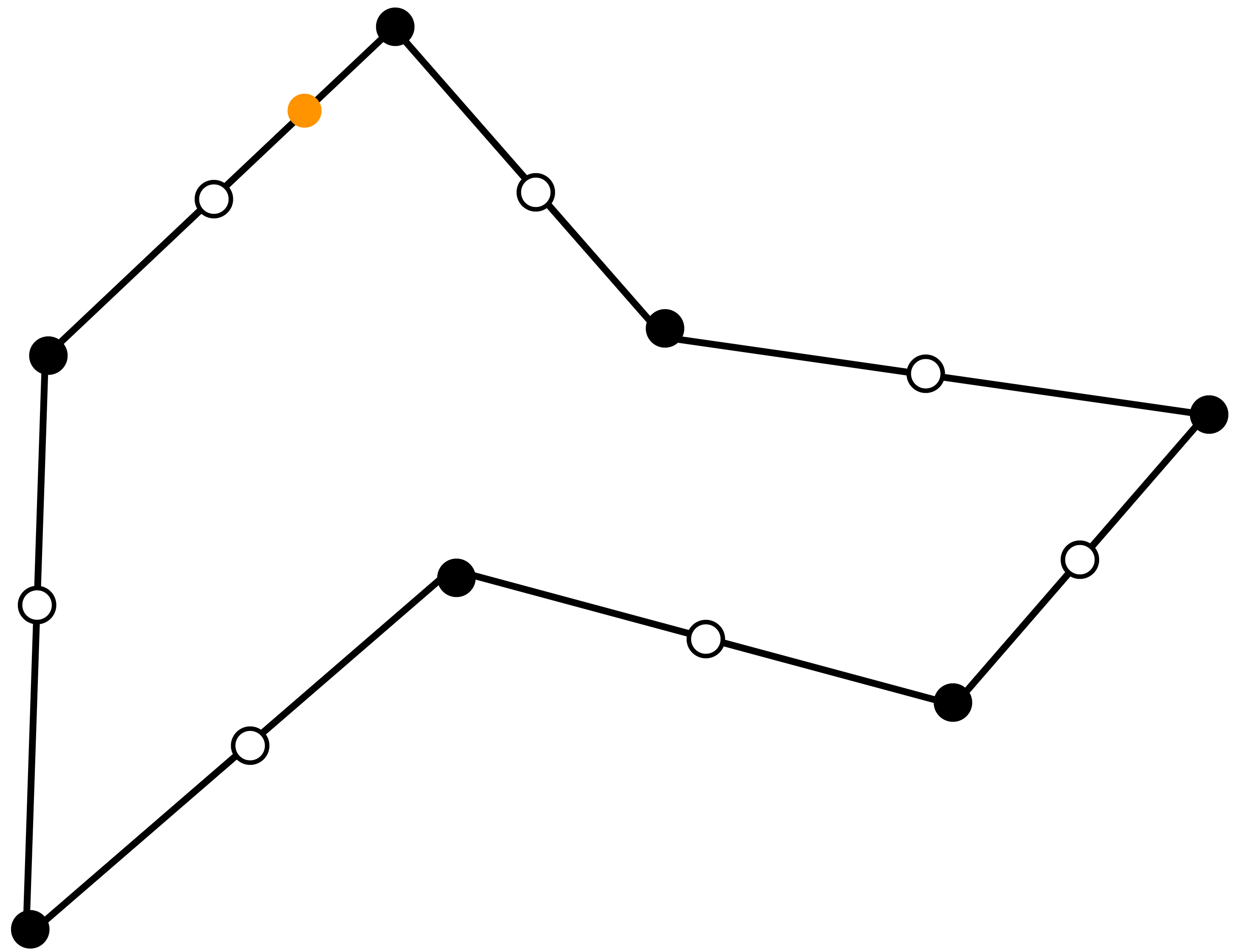
refined control polygon

Corner Cutting



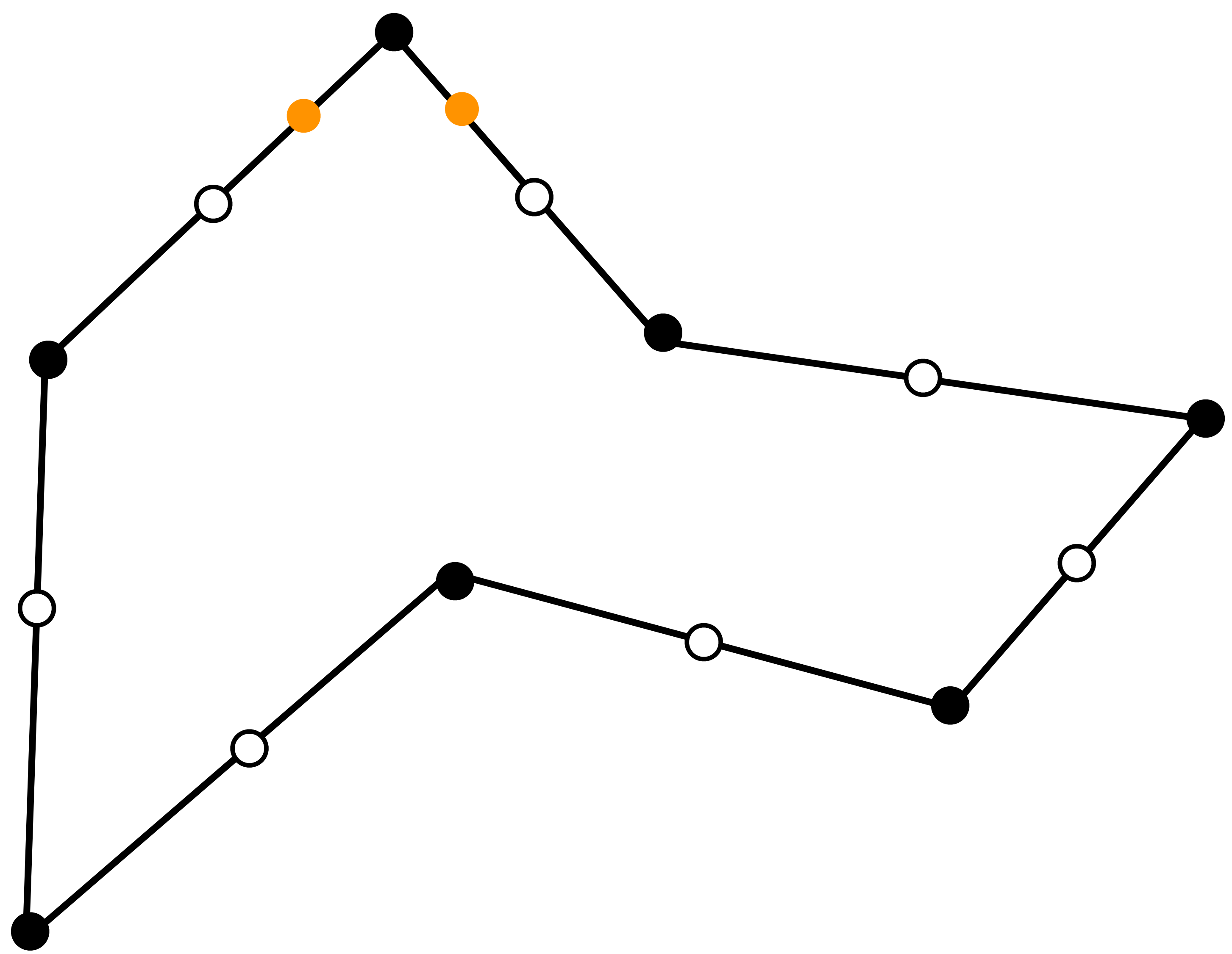
apply the averaging mask: $[r_0, r_1] = [1/2, 1/2]$

Corner Cutting



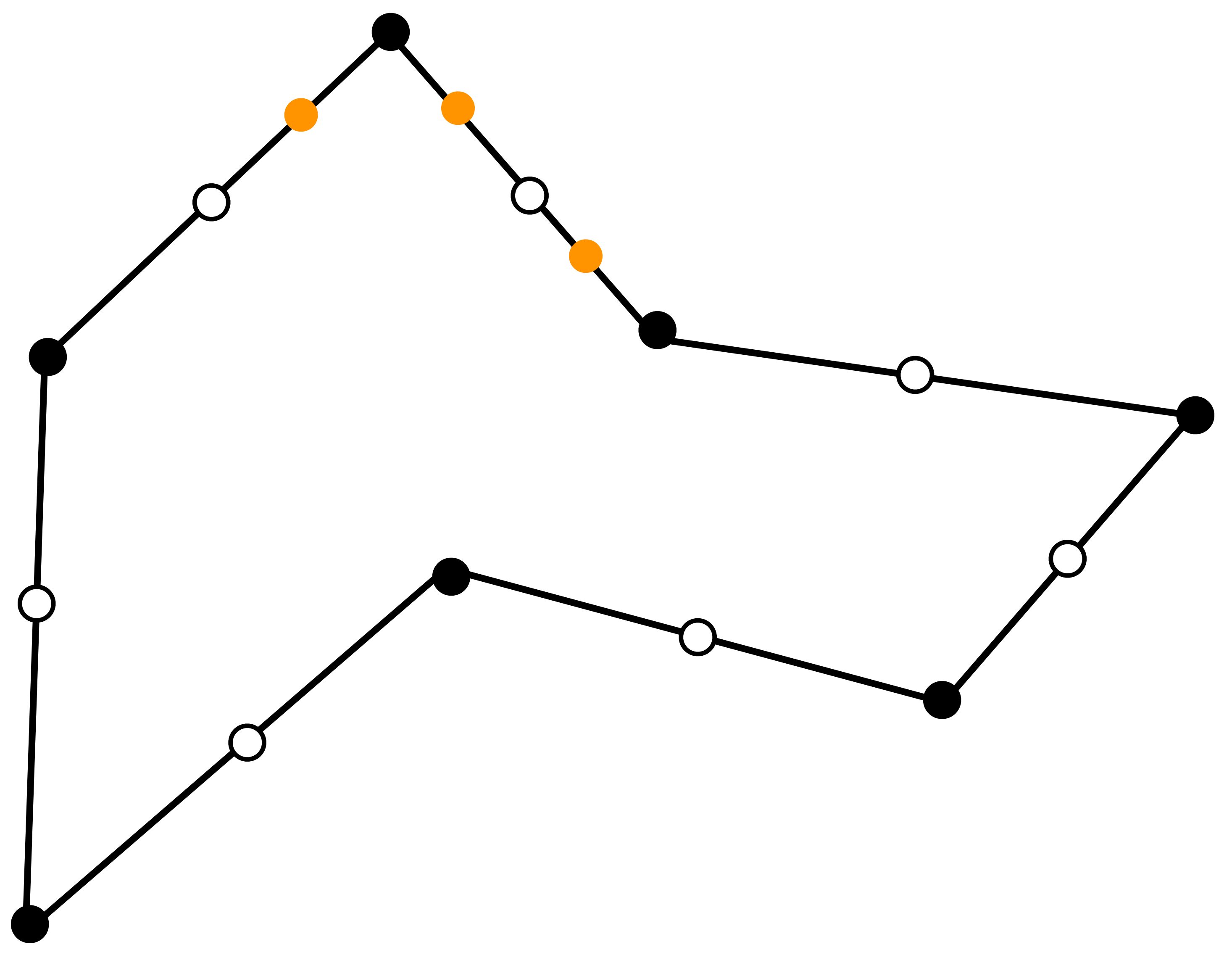
apply the averaging mask: $[r_0, r_1] = [1/2, 1/2]$

Corner Cutting



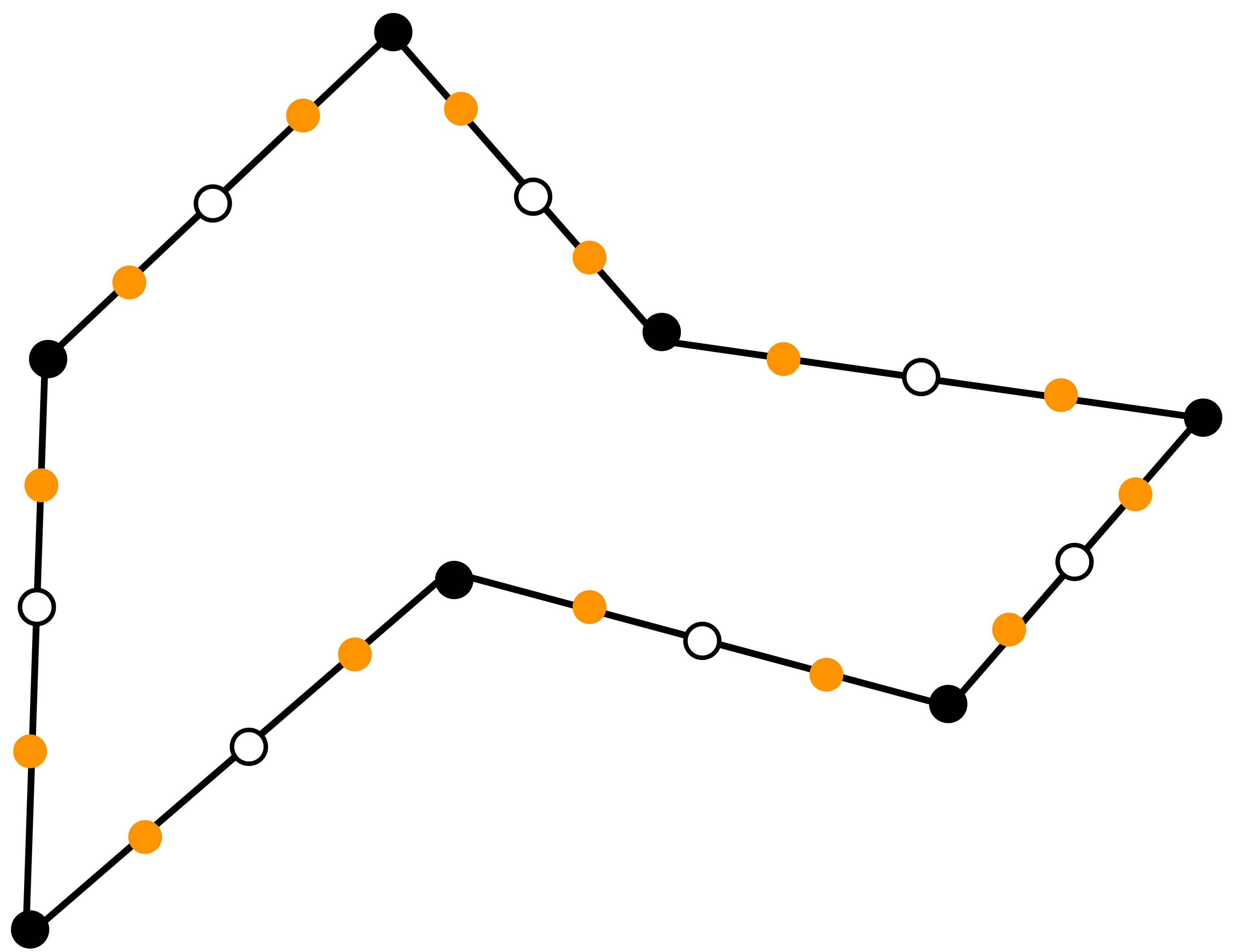
apply the averaging mask: $[r_0, r_1] = [1/2, 1/2]$

Corner Cutting



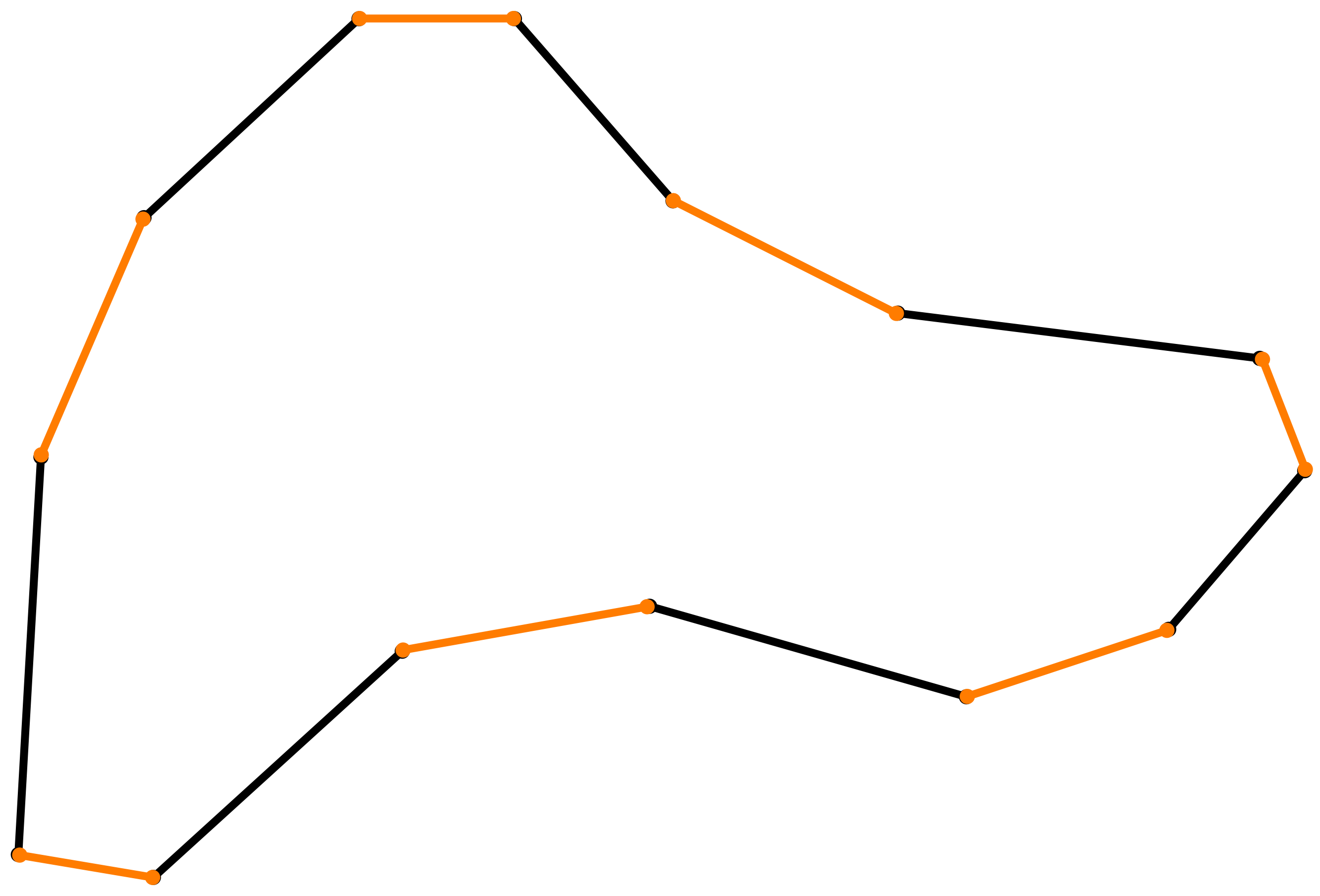
apply the averaging mask: $[r_0, r_1] = [1/2, 1/2]$

Corner Cutting



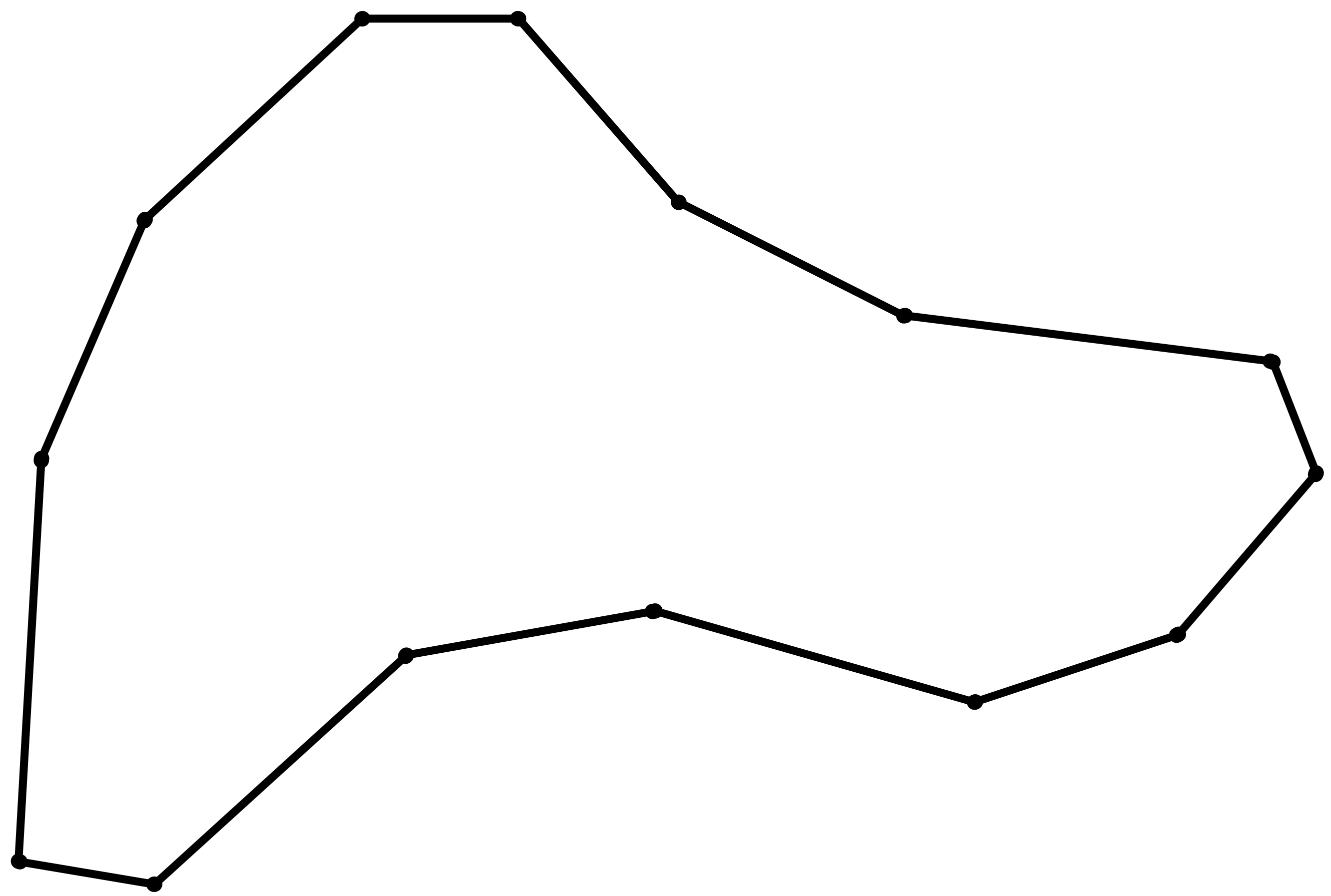
apply the averaging mask: $[r_0, r_1] = [1/2, 1/2]$

Corner Cutting



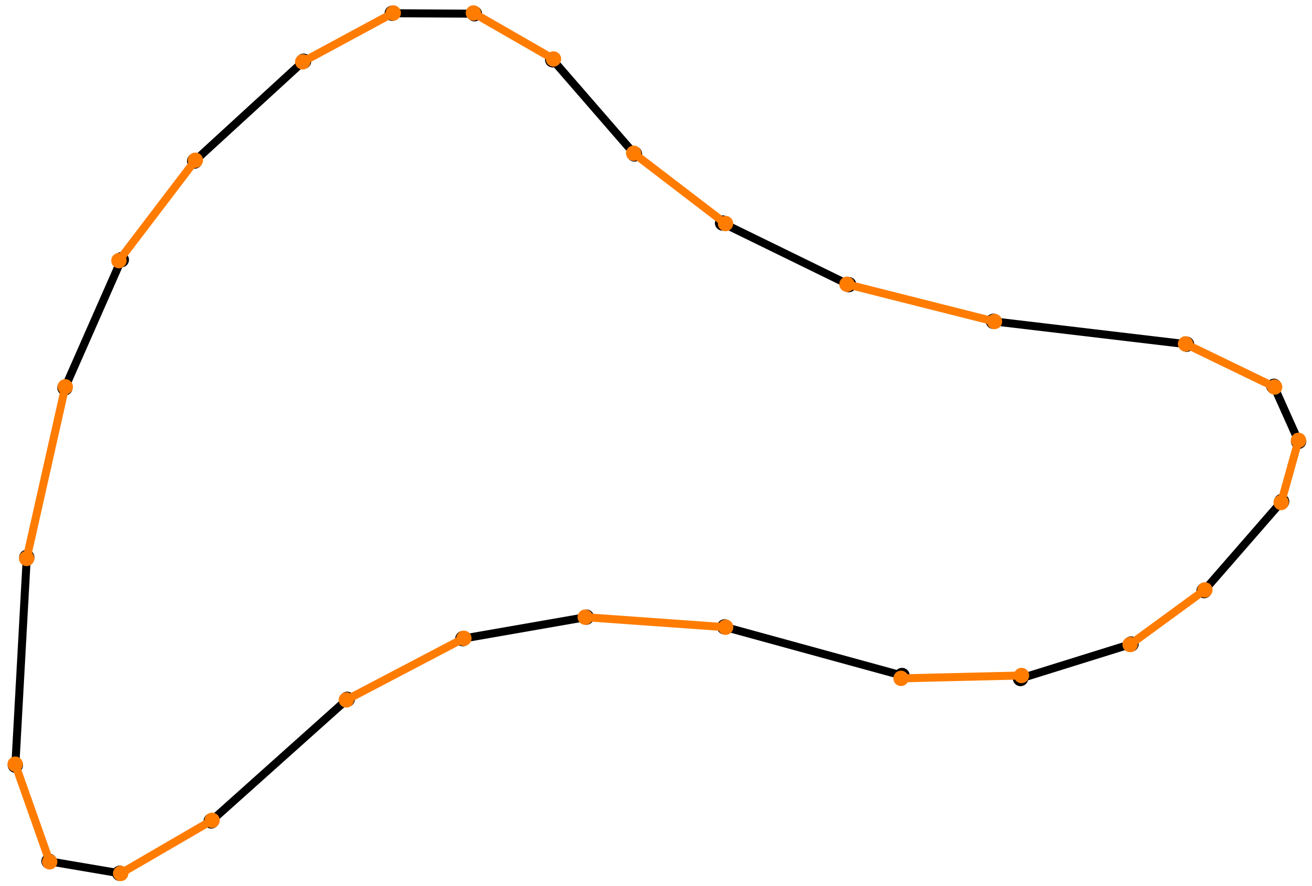
connect new vertices

Corner Cutting



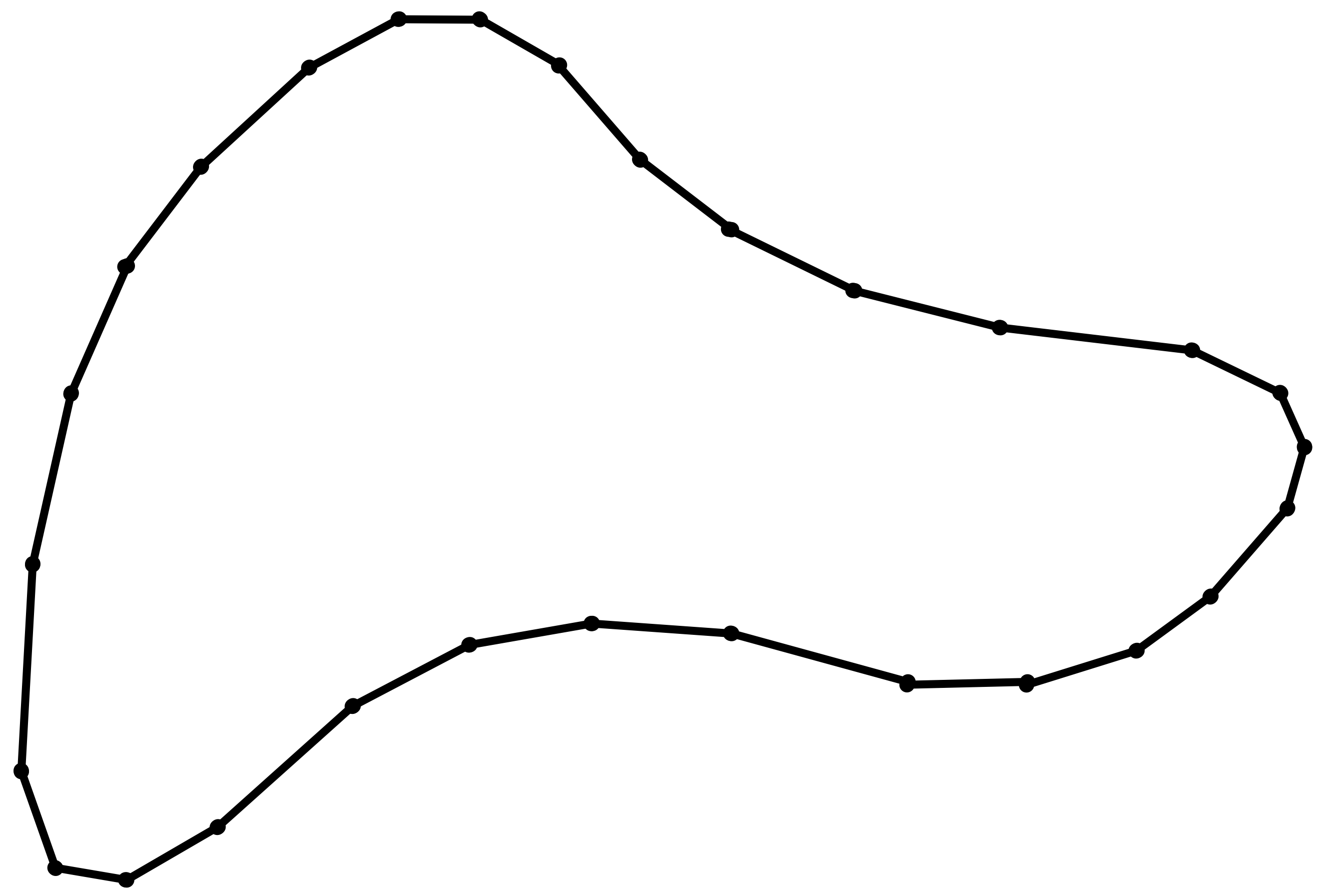
new control polygon

Corner Cutting



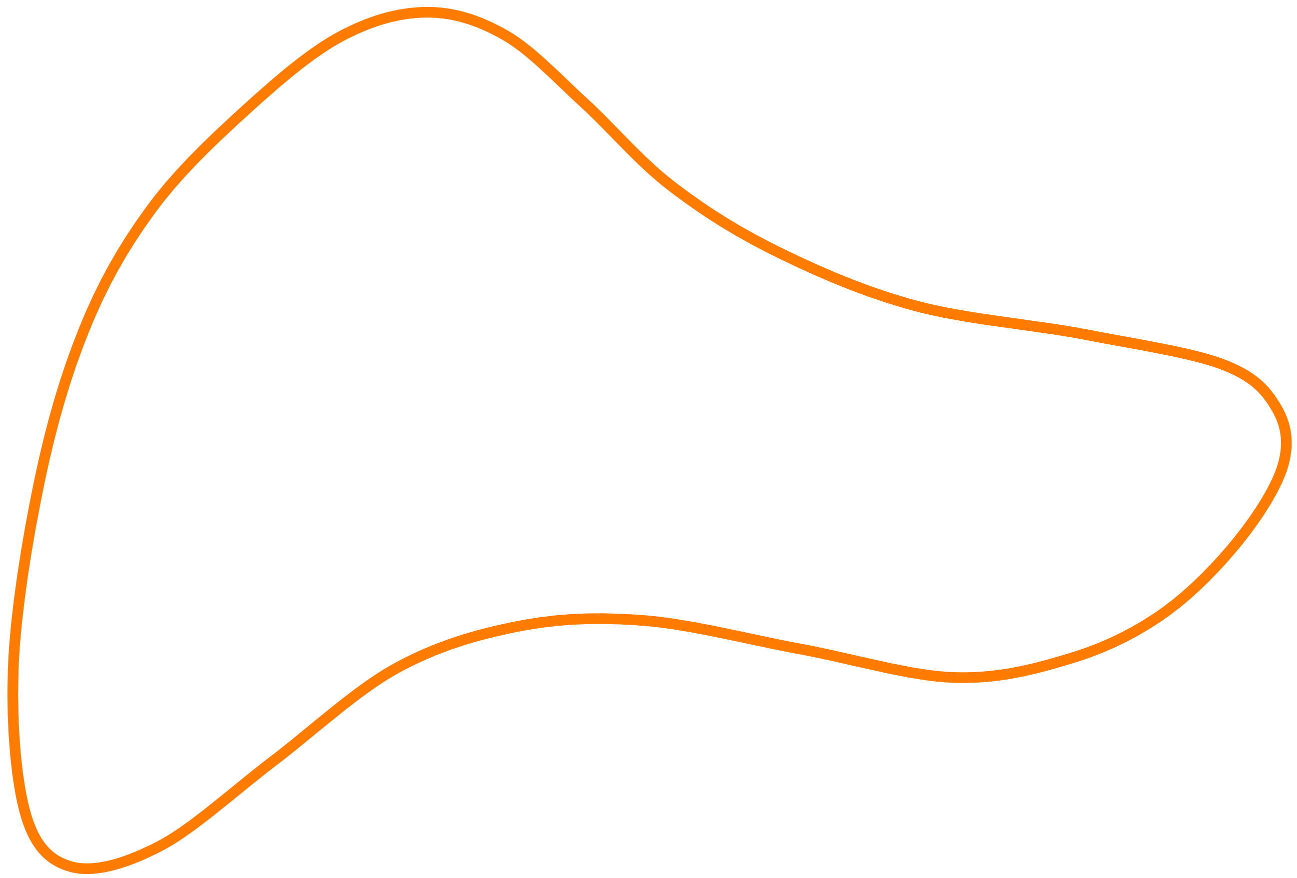
after another iteration

Corner Cutting



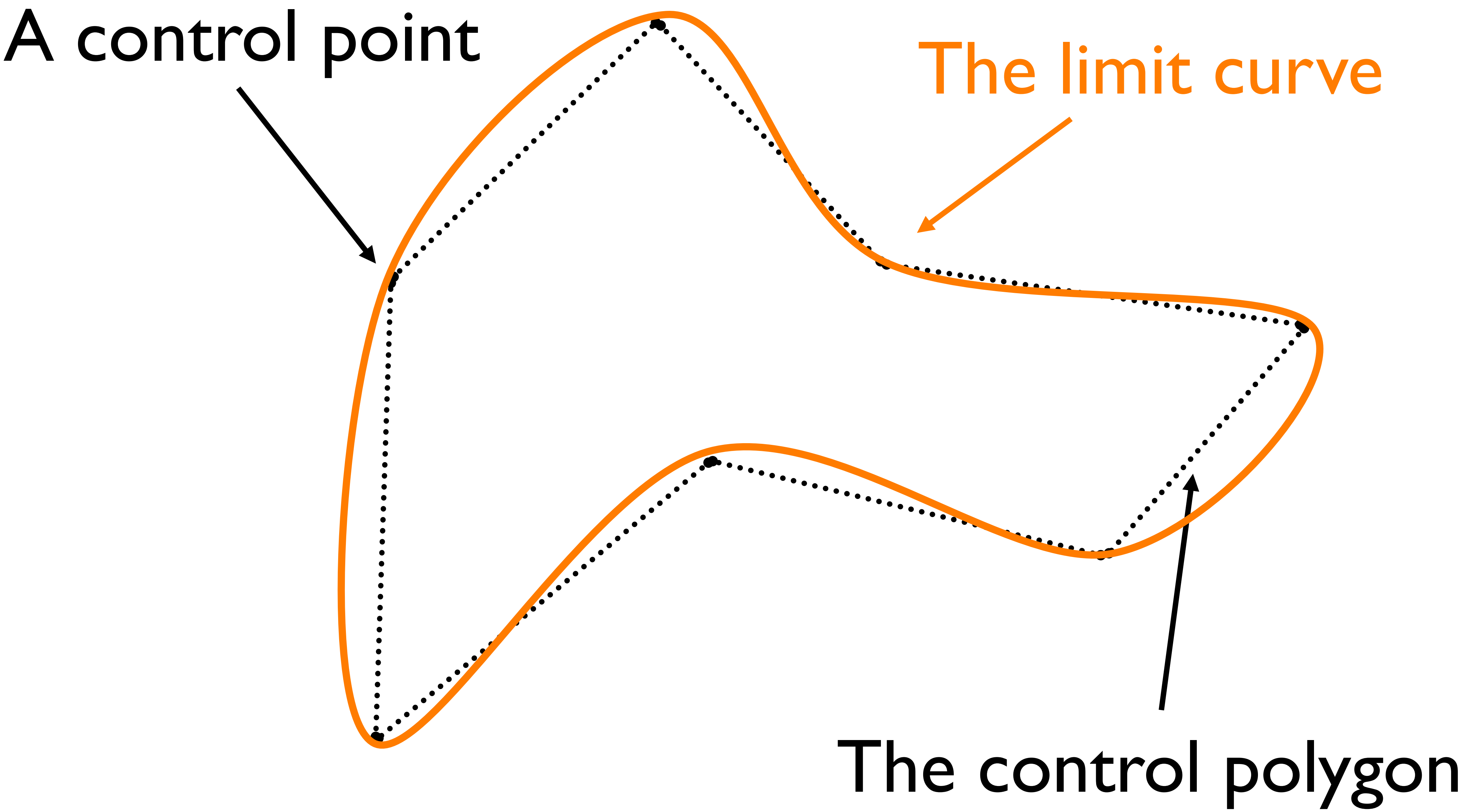
after another iteration

Corner Cutting

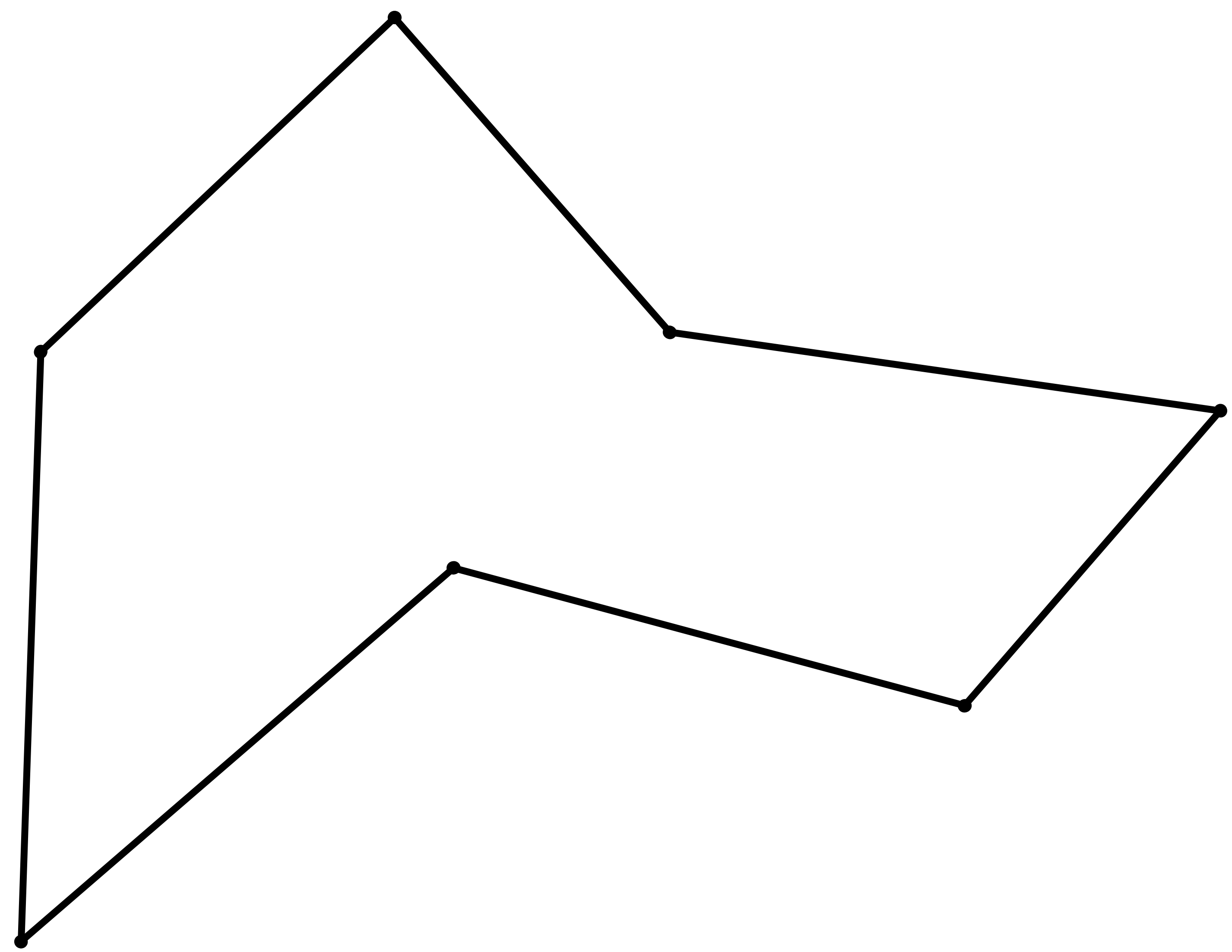


limit curve

Example 2: The 4-point scheme

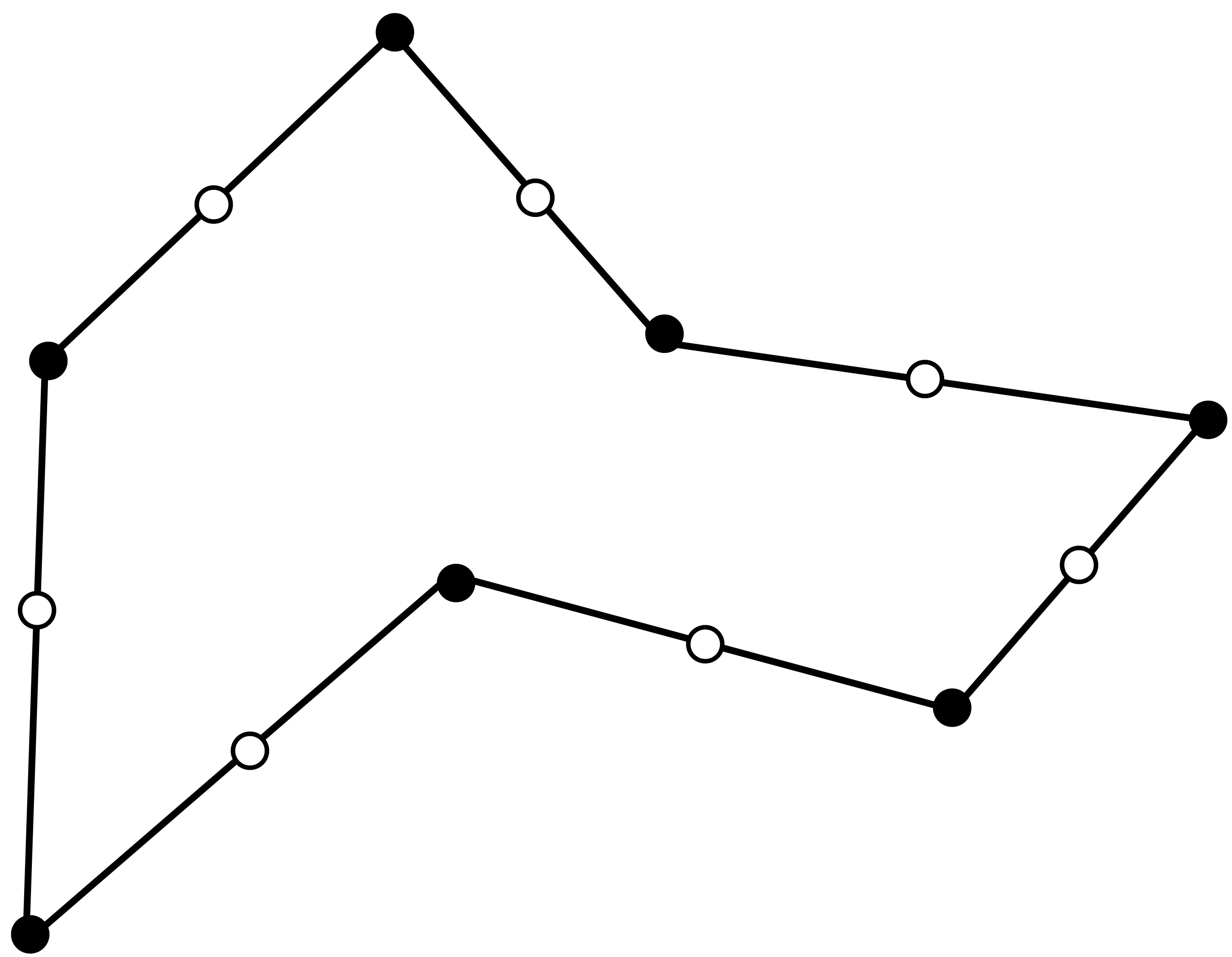


The 4-point scheme



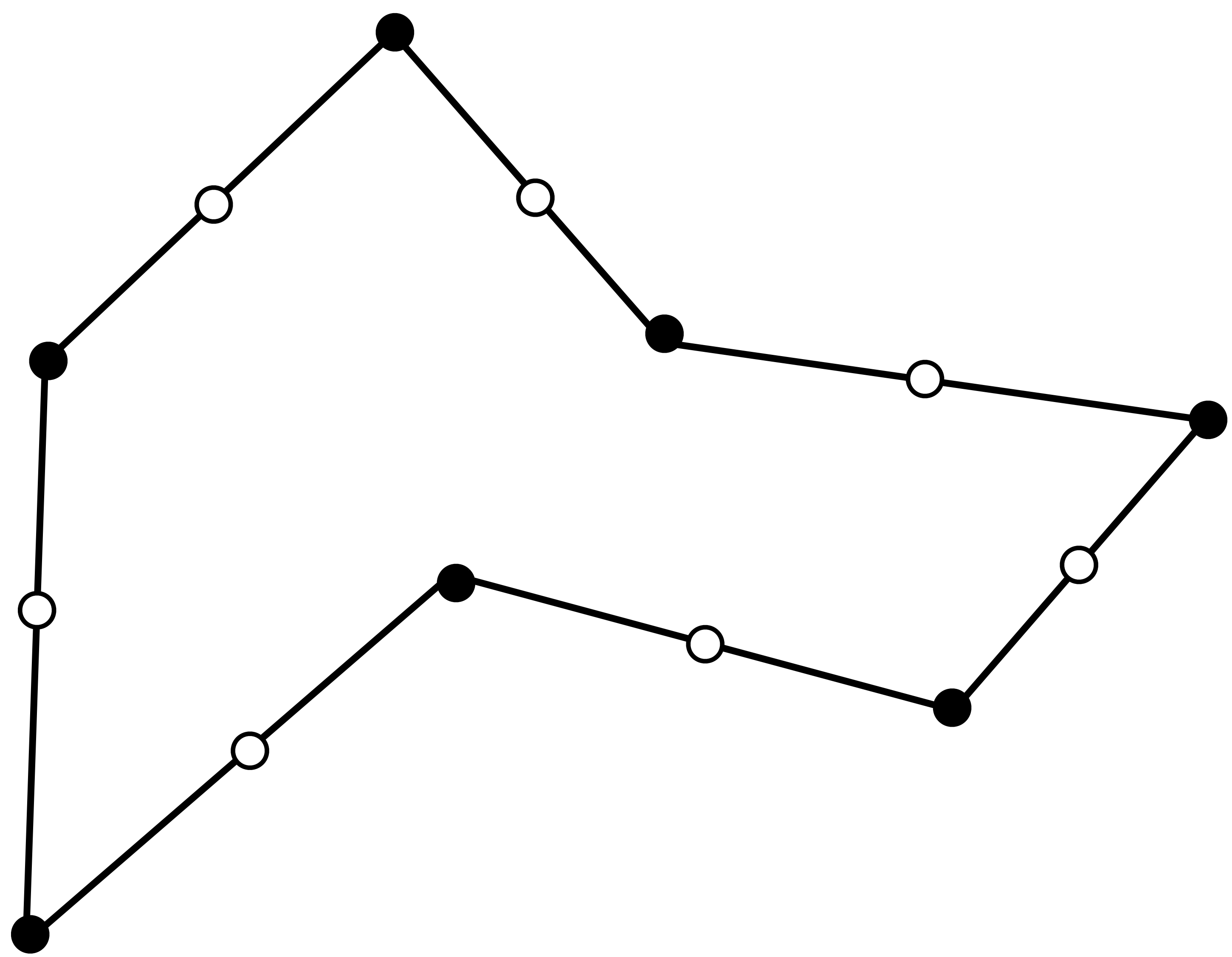
original control polygon

The 4-point scheme



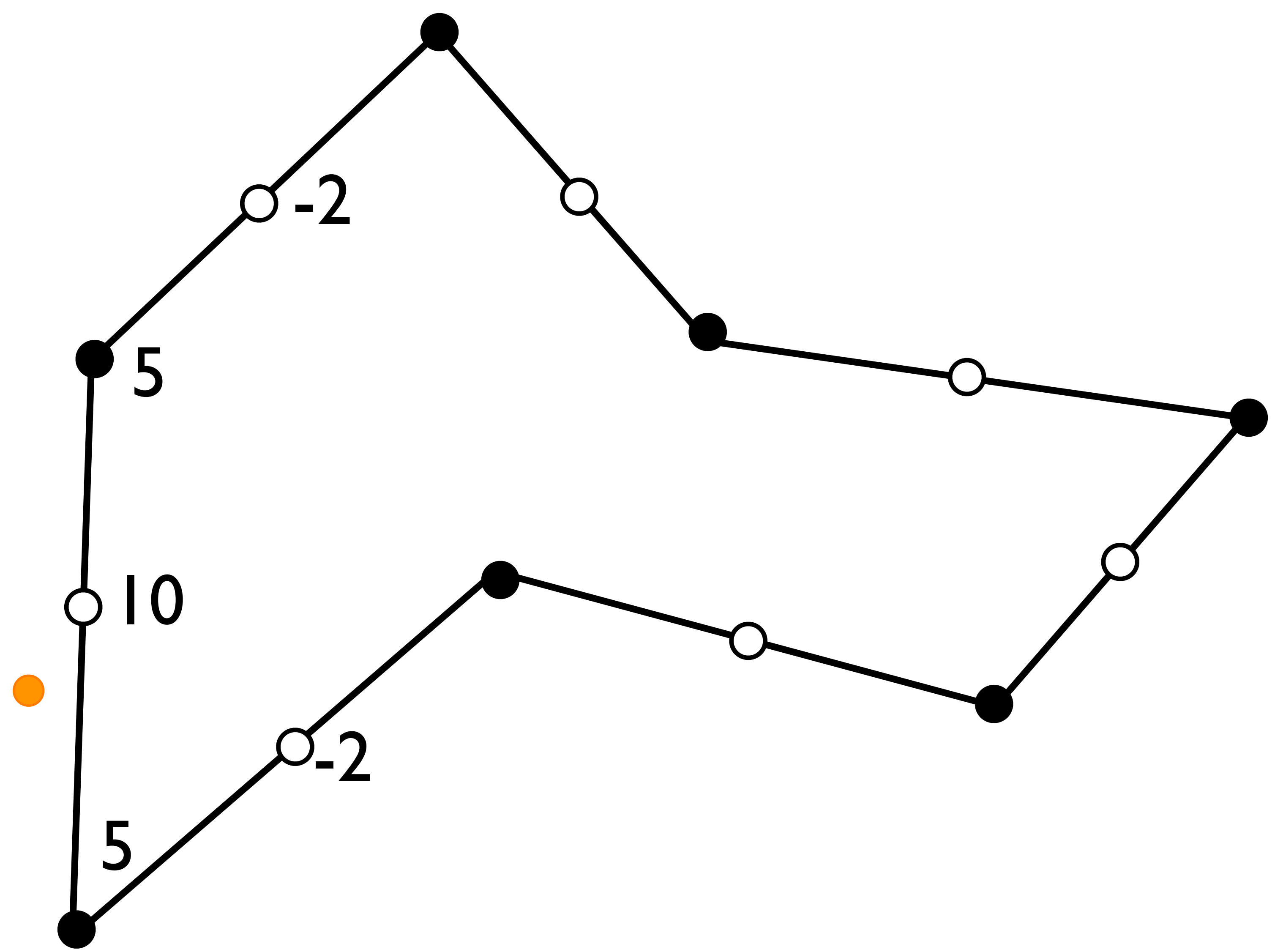
refined control polygon

The 4-point scheme



averaging mask: $[r_{-2}, r_{-1}, r_0, r_1, r_2] = (1/16)[-2,5,10,5,-2]$
applied only at odd (new) vertices!

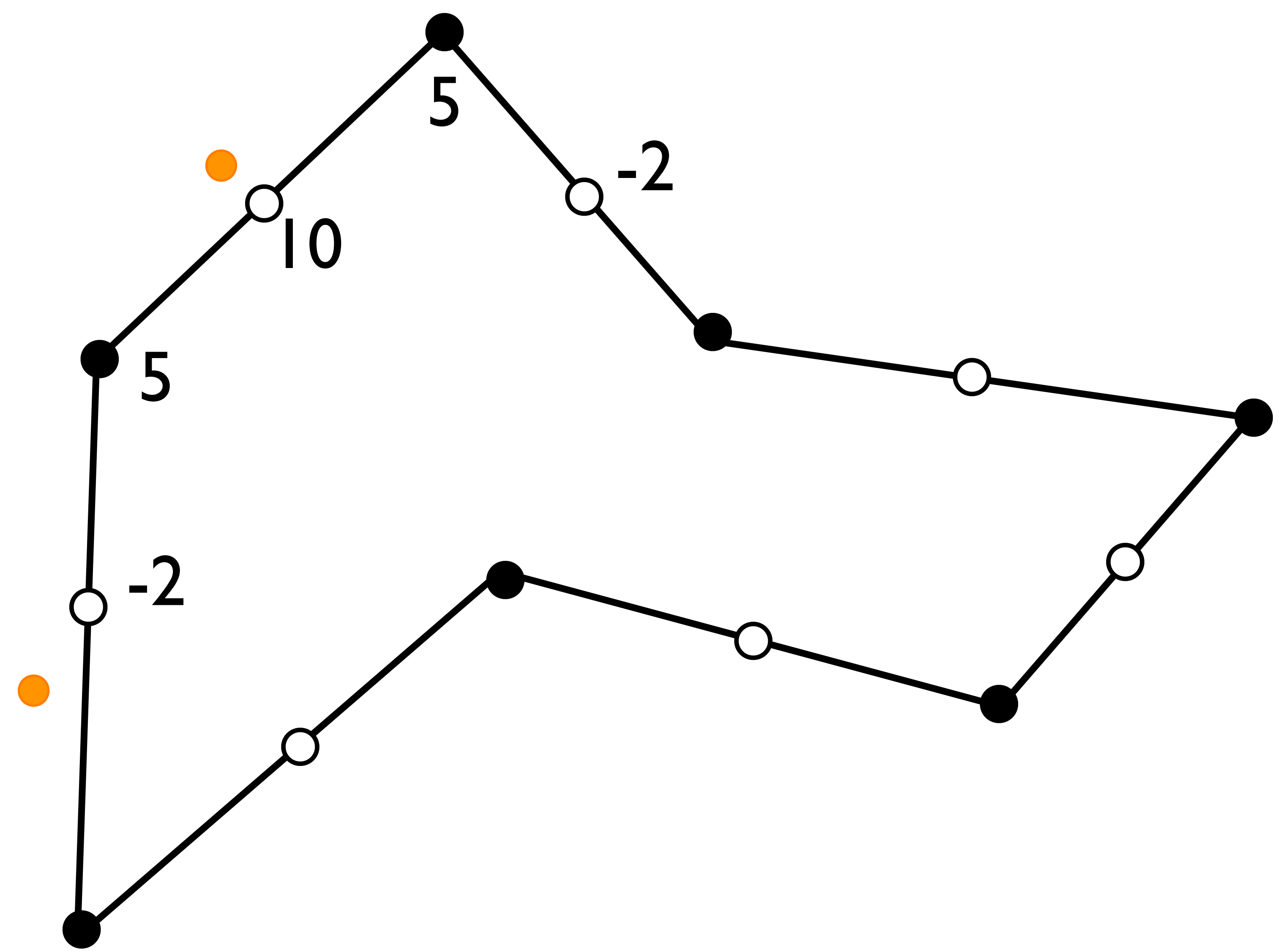
The 4-point scheme



averaging mask: $[r_{-2}, r_{-1}, r_0, r_1, r_2] = (1/16)[-2,5,10,5,-2]$

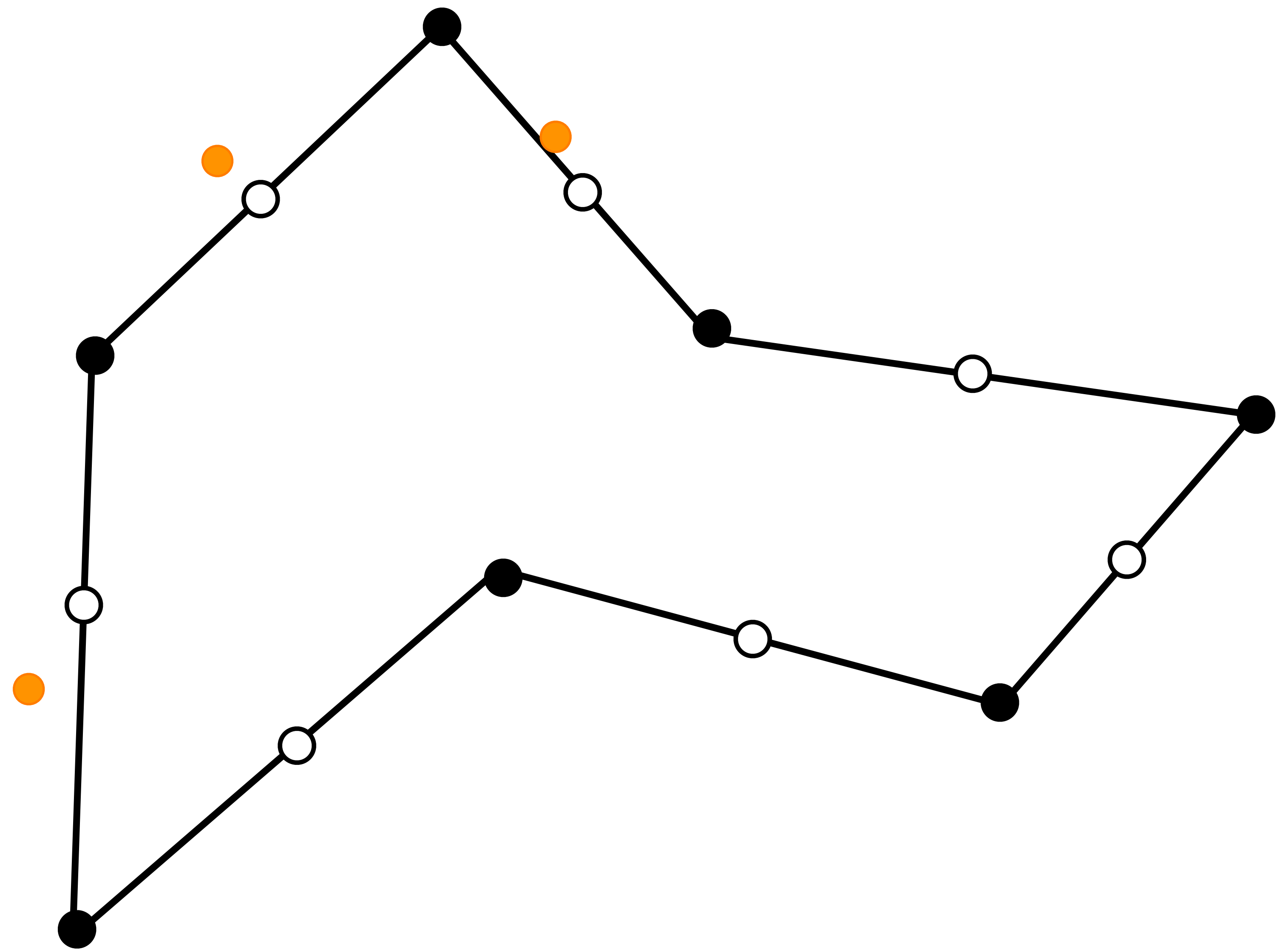
applied only at odd (new) vertices!

The 4-point scheme



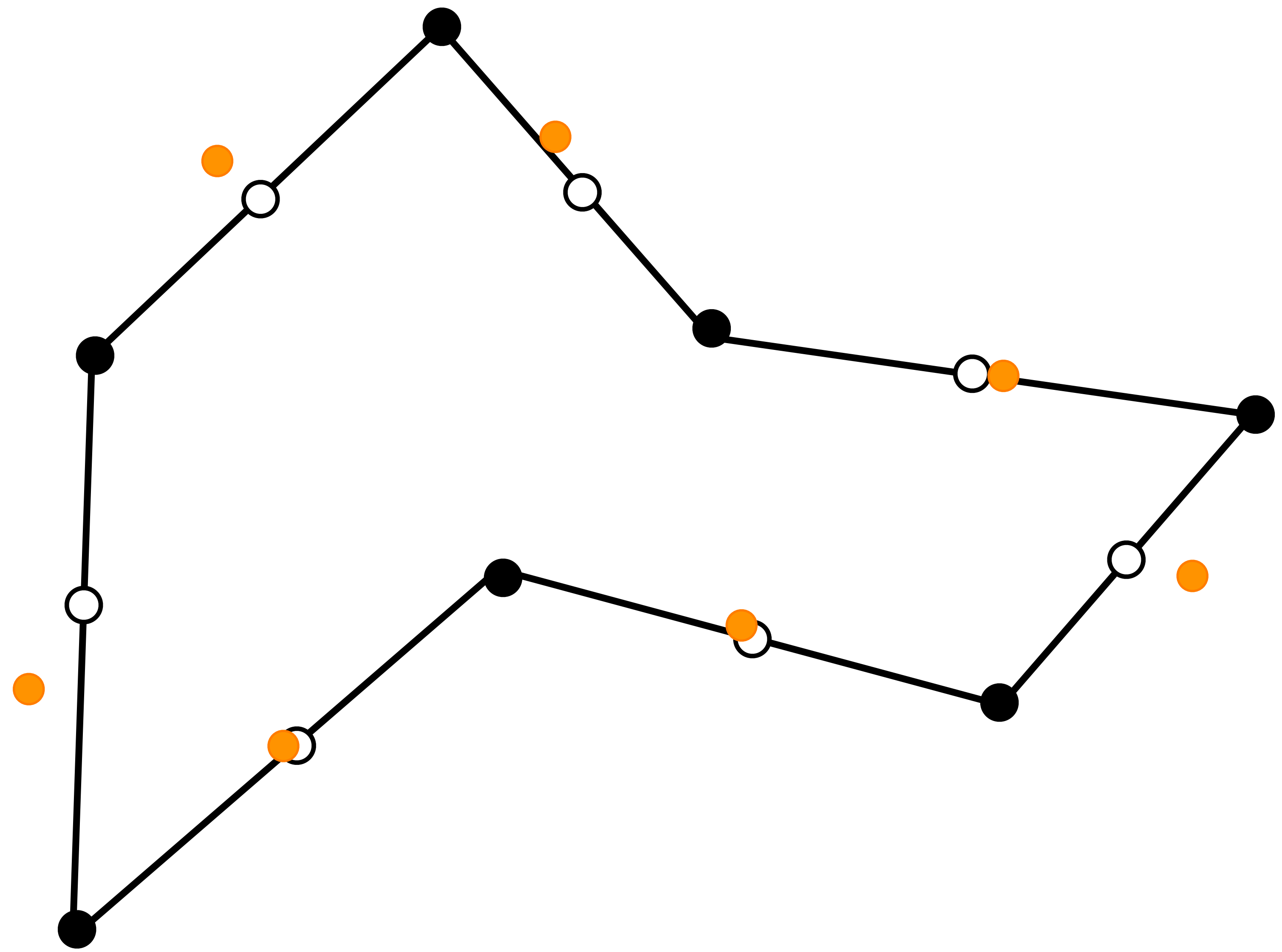
averaging mask: $[r_{-2}, r_{-1}, r_0, r_1, r_2] = (1/16)[-2,5,10,5,-2]$
applied only at odd (new) vertices!

The 4-point scheme



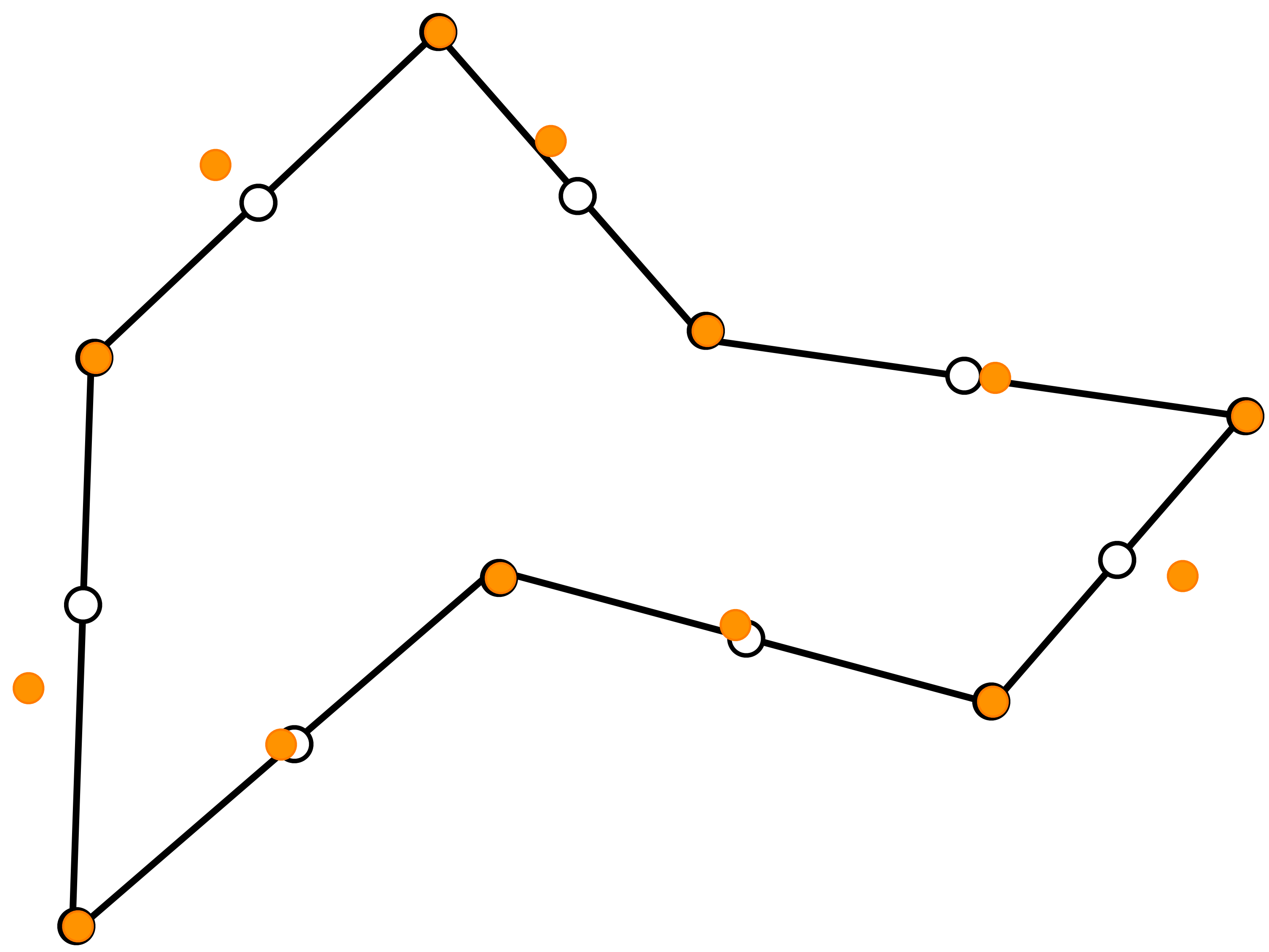
averaging mask: $[r_{-2}, r_{-1}, r_0, r_1, r_2] = (1/16)[-2,5,10,5,-2]$
applied only at odd (new) vertices!

The 4-point scheme



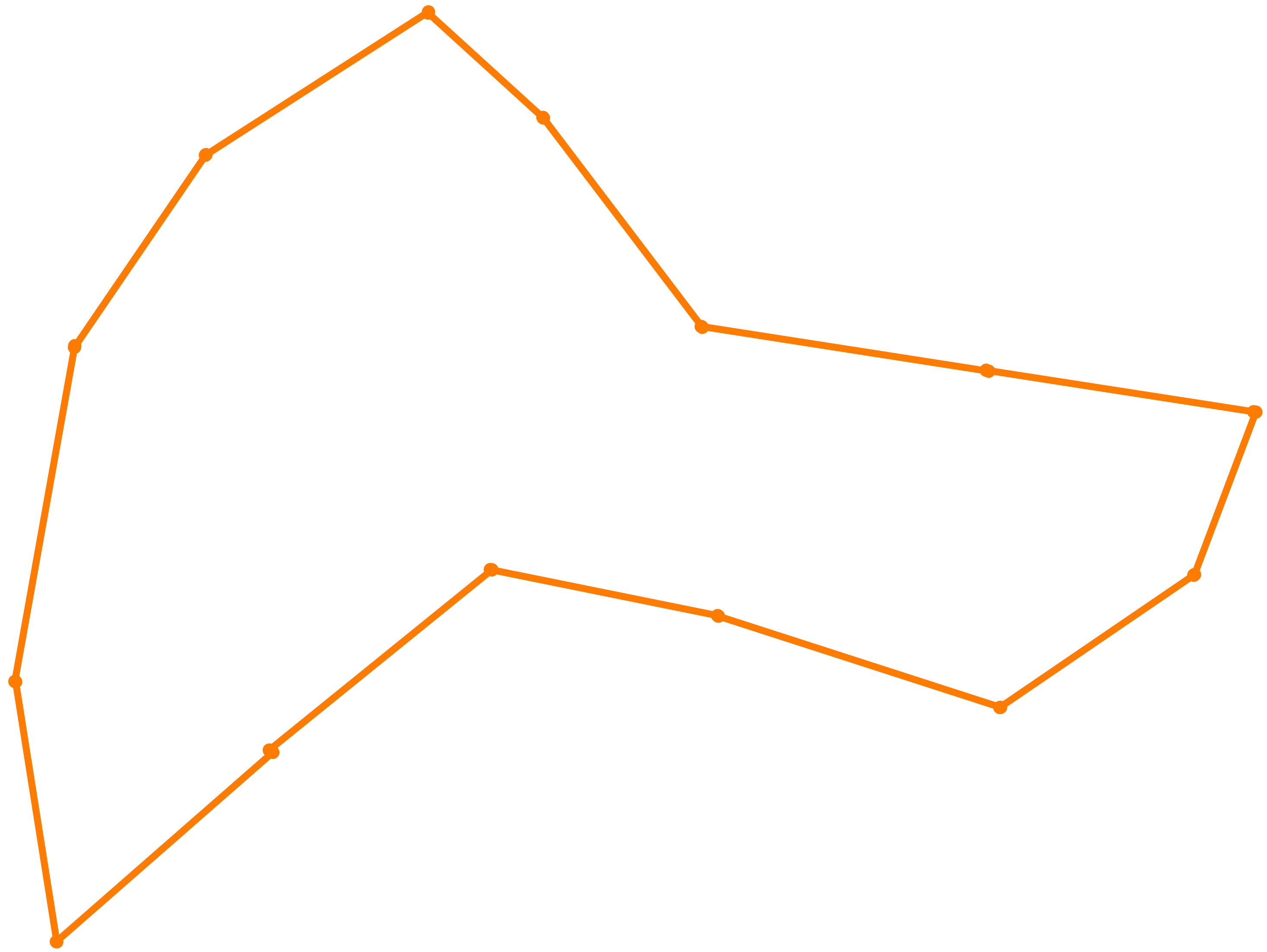
averaging mask: $[r_{-2}, r_{-1}, r_0, r_1, r_2] = (1/16)[-2,5,10,5,-2]$
applied only at odd (new) vertices!

The 4-point scheme



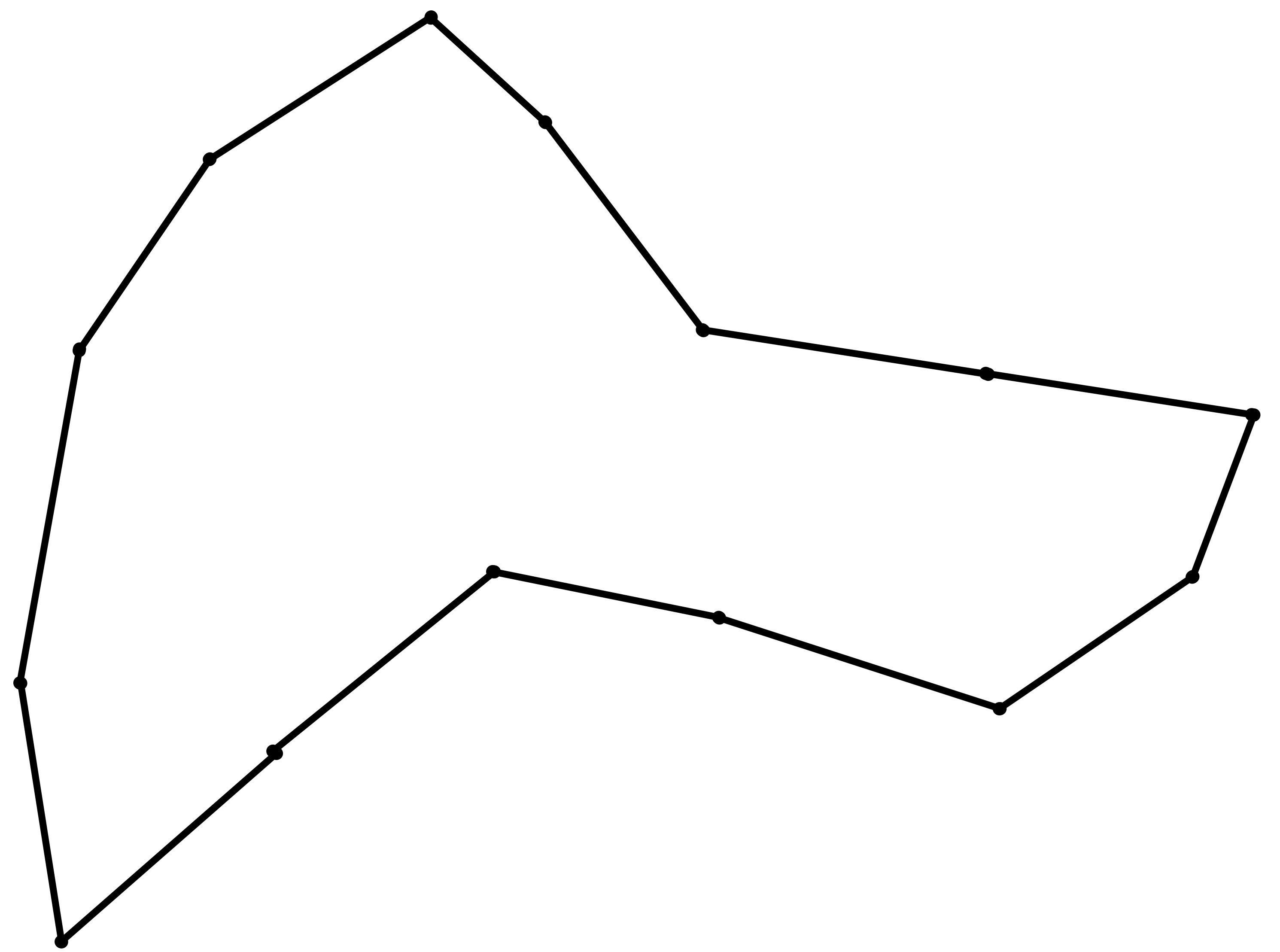
averaging mask: $[r_{-2}, r_{-1}, r_0, r_1, r_2] = (1/16)[-2,5,10,5,-2]$
applied only at odd (new) vertices!

The 4-point scheme



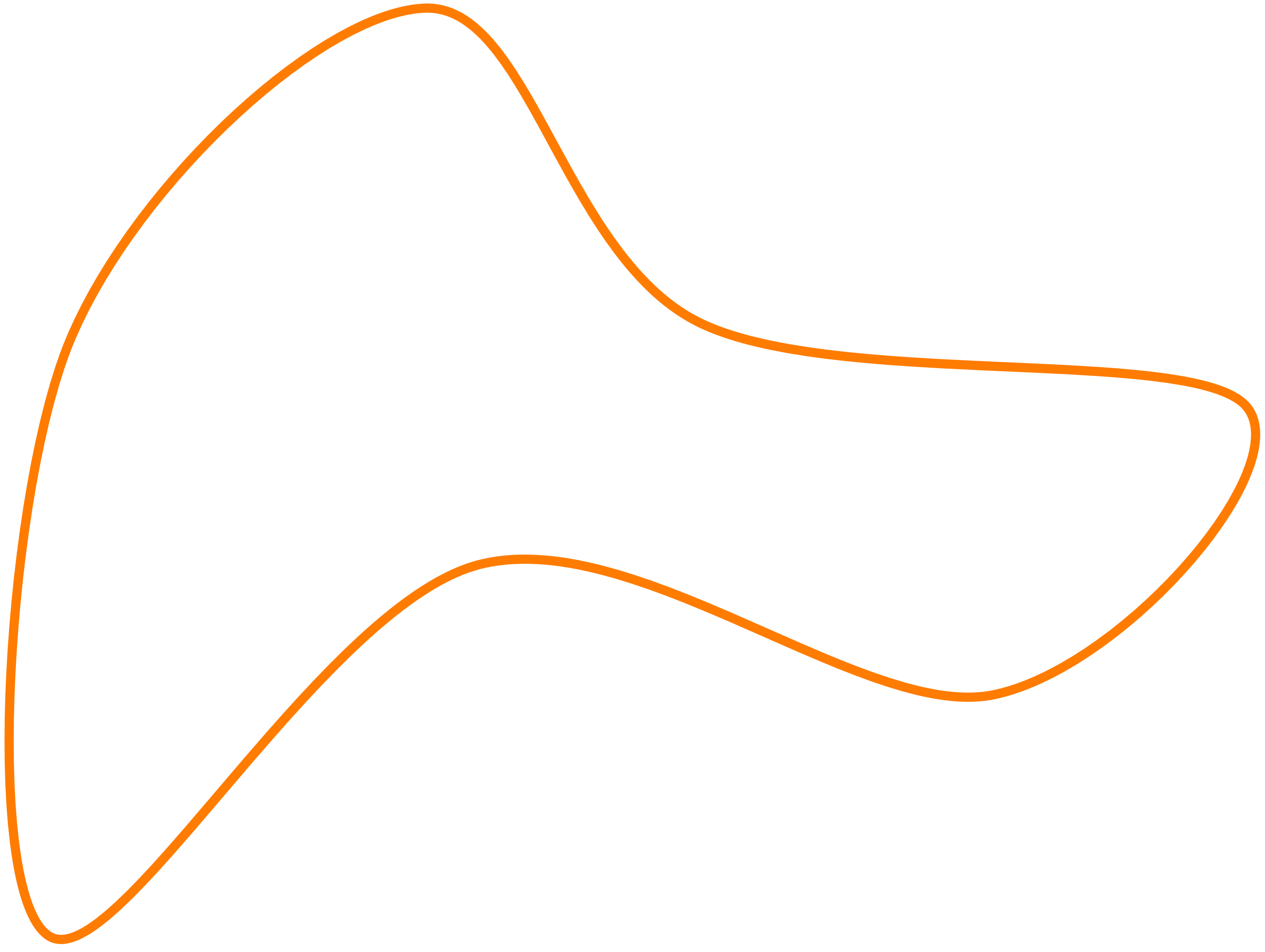
connect new vertices

The 4-point scheme



new control polygon

The 4-point scheme



limit curve

Examples

- All subdivision schemes share the splitting step, but differ in their averaging mask:
  - Corner-cutting - quadratic B-Spline:
    $r = (r_0, r_1) = \frac{1}{2}(1, 1)$
  - Cubic B-Spline: $r = (r_{-1}, r_0, r_1) = \frac{1}{2}(\frac{1}{2}, 1, \frac{1}{2})$
  - Fractal curve: $r = \frac{1}{2}(1+\sqrt{3}, 1-\sqrt{3})$

Analysis

- How do we know a subdivision scheme converges?
- What is the smoothness of the limit curve?
- Are there closed-form expressions for points on the limit curve?

Reminder: Eigenvalues and Eigenvectors (1)

- Right eigenvectors v_i with associated eigenvalues λ_i of matrix M satisfy:

$$Mv_i = \lambda_i v_i \qquad MV = \mathrm{diag}(\lambda_i)\, V$$

- A non-defective n×n matrix M has n linearly independent eigenvectors, forming a basis for the corresponding vector space.

- Any vector w may be expressed as a linear combination of the right eigenvectors:

$$w = \sum_{i=1}^{n} a_i v_i$$

- How can we obtain the linear coefficients a_i?

Reminder: Eigenvalues and Eigenvectors (2)

- Left eigenvectors u_i with associated eigenvalues λ_i of matrix M satisfy:

$$u_i M = \lambda_i u_i \qquad UM = U\mathrm{diag}(\lambda_i)$$

- Let V be the matrix whose columns are the right eigenvectors of M, and let U be the inverse of V. The rows of U are the left eigenvectors of M:

$$\begin{aligned} MV &= \mathrm{diag}(\lambda_i)V \\ UMVU &= U\mathrm{diag}(\lambda_i)\,VU \\ UM &= U\mathrm{diag}(\lambda_i) \end{aligned}$$

Reminder: Eigenvalues and Eigenvectors (3)

- Any vector w may be expressed as a linear combination of the right eigenvectors:

$$w = \sum_{i=1}^{n} a_i v_i$$

- Each linear coefficient a_j above may be found using the j-th left eigenvector:

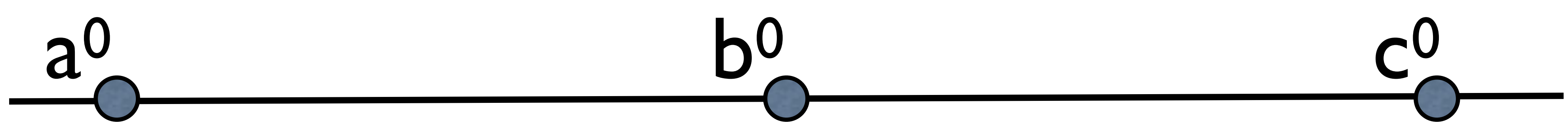
$$\begin{aligned} u_j\, w &= u_j \sum_{i=1}^{n} a_i v_i \\ &= \sum_{i=1}^{n} a_i\, u_j\, v_i \\ &= a_j \end{aligned}$$

Convergence Analysis (1)

- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25 \;\; 0.5 \;\; 0.25]$

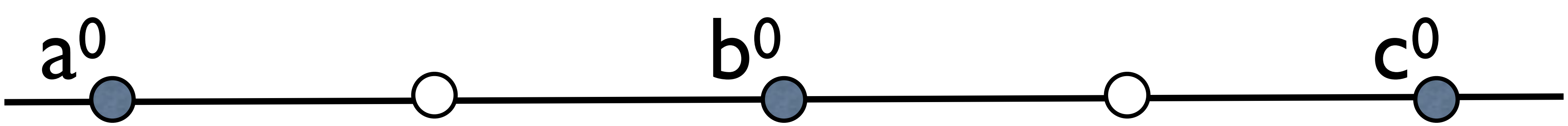
Convergence Analysis (1)

- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25 \;\; 0.5 \;\; 0.25]$



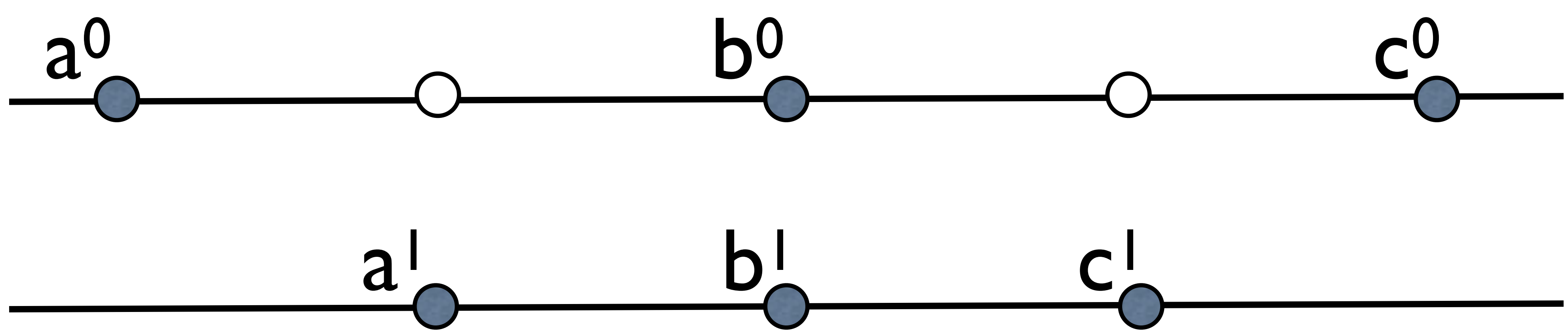
Convergence Analysis (1)

- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25 \;\; 0.5 \;\; 0.25]$



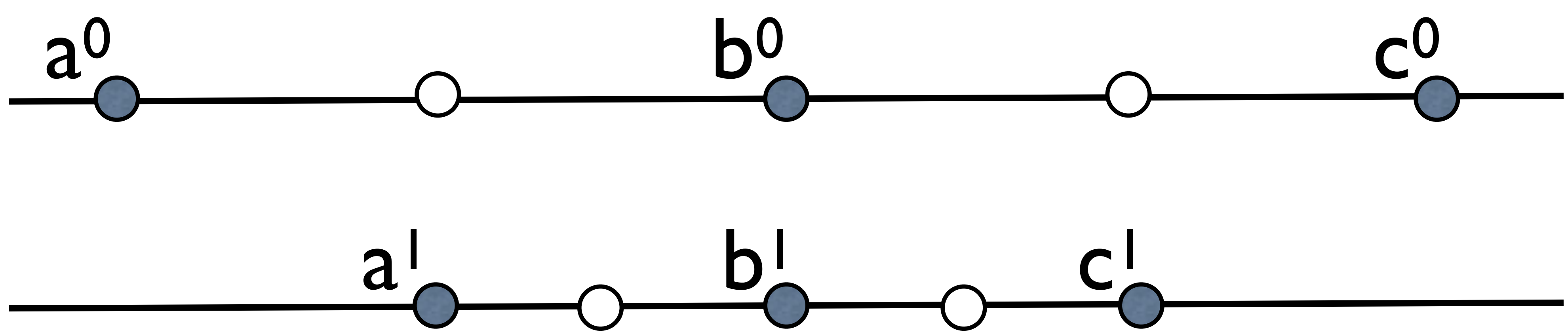
Convergence Analysis (1)

- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25\ \ 0.5\ \ 0.25]$



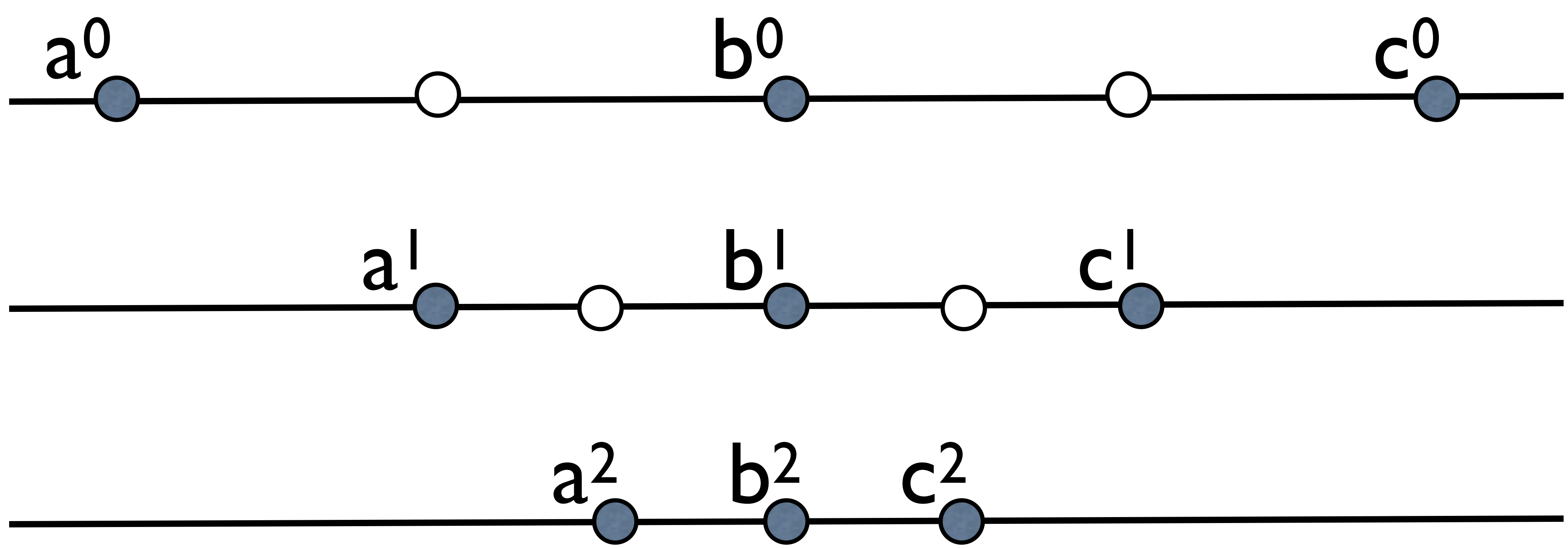
Convergence Analysis (1)

- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25\ \ 0.5\ \ 0.25]$



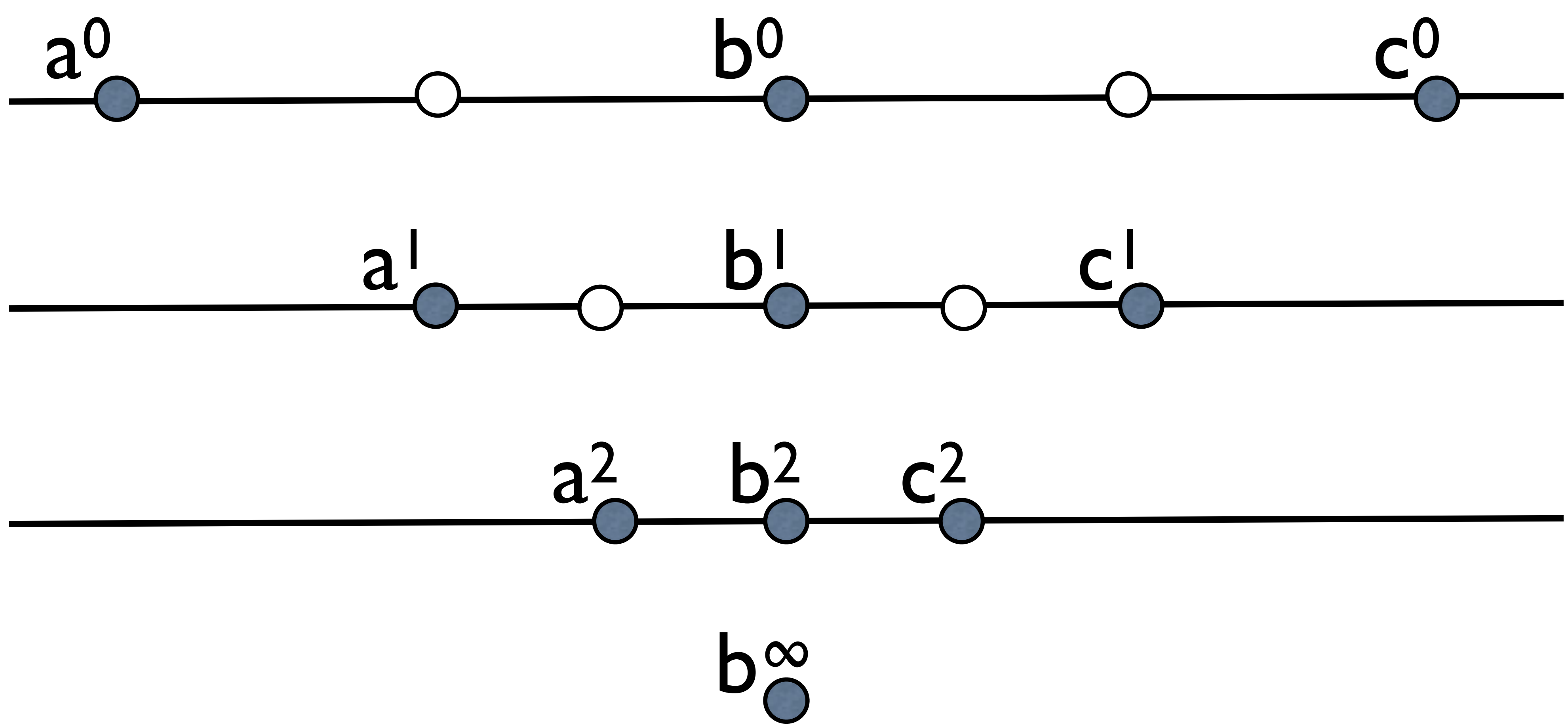
Convergence Analysis (1)

- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25 \;\; 0.5 \;\; 0.25]$

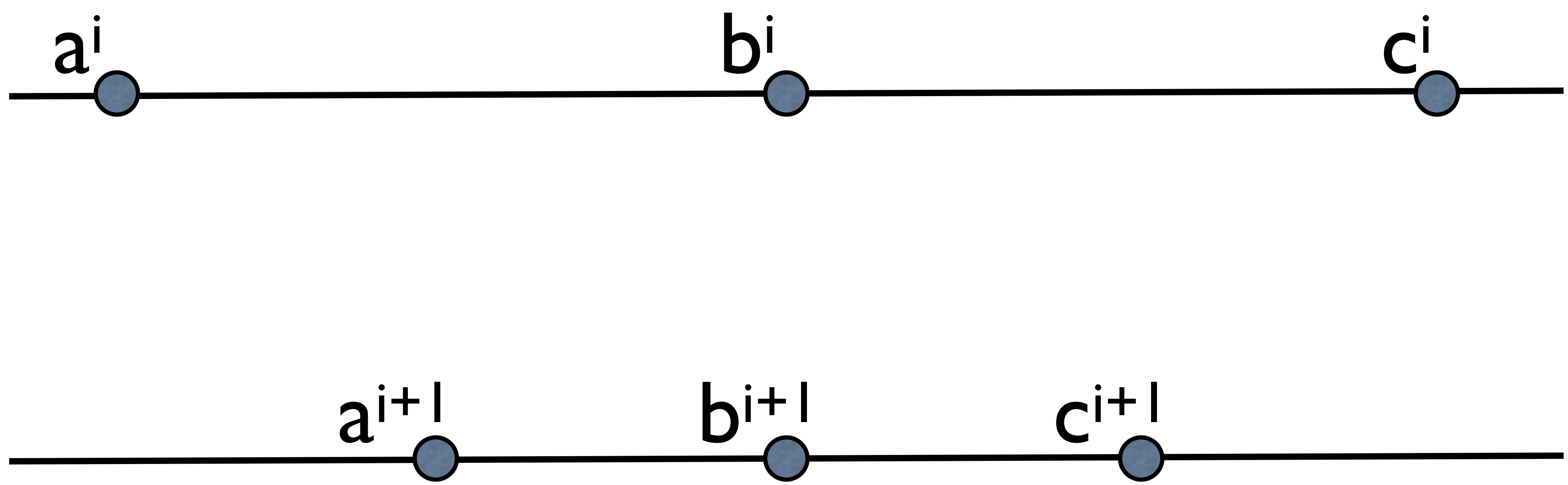


Convergence Analysis (1)

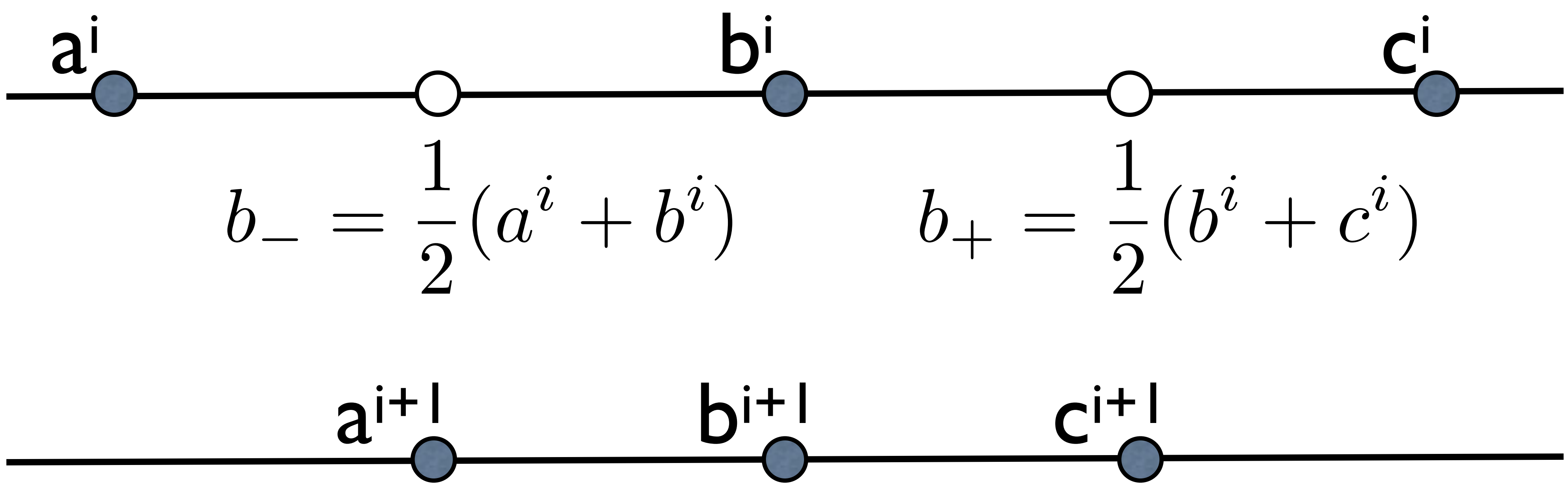
- Let's try to analyze the convergence of the (approximating) cubic B-spline subdivision scheme.
- Averaging mask: $[r_{-1}, r_0, r_1] = [0.25 \;\; 0.5 \;\; 0.25]$



Convergence Analysis (2)

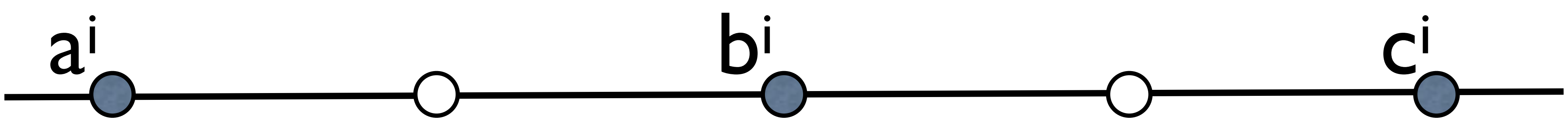


Convergence Analysis (2)

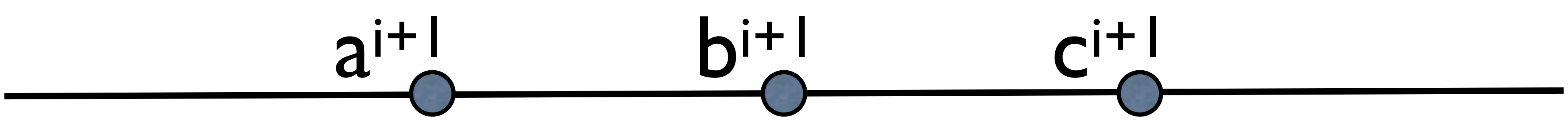


$$b_- = \frac{1}{2}(a^i + b^i) \qquad b_+ = \frac{1}{2}(b^i + c^i)$$

Convergence Analysis (2)

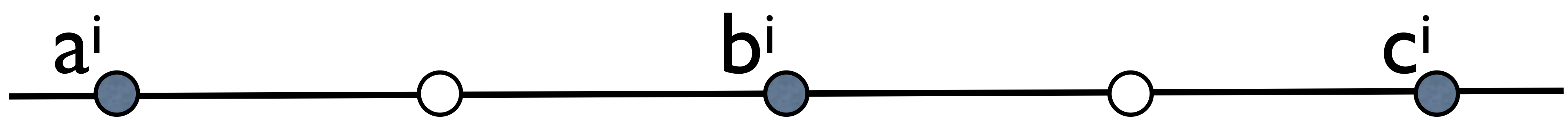


$$b_- = \frac{1}{2}(a^i + b^i) \qquad b_+ = \frac{1}{2}(b^i + c^i)$$

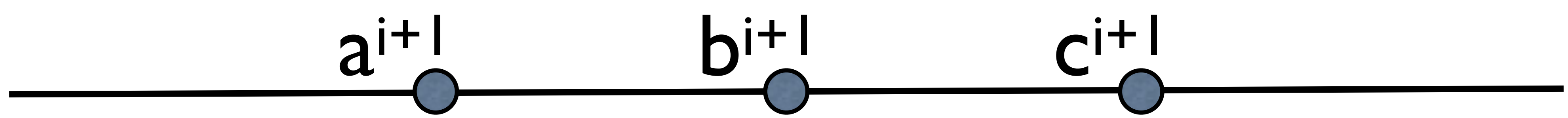


$$a^{i+1} = \frac{1}{4}a^i + \frac{1}{2}b_- + \frac{1}{4}b^i = \frac{1}{2}(a^i + b^i)$$

Convergence Analysis (2)



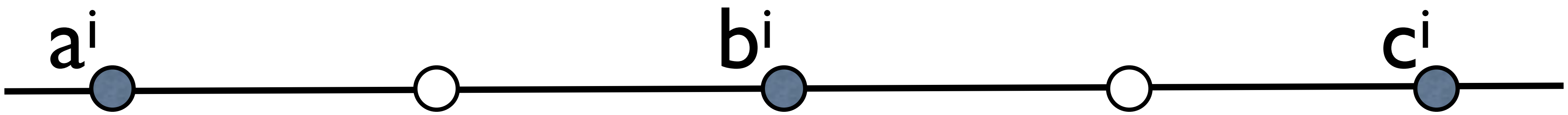
$$b_- = \frac{1}{2}(a^i + b^i) \qquad b_+ = \frac{1}{2}(b^i + c^i)$$



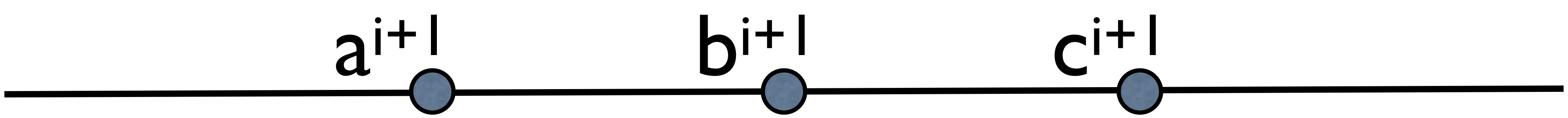
$$a^{i+1} = \frac{1}{4}a^i + \frac{1}{2}b_- + \frac{1}{4}b^i = \frac{1}{2}(a^i + b^i)$$

$$b^{i+1} = \frac{1}{4}b_- + \frac{1}{2}b^i + \frac{1}{4}b_+ = \frac{1}{8}(a^i + 6b^i + c^i)$$

Convergence Analysis (2)



$$b_- = \frac{1}{2}(a^i + b^i) \qquad b_+ = \frac{1}{2}(b^i + c^i)$$



$$a^{i+1} = \frac{1}{4}a^i + \frac{1}{2}b_- + \frac{1}{4}b^i = \frac{1}{2}(a^i + b^i)$$

$$b^{i+1} = \frac{1}{4}b_- + \frac{1}{2}b^i + \frac{1}{4}b_+ = \frac{1}{8}(a^i + 6b^i + c^i)$$

$$c^{i+1} = \frac{1}{4}b^i + \frac{1}{2}b_+ + \frac{1}{4}c^i = \frac{1}{2}(b^i + c^i)$$

Convergence Analysis (3)

- The subdivision process at the point b^i may be described using a matrix-vector multiplication:

$$L \begin{bmatrix} a^i \\ b^i \\ c^i \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{bmatrix} \begin{bmatrix} a^i \\ b^i \\ c^i \end{bmatrix} = \begin{bmatrix} a^{i+1} \\ b^{i+1} \\ c^{i+1} \end{bmatrix}$$

Convergence Analysis (3)

- The subdivision process at the point b^i may be described using a matrix-vector multiplication:

$$L \begin{bmatrix} a^i \\ b^i \\ c^i \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{bmatrix} \begin{bmatrix} a^i \\ b^i \\ c^i \end{bmatrix} = \begin{bmatrix} a^{i+1} \\ b^{i+1} \\ c^{i+1} \end{bmatrix}$$

$$\begin{bmatrix} a^k \\ b^k \\ c^k \end{bmatrix} = L \begin{bmatrix} a^{k-1} \\ b^{k-1} \\ c^{k-1} \end{bmatrix} = L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix}$$

Convergence Analysis (4)

- We are looking for the limit: $$\begin{bmatrix} a^{\infty} \\ b^{\infty} \\ c^{\infty} \end{bmatrix} = \lim_{k \to \infty} L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix}$$
- We will find it using the eigenvalues and the eigenvectors of the subdivision matrix L!
- The (right) eigenvectors and their eigenvalues of L are:

$$v_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad v_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \qquad v_3 = \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}$$

$$\lambda_1 = 1 \qquad \lambda_2 = \frac{1}{2} \qquad \lambda_3 = \frac{1}{4}$$

Convergence Analysis (5)

- These eigenvectors form a basis, therefore:

$$\begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$$

$$\lim_{k \to \infty} L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \alpha_1 v_1 = \begin{bmatrix} \alpha_1 \\ \alpha_1 \\ \alpha_1 \end{bmatrix}$$

Convergence Analysis (5)

- These eigenvectors form a basis, therefore:

$$\begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$$

$$\begin{aligned} L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} &= \alpha_1 L^k v_1 + \alpha_2 L^k v_2 + \alpha_3 L^k v_3 \\ &= \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \alpha_3 \lambda_3^k v_3 \end{aligned}$$

$$\lim_{k \to \infty} L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \alpha_1 v_1 = \begin{bmatrix} \alpha_1 \\ \alpha_1 \\ \alpha_1 \end{bmatrix}$$

Convergence Analysis (5)

- These eigenvectors form a basis, therefore:

$$\begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$$

$$\begin{aligned} L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} &= \alpha_1 L^k v_1 + \alpha_2 L^k v_2 + \alpha_3 L^k v_3 \\ &= \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \alpha_3 \lambda_3^k v_3 \end{aligned}$$

$$\lim_{k \to \infty} L^k \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \alpha_1 v_1 = \begin{bmatrix} \alpha_1 \\ \alpha_1 \\ \alpha_1 \end{bmatrix}$$

$$\boxed{\lim_{k \to \infty} b^k \quad = \quad \alpha_1}$$

Convergence Analysis (6)

- Let's look at the **left** eignevectors of L: u_1, u_2, u_3
- In particular:

$$u_1 = \frac{1}{6}[1, 4, 1]$$

$$\alpha_1 = u_1 \cdot \begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 \end{bmatrix}\begin{bmatrix} a^0 \\ b^0 \\ c^0 \end{bmatrix}$$

- So, we have a simple closed-form expression for the limit point:

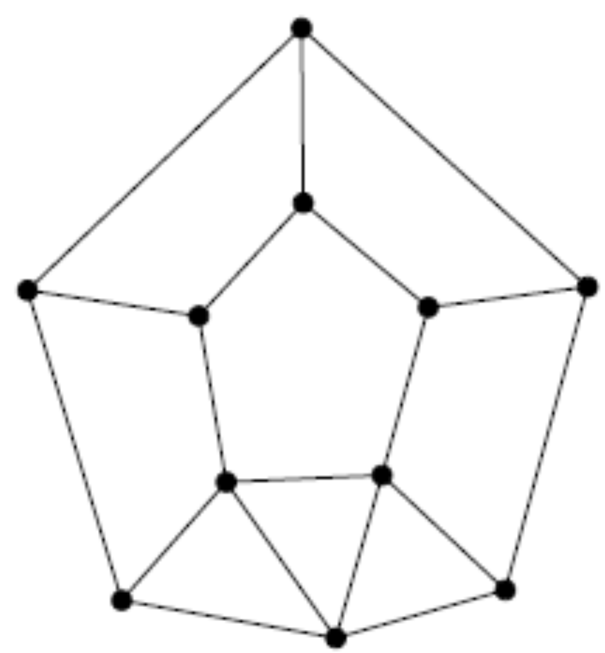
$$b^\infty = \alpha_1 = \frac{1}{6}(a^0 + 4b^0 + c^0)$$

Control Mesh Refinement

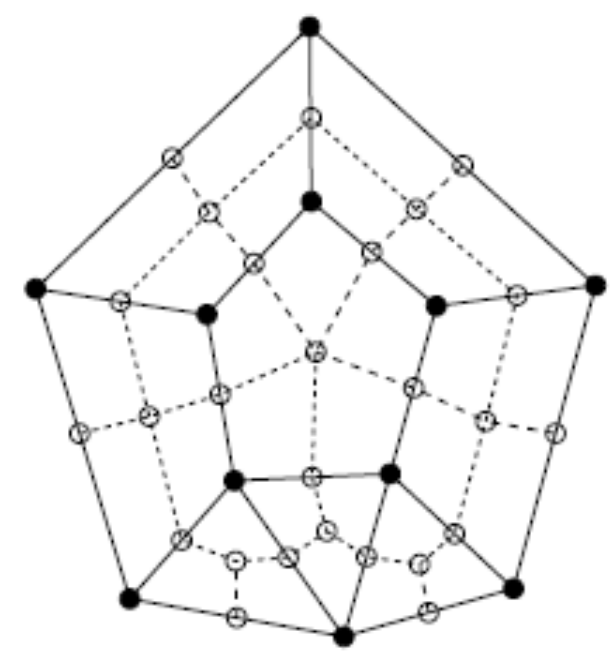
- Two steps:
  - *Splitting step:* introduce new vertices, edges, and faces into the mesh, resulting in a refined mesh.
  - *Averaging step:* compute a new location for each vertex in the refined mesh by averaging its local neighborhood.

Catmull-Clark Subdivision

- Splitting step: a new vertex is introduced in the middle of each edge and inside each face.



Original

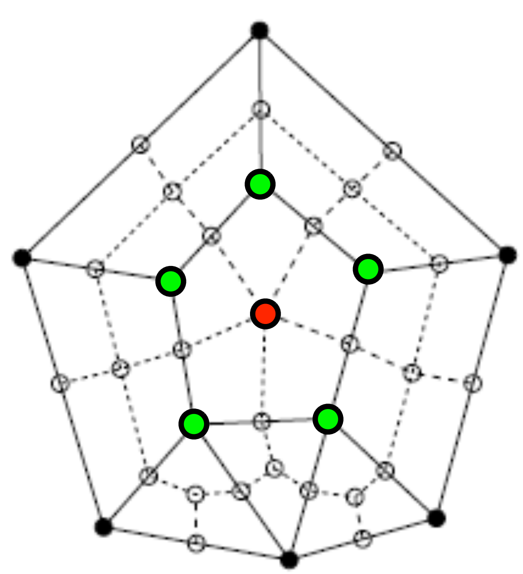


After splitting

- After one subdivision step, all faces have four vertices (quads).

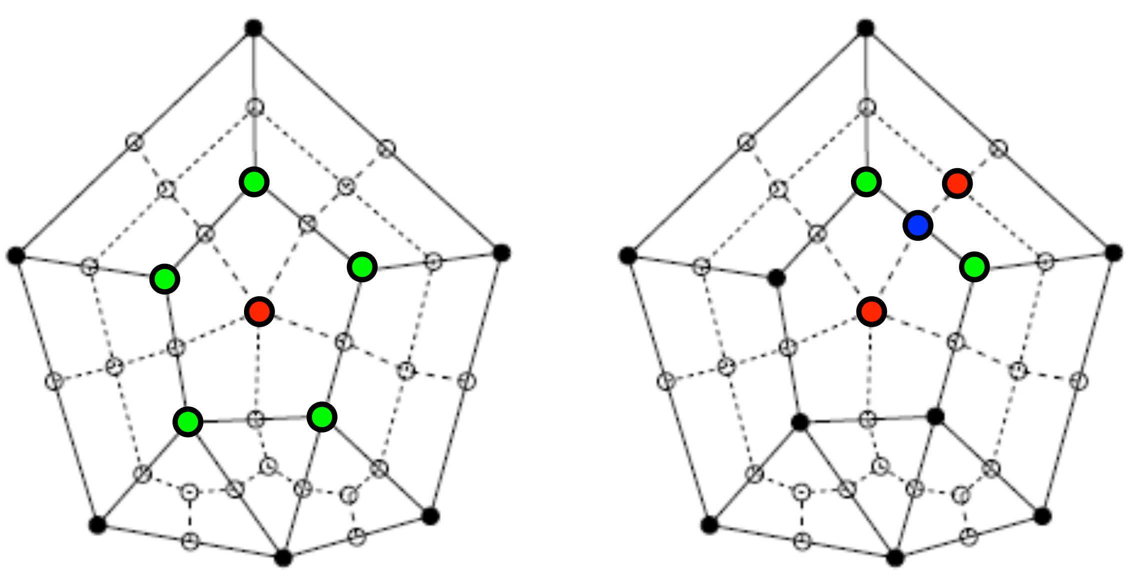
Catmull-Clark Subdivision

- Each “face vertex” gets the average of the face corners.



Catmull-Clark Subdivision

- Each “face vertex” gets the average of the face corners.
- Each “edge vertex” gets the average of its endpoints and adjacent “face vertices”.



Catmull-Clark Subdivision

- Each "face vertex" gets the average of the face corners.
- Each "edge vertex" gets the average of its endpoints and adjacent "face vertices".
- Each other vertex gets the average of it's old position, and adjacent "face" and "edge" vertices.