CPSC 424 - Tutorial 5

February 10th, 2025

TA: Suzuran Takikawa (stakikaw@cs.ubc.ca)
Credits to our awesome former TA Chenxi Liu for much of these slides!

Eigenvalues and Eigenvectors

Given a square matrix $A \in \mathbb{R}^{n \times n}$, x is an eigenvector of A if

$$Ax = \lambda x,$$
$$x \neq 0$$

λ is the eigenvalue corresponding to x. (λ, x) is an eigenpair.

Non-defective Matrix

Let $A \in \mathbb{R}^{n \times n}$, have eigenpairs $(\lambda_1, x_1), (\lambda_2, x_2), \ldots, (\lambda_n, x_n)$, and $\{x_1, x_2, \ldots, x_n\}$ forms a basis of $\mathbb{R}^n$.

Non-defective Matrix

Let $A \in \mathbb{R}^{n\times n}$, have eigenpairs $(\lambda_1, x_1), (\lambda_2, x_2), \ldots, (\lambda_n, x_n)$, and $\{x_1, x_2, \ldots, x_n\}$ forms a basis of $\mathbb{R}^n$.

A is called a non-defective or diagonalizable matrix.

Diagonal matrix

$$
\begin{aligned}
A\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_n\end{bmatrix} &= \begin{bmatrix}A\mathbf{x}_1 & A\mathbf{x}_2 & \ldots & A\mathbf{x}_n\end{bmatrix} \\
&= \begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \ldots & \lambda_n\mathbf{x}_n\end{bmatrix} \\
&= \underbrace{\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_n\end{bmatrix}}_{X} \underbrace{\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{bmatrix}}_{\Lambda}.
\end{aligned}
$$

Non-defective Matrix

Let $A \in \mathbb{R}^{n \times n}$, have eigenpairs $(\lambda_1, x_1), (\lambda_2, x_2), \ldots, (\lambda_n, x_n)$, and $\{x_1, x_2, \ldots, x_n\}$ forms a basis of $\mathbb{R}^n$.

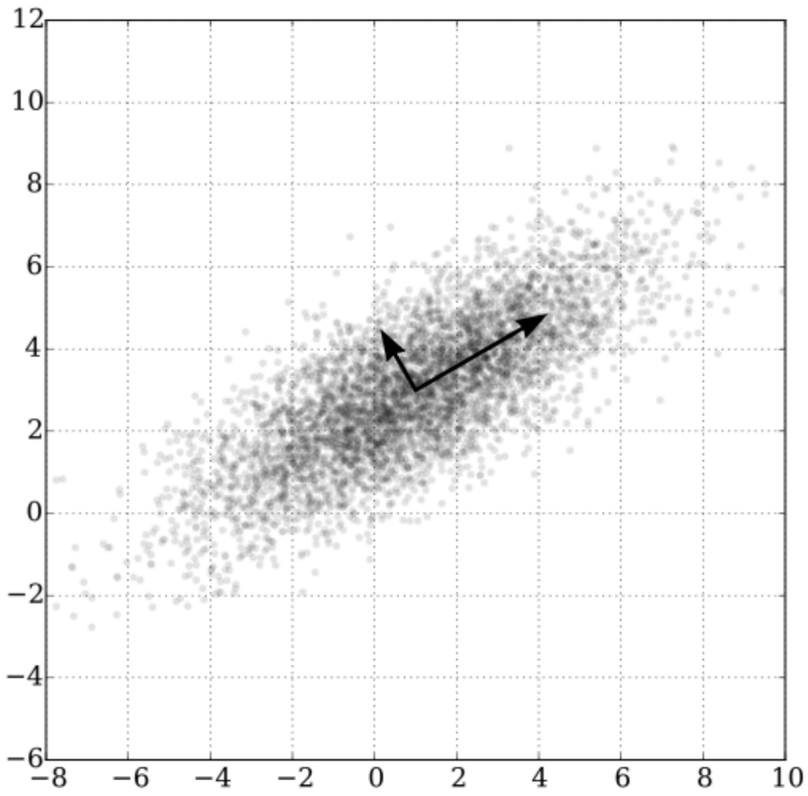
A is called a non-defective or diagonalizable matrix.

$$AX = X\Lambda$$
$$A = X\Lambda X^{-1}$$

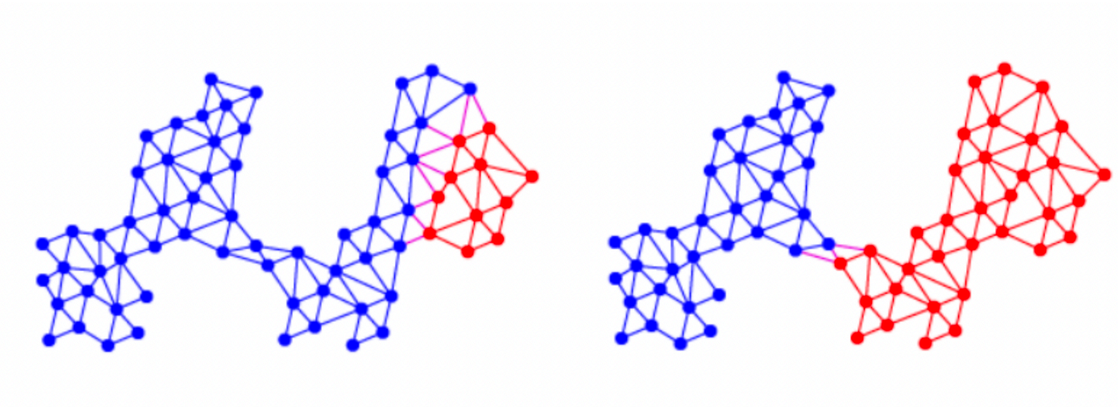
This process is called the Eigen Decomposition

Applications not just subdivision…

Principal Component Analysis (PCA)



Graph Cuts and Spectral Clustering



Further readings:

https://people.eecs.berkeley.edu/~jrs/papers/partnotes.pdf

Subdivision Routine

Let C_0 be a point on the control polyline. We now want to determine its new position C_1 after a new round of subdivision:

1. Determine its neighborhood.
2. Split: Insert new vertices into this neighborhood.
3. Average: Compute the new position based on vertices in this neighborhood.

Note: Not all subdivision methods can be represented with this routine. Can you think of one?

Limit Curve

Q: Given an input curve. What is a limit curve of a subdivision method?

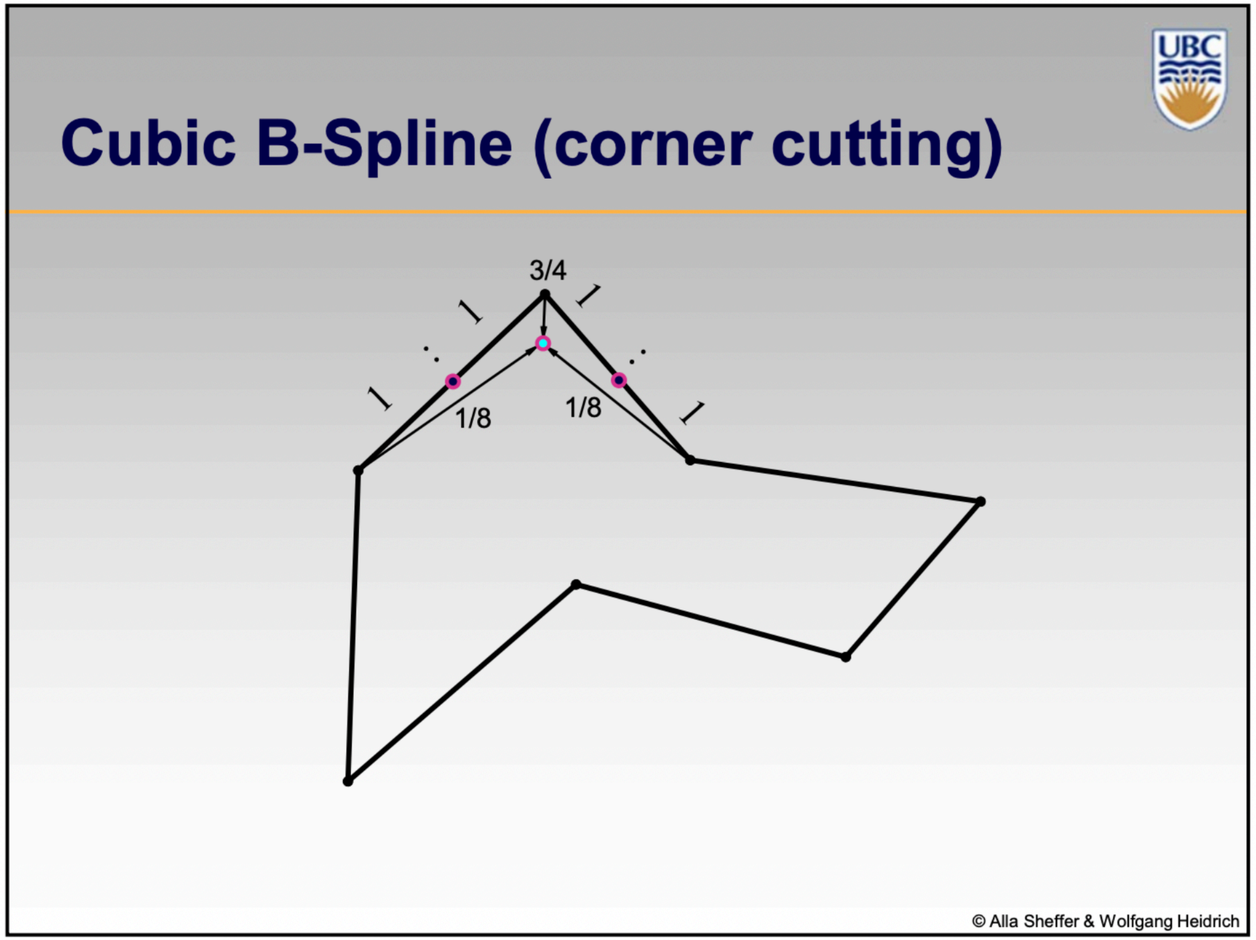
Limit Curve

Q: Given an input curve. What is a limit curve of a subdivision method?

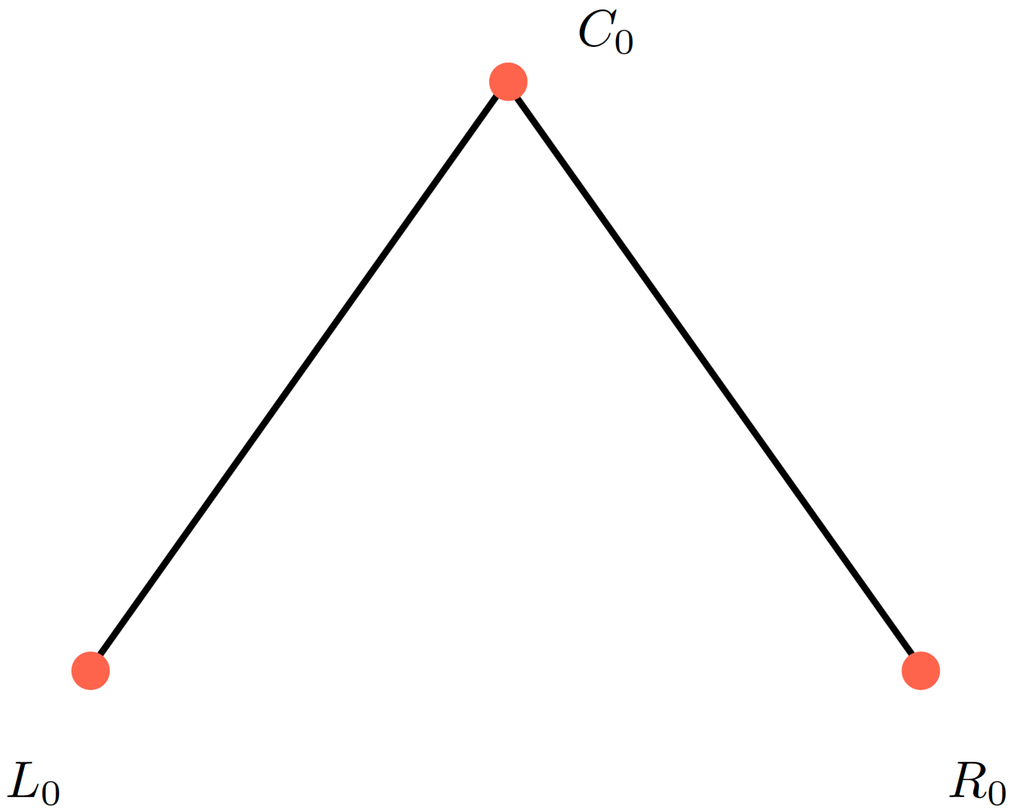
$$P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots$$

$$P_\infty = \lim_{i \rightarrow \infty} P_i$$

Cubic Corner Cutting



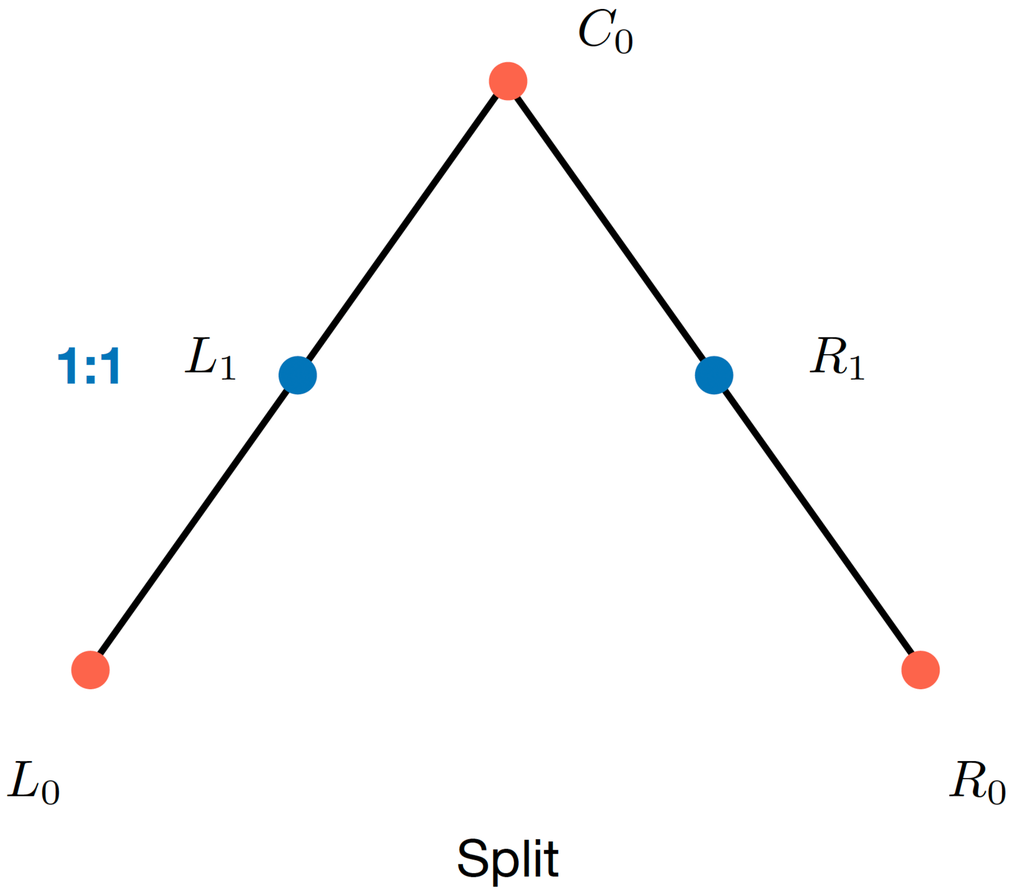
Cubic Corner Cutting



Neighborhood

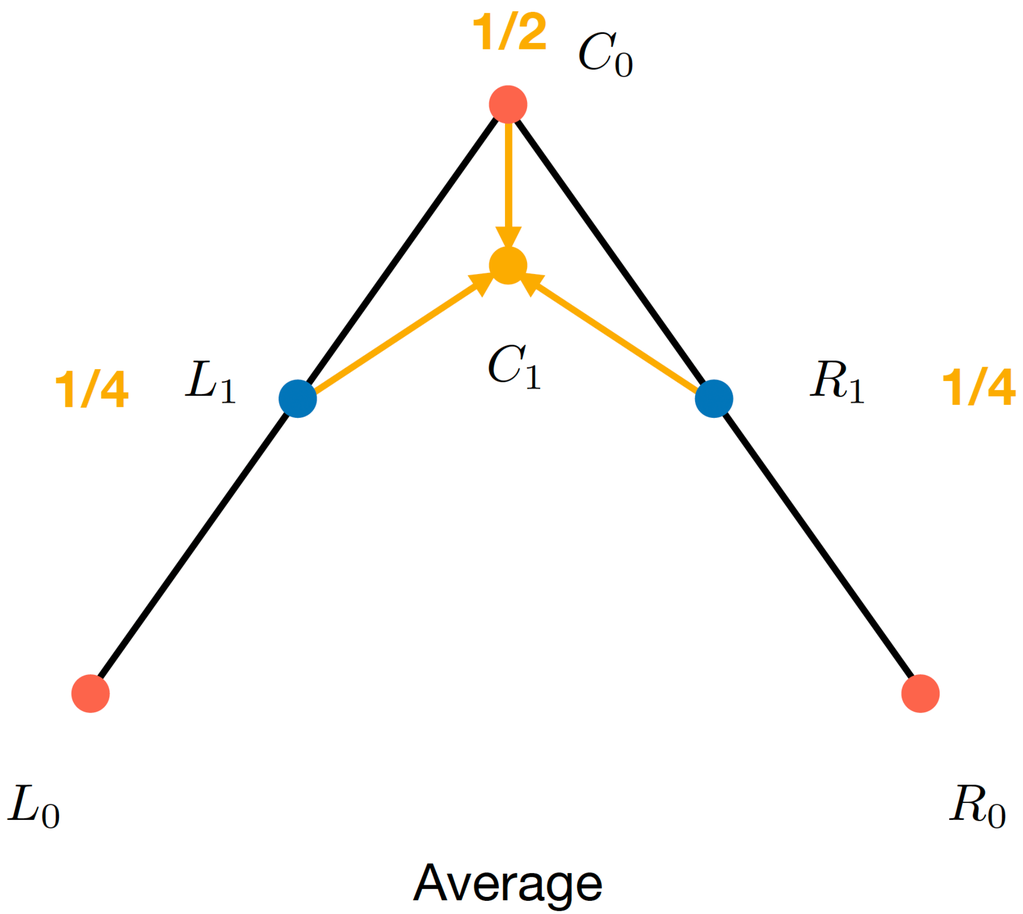
Credits: https://courses.cs.washington.edu/courses/cse557/14au/lectures/markup/subdivision-markp-1pp.pdf

Cubic Corner Cutting



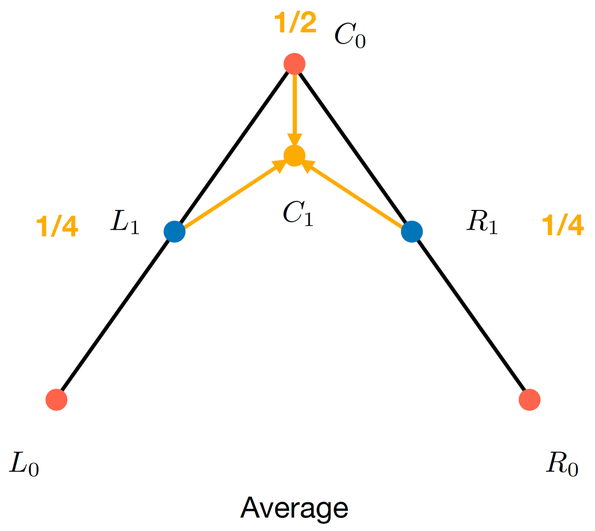
Credits: https://courses.cs.washington.edu/courses/cse557/14au/lectures/markup/subdivision-markp-1pp.pdf

Cubic Corner Cutting



Credits: https://courses.cs.washington.edu/courses/cse557/14au/lectures/markup/subdivision-markp-1pp.pdf

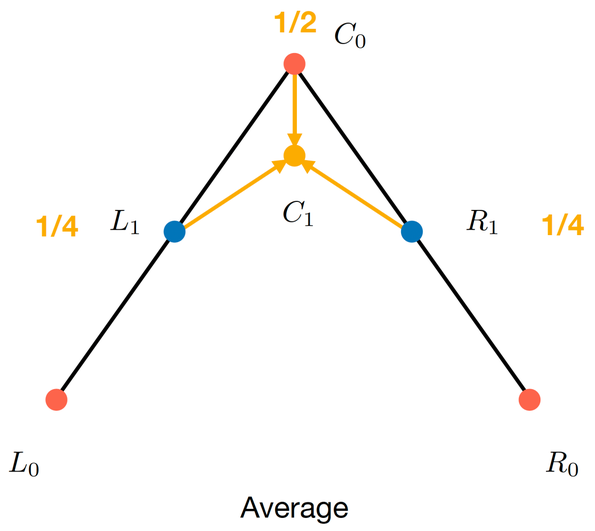
Cubic Corner Cutting



$$\begin{bmatrix} — & L_1^T & — \\ — & C_1^T & — \\ — & R_1^T & — \end{bmatrix} = \begin{bmatrix} & ? & \end{bmatrix} \begin{bmatrix} — & L_0^T & — \\ — & C_0^T & — \\ — & R_0^T & — \end{bmatrix}$$

Credits: https://courses.cs.washington.edu/courses/cse557/14au/lectures/markup/subdivision-markp-1pp.pdf

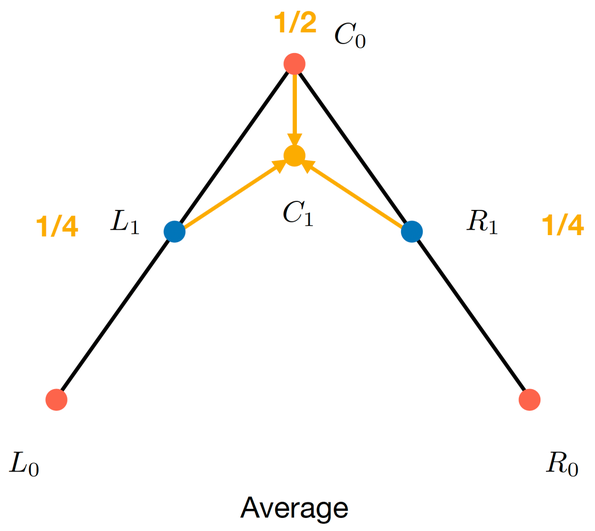
Cubic Corner Cutting



$$\begin{bmatrix} — & L_1^T & — \\ — & C_1^T & — \\ — & R_1^T & — \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} — & L_0^T & — \\ — & C_0^T & — \\ — & R_0^T & — \end{bmatrix}$$

Credits: https://courses.cs.washington.edu/courses/cse557/14au/lectures/markup/subdivision-markp-1pp.pdf

Cubic Corner Cutting



$$\begin{bmatrix} — & L_{i+1}^T & — \\ — & C_{i+1}^T & — \\ — & R_{i+1}^T & — \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} — & L_i^T & — \\ — & C_i^T & — \\ — & R_i^T & — \end{bmatrix}$$

Credits: https://courses.cs.washington.edu/courses/cse557/14au/lectures/markup/subdivision-markp-1pp.pdf

Cubic Corner Cutting: Local Subdivision Matrix

$$\begin{bmatrix} L_{i+1}^x & L_{i+1}^y \\ C_{i+1}^x & C_{i+1}^y \\ R_{i+1}^x & R_{i+1}^y \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} L_i^x & L_i^y \\ C_i^x & C_i^y \\ R_i^x & R_i^y \end{bmatrix}$$

$$P_{i+1} = AP_i$$

Cubic Corner Cutting: Limit Positions

$$P_\infty = A^\infty P_0 = ?$$

Cubic Corner Cutting: Limit Positions

$$P_\infty = A^\infty P_0 = ?$$

$$AX = X\Lambda$$

$$A = X\Lambda X^{-1}$$

Cubic Corner Cutting: Limit Positions

$$P_\infty = A^\infty P_0 = ?$$

$$AX = X\Lambda$$
$$A = X\Lambda X^{-1}$$

$$P_\infty = X\Lambda^\infty X^{-1} P_0$$

Cubic Corner Cutting: Limit Positions

$$P_\infty = A^\infty P_0 = ?$$

$$AX = X\Lambda$$
$$A = X\Lambda X^{-1}$$

$$P_\infty = X\Lambda^\infty X^{-1} P_0$$

Cubic Corner Cutting: Limit Positions

But what will Λ^∞ be?

Recall that subdivision matrices have two properties required for convergence:

$$\lambda_0 = 1,$$

$$|\lambda_i| < 1, \; i = 1,\ldots,n-1$$

Cubic Corner Cutting: Limit Positions

But what will Λ^∞ be?

Recall that subdivision matrices have two properties required for convergence:

$$\lambda_0 = 1,$$

$$|\lambda_i| < 1, \;\; i = 1,\ldots,n-1$$

If $\lambda_0 < 1$, Λ^∞ diminishes to zero
If $\lambda_0 > 1$, Λ^∞ explodes to infinity

Cubic Corner Cutting: Limit Positions

$$P_\infty = A^\infty P_0 = ?$$

$$AX = X\Lambda$$

$$A = X\Lambda X^{-1}$$

$$P_\infty = X \begin{bmatrix} \lambda_1 = 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} X^{-1} P_0$$

Cubic Corner Cutting: Limit Positions

$$A = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

$$(\lambda_1, \mathbf{x_1}) = (1, (1, 1, 1)^T)$$

$$(\lambda_2, \mathbf{x_2}) = (\frac{1}{2}, (-1, 0, 1)^T)$$

$$(\lambda_3, \mathbf{x_3}) = (\frac{1}{4}, (2, -1, 2)^T)$$

Cubic Corner Cutting: Limit Positions

$$A = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

$$(\lambda_1, \mathbf{x_1}) = (1, (1,1,1)^T)$$
$$(\lambda_2, \mathbf{x_2}) = (\frac{1}{2}, (-1,0,1)^T)$$
$$(\lambda_3, \mathbf{x_3}) = (\frac{1}{4}, (2,-1,2)^T)$$

$$X = \begin{bmatrix} 1 & -1 & 2 \\ 1 & 0 & -1 \\ 1 & 1 & 2 \end{bmatrix} \qquad X^{-1} = \begin{bmatrix} 1/6 & 2/3 & 1/6 \\ -1/2 & 0 & 1/2 \\ 1/6 & -1/3 & 1/6 \end{bmatrix}$$

Cubic Corner Cutting: Limit Positions

$$\mathrm{L}_0 = (3,1)^T, \quad \mathrm{C}_0 = (4,3)^T, \quad \mathrm{R}_0 = (5,1)^T$$

$$\mathrm{C}_\infty = ?$$

$$P_\infty = X\Lambda^\infty X^{-1} P_0$$

$$A = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & -1 & 2 \\ 1 & 0 & -1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$(\lambda_1, \mathbf{x_1}) = (1, (1,1,1)^T)$$

$$(\lambda_2, \mathbf{x_2}) = (\frac{1}{2}, (-1,0,1)^T)$$

$$(\lambda_3, \mathbf{x_3}) = (\frac{1}{4}, (2,-1,2)^T)$$

$$X^{-1} = \begin{bmatrix} 1/6 & 2/3 & 1/6 \\ -1/2 & 0 & 1/2 \\ 1/6 & -1/3 & 1/6 \end{bmatrix}$$

Cubic Corner Cutting: Limit Positions

$$\mathrm{L}_0 = (3,1)^T, \quad \mathrm{C}_0 = (4,3)^T, \quad \mathrm{R}_0 = (5,1)^T$$

$$\mathrm{C}_\infty = ?$$

$$P_\infty = X\Lambda^\infty X^{-1} P_0$$

$$P_\infty = \begin{bmatrix} 1 & -1 & 2 \\ 1 & 0 & -1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/6 & 2/3 & 1/6 \\ -1/2 & 0 & 1/2 \\ 1/6 & -1/3 & 1/6 \end{bmatrix} P_0$$

Cubic Corner Cutting: Limit Positions

$$\mathrm{L}_0 = (3,1)^T, \quad \mathrm{C}_0 = (4,3)^T, \quad \mathrm{R}_0 = (5,1)^T$$

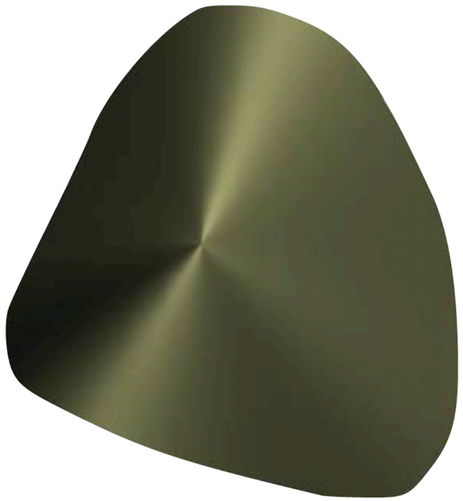
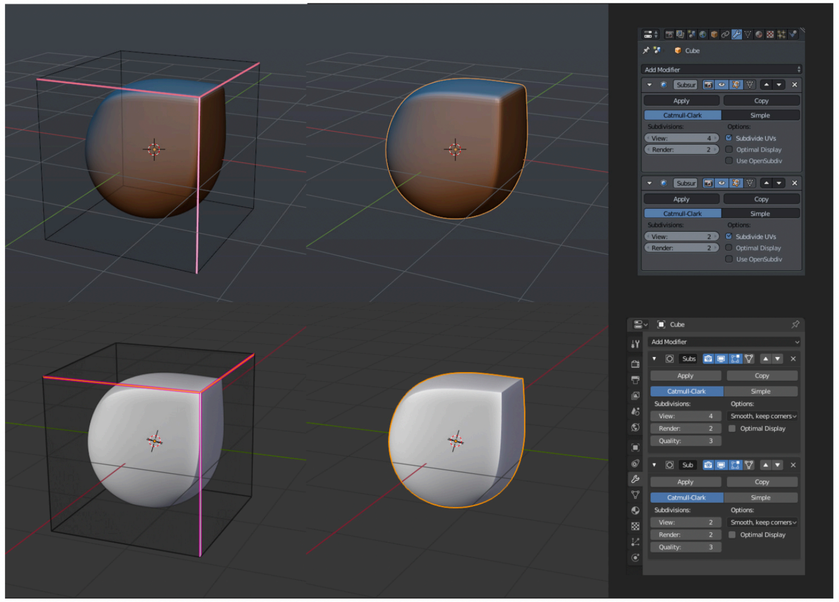
$$\mathrm{C}_\infty = ?$$

$$P_\infty = \begin{bmatrix} 1 & -1 & 2 \\ 1 & 0 & -1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/6 & 2/3 & 1/6 \\ -1/2 & 0 & 1/2 \\ 1/6 & -1/3 & 1/6 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 4 & 3 \\ 5 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 7/3 \\ 4 & 7/3 \\ 4 & 7/3 \end{bmatrix}$$

$$C_\infty = (4,7/3)$$

Subdivision: Pitfalls in Practice

Subdivision is a mature topic with many reliable implementations:
Any modeling software (Blender, 3ds Max, AutoCAD, etc);
Open-source libraries (e.g. OpenSubdiv from Pixar).

It's unlikely that one will implement subdivision from scratch in practice but it's important to know the tricky/special bits when using existing subdivision implementation:

1. Creases.
2. Boundaries.
3. Extraordinary Points.