Jagged Array

Jagged Array: Irregular 2D array

- Rows of the array has variable sizes

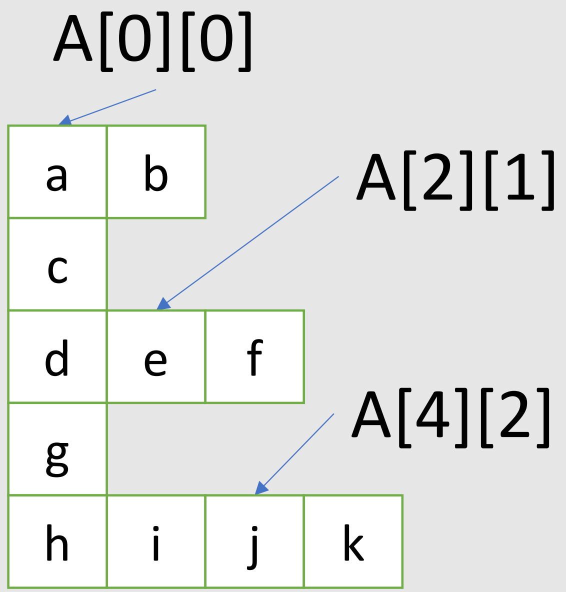
A = [[a,b],[c],[d,e,f],[g],[h,i,j,k]]

At the low-level, computer can only handle 1D array

Jagged Array: Irregular 2D array

- Rows of the array has variable sizes

A = [[a,b],[c],[d,e,f],[g],[h,i,j,k]]



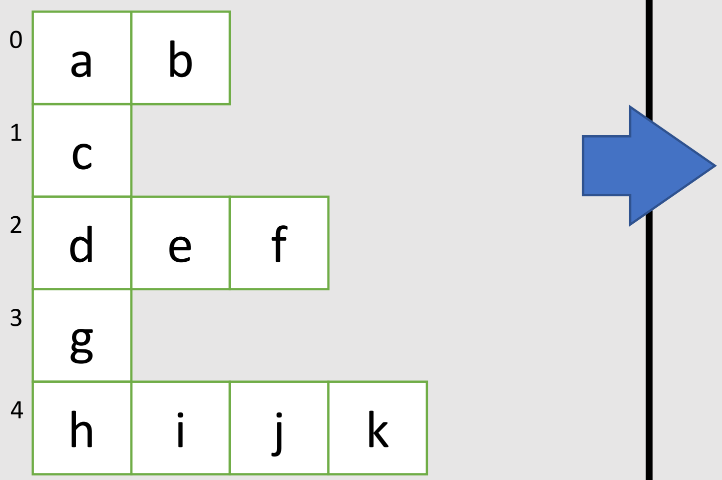
Array of array is inefficient !

```
std::vector< std::vector<int> > arrayOfArray;
```

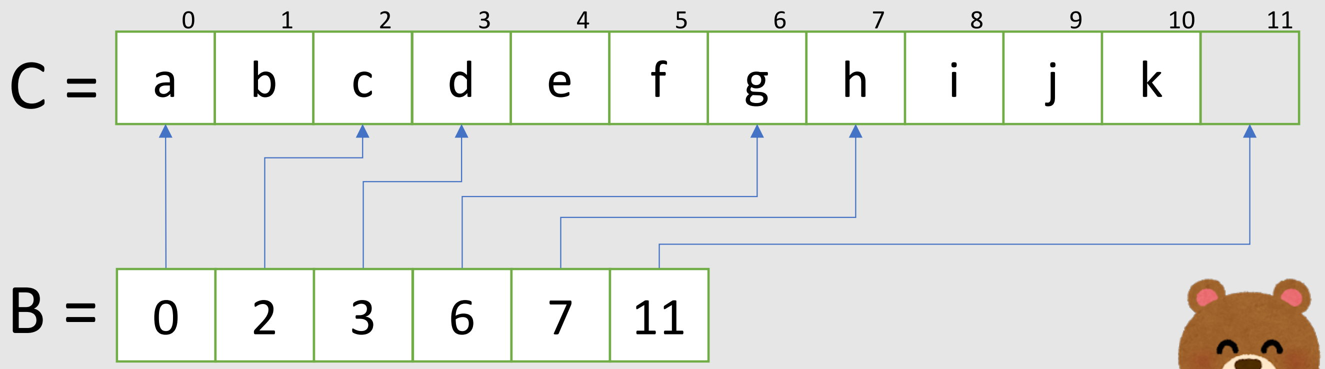
Jagged Array: Irregular 2D array

- A jagged array can be expressed by two 1D arrays

A = [[5,7],[1],[9,3,4],[3],[5,5,4,3]]

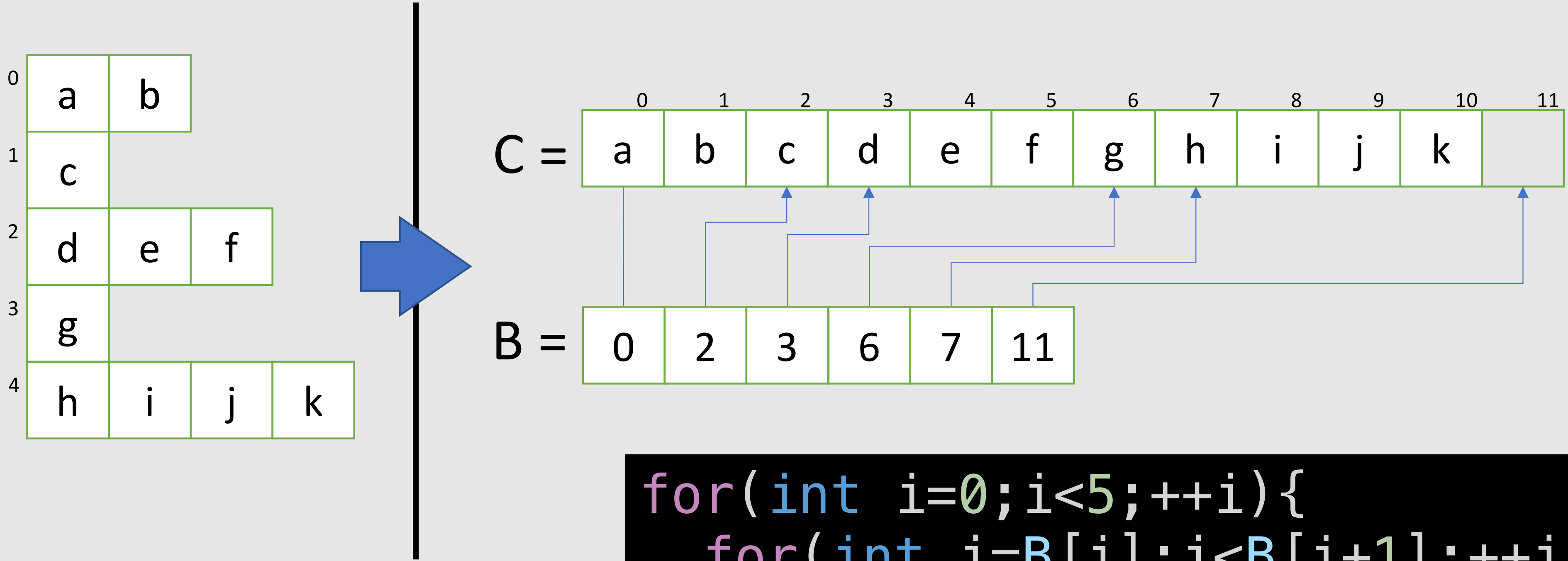


$$A[i][j] = C[B[i]+j]$$



Array of stating indexes for each row

Loop Over Jagged Array




```
for(int i=0;i<5;++i){
  for(int j=B[i];j<B[i+1];++j){
    float v = C[j];
  }
}
```

Sparse Linear System

Gradient Operator **Distributes** over Addition

$$W = W_1 + W_2 + \cdots = \sum W_i$$

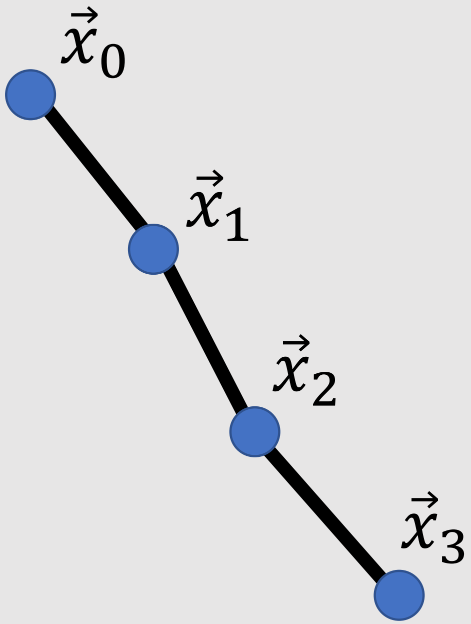
gradient $$\nabla W = \nabla W_1 + \nabla W_2 + \cdots = \sum \nabla W_i$$

hessian $$\nabla^2 W = \nabla^2 W_1 + \nabla^2 W_2 + \cdots = \sum \nabla^2 W_i$$

Sparsity of a Hessian Matrix: Polyline

$$\nabla^2 W(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3) = \nabla^2 W_1(\vec{x}_0, \vec{x}_1) + \nabla^2 W_2(\vec{x}_1, \vec{x}_2) + \nabla^2 W_3(\vec{x}_2, \vec{x}_3)$$

$$\begin{bmatrix} * & * & 0 & 0 \\ * & * & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & * & * & 0 \\ 0 & * & * & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{bmatrix}$$

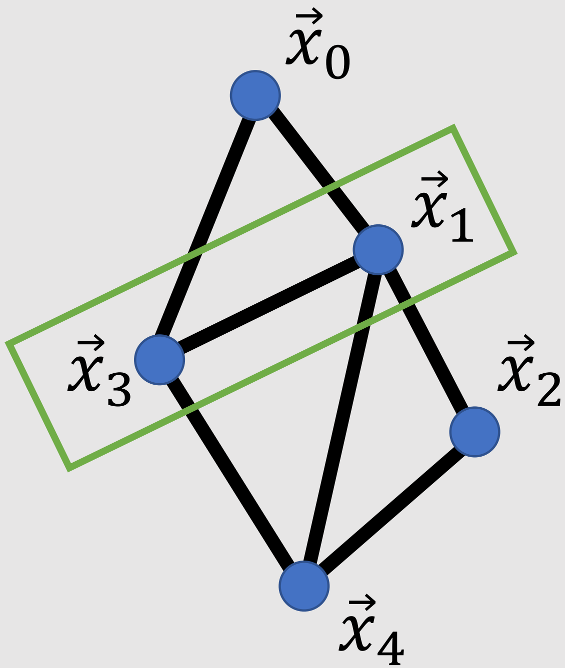


$$\begin{bmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix}$$

band matrix

Sparsity of a Hessian Matrix: Edges

$$\nabla^2 W(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4) = \nabla^2 W'(\vec{x}_0, \vec{x}_1) + \nabla^2 W'(\vec{x}_0, \vec{x}_3) + \nabla^2 W'(\vec{x}_1, \vec{x}_2)$$
$$+ \nabla^2 W'(\vec{x}_1, \vec{x}_3) + \nabla^2 W'(\vec{x}_1, \vec{x}_4) + \nabla^2 W'(\vec{x}_2, \vec{x}_4) + \nabla^2 W'(\vec{x}_3, \vec{x}_4)$$

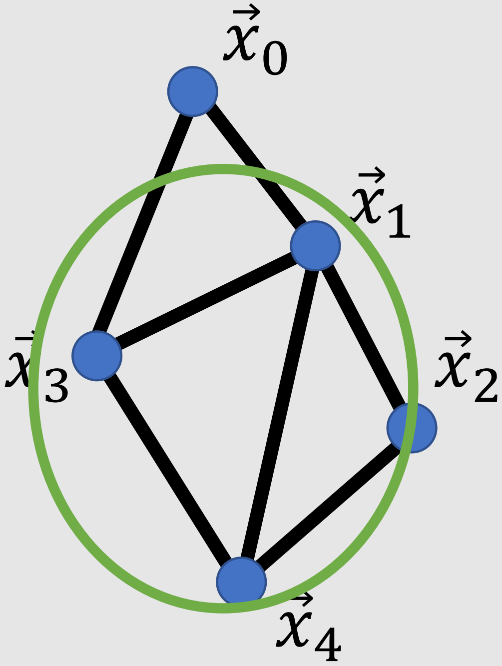


$$\begin{bmatrix} * & * & 0 & * & 0 \\ * & * & * & * & * \\ 0 & * & * & 0 & * \\ * & * & 0 & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

For each row, the non-zero pattern is associated with one-ring neighborhood

Sparsity of a Hessian Matrix: Triangles

$$\nabla^2 W(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4) = \nabla^2 W'(\vec{x}_0, \vec{x}_3, \vec{x}_1) + \boxed{\nabla^2 W'(\vec{x}_1, \vec{x}_3, \vec{x}_4)} + \nabla^2 W'(\vec{x}_1, \vec{x}_4, \vec{x}_2)$$

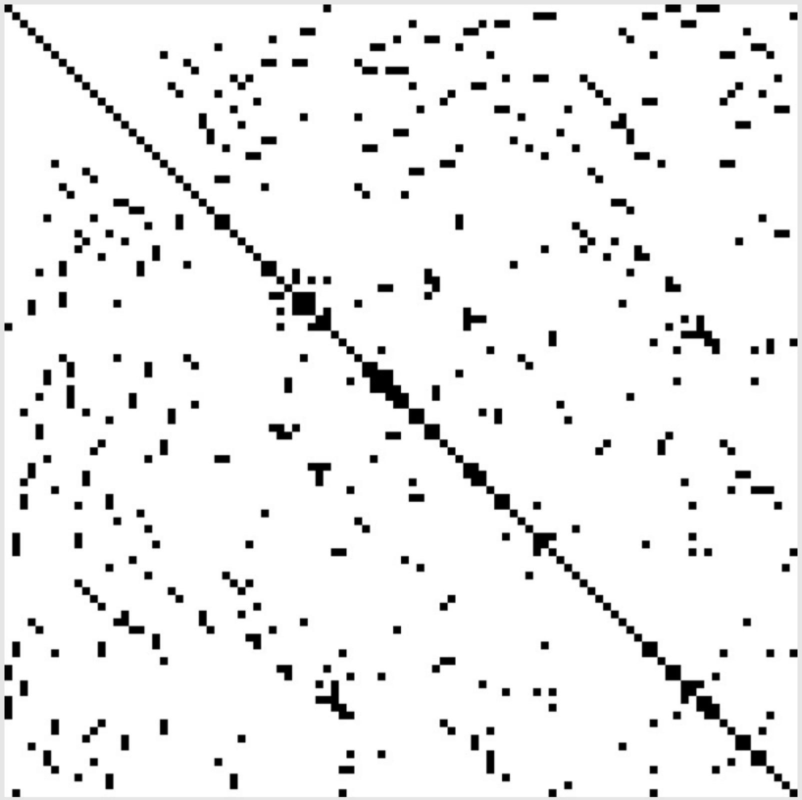


$$\begin{bmatrix} * & * & 0 & * & 0 \\ * & \boxed{*} & * & \boxed{*} & \boxed{*} \\ 0 & * & * & 0 & * \\ * & \boxed{*} & 0 & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & * & \boxed{*} & \boxed{*} \end{bmatrix}$$

For each row, the non-zero pattern is associated with one-ring neighborhood

Sparse Matrix is Common in Optimization

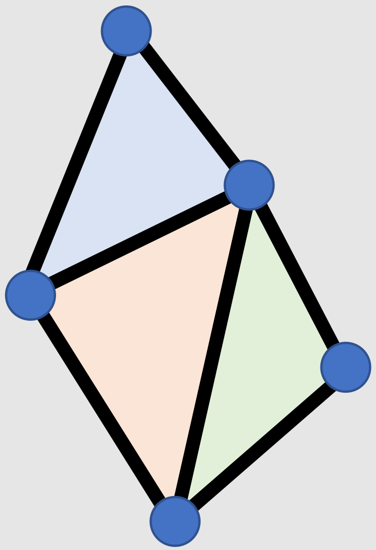
- FEM,FDM produce a very sparse matrix (e.g., 30 entries par row)

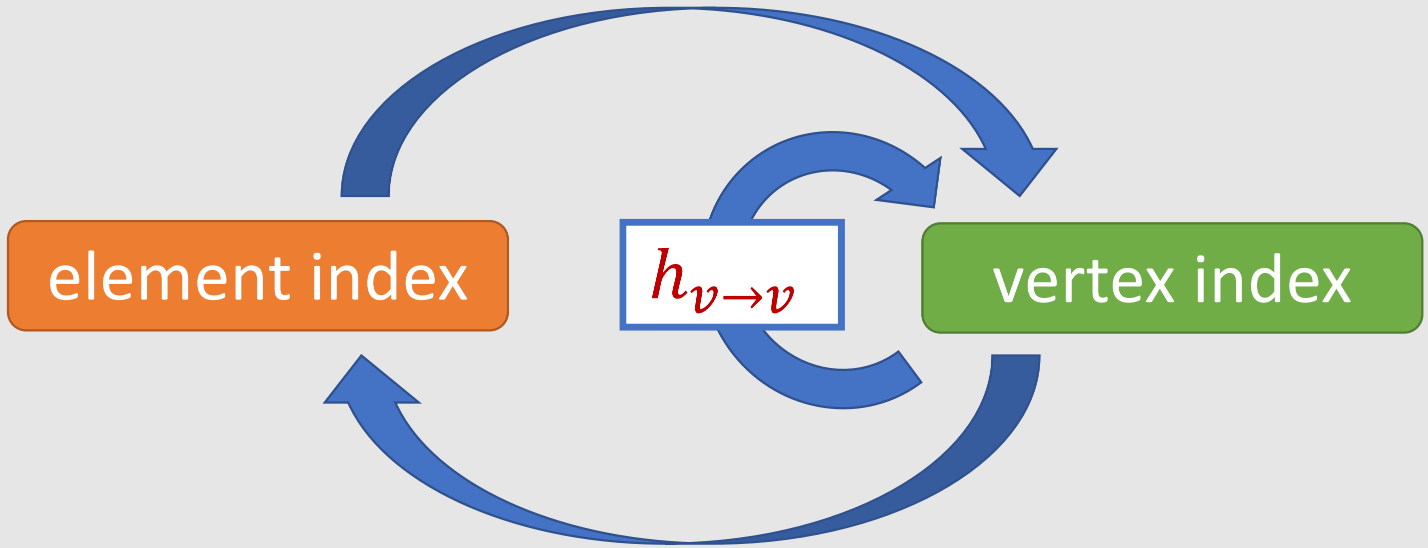


(Wikipedia: Sparse Matrix)

Constructing One-Ring Neighborhood Graph

element to vertex array: $f_{e \to v}$
(regular 2D array)





1. vertex to element array: $g_{v \to e}$
   (jagged array, inverse of $f_{e \to v}$)

2. one-ring neighborhood:
   $h_{v \to v} : f_{v \to e}(g_{v \to e}(v))$
   (jagged array)

Coordinate (COO) Data Structure

- A.k.a triplet format, 3-tuple format
- Interface for matrix libraries (e.g., Pytorch, Eigen)

$$A=\begin{bmatrix} a & 0 & 0 & b & 0 \\ 0 & c & 0 & 0 & d \\ 0 & 0 & e & 0 & 0 \\ f & 0 & 0 & g & 0 \\ 0 & h & 0 & 0 & i \end{bmatrix}$$

Row	Column	Value
0	0	a
0	3	b
1	1	c
1	4	d
…	…	…

Compressed Sparse Row (CSR) Data Structure

- Data structure for efficient matrix-vector product using jagged array

$$A=\begin{bmatrix} a & 0 & 0 & b & 0 \\ 0 & c & 0 & 0 & d \\ 0 & 0 & e & 0 & 0 \\ f & 0 & 0 & g & 0 \\ 0 & h & 0 & 0 & i \end{bmatrix}$$

$$A[i][C[B[i]+k]] = D[B[i]+k]$$

column Index: C =

0	1	2	3	4	5	6	7	8	9
0	3	1	4	2	0	3	1	4	

value: D =

a	b	c	d	e	f	g	h	i	

indexes of C&D: B =

0	2	4	5	7	9

Laplacian Mesh Deformation

Laplacian Surface Editing[Olga et al. 2004]

"Laplacian Surface Editing" paper published at Symposium on Geometry Processing 2004 by Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Roessl and Hans-Peter Seidel.
https://igl.ethz.ch/projects/Laplacian-mesh-processing/Laplacian-mesh-editing/

It’s One of the Most Important Tools in CG

- So many papers use Laplacian Deformation to solve their problems

Olga Sorkine-Hornung

FOLLOWING

Professor of Computer Science, ETH Zurich
Verified email at inf.ethz.ch - Homepage

Computer Graphics Geometry Processing Digital Fabrication Image and Video Processing

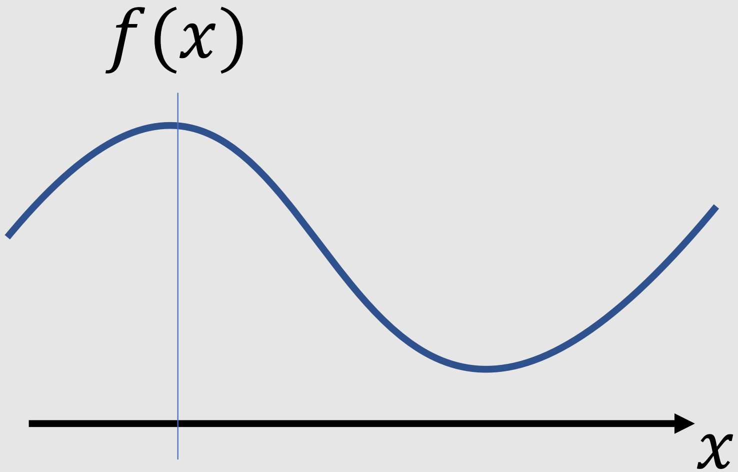
TITLE	CITED BY	YEAR
Laplacian surface editing O Sorkine, D Cohen-Or, Y Lipman, M Alexa, C Rössl, HP Seidel Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry …	1361	2004
As-rigid-as-possible surface modeling O Sorkine, M Alexa Symposium on Geometry processing 4, 109-116	1036	2007
Optimized scale-and-stretch for image resizing YS Wang, CL Tai, O Sorkine, TY Lee ACM SIGGRAPH Asia 2008 papers, 1-8	811	2008
On linear variational surface deformation methods M Botsch, O Sorkine IEEE transactions on visualization and computer graphics 14 (1), 213-230	688	2007

What is Laplacian?

Continuous Setting

Second derivative

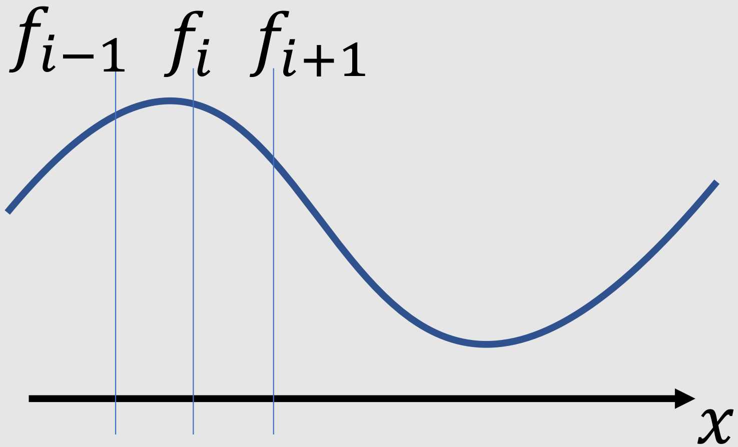
$$\Delta f = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right)$$



Discrete Setting

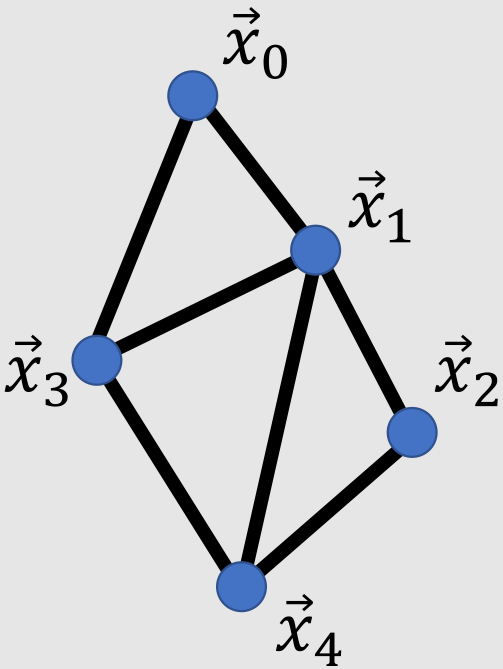
Laplacian is the difference from the average of neighbors

$$\Delta f \simeq f_i - \frac{f_{i+1} + f_{i-1}}{2}$$



Adjacency Matrix

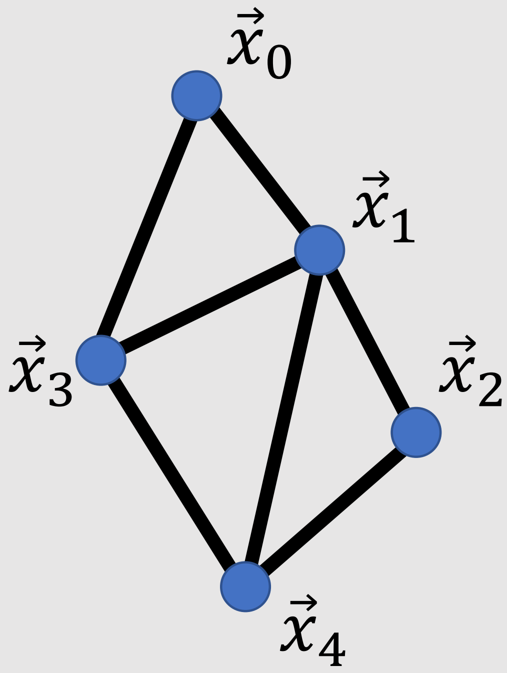
- Connected edges takes 1 in the matrix
- Eigenvalue of Adjacency Matrix -> search engine (PageRank)



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Graph Laplacian Matrix

- All the connected edges takes -1 and diagonal takes valence

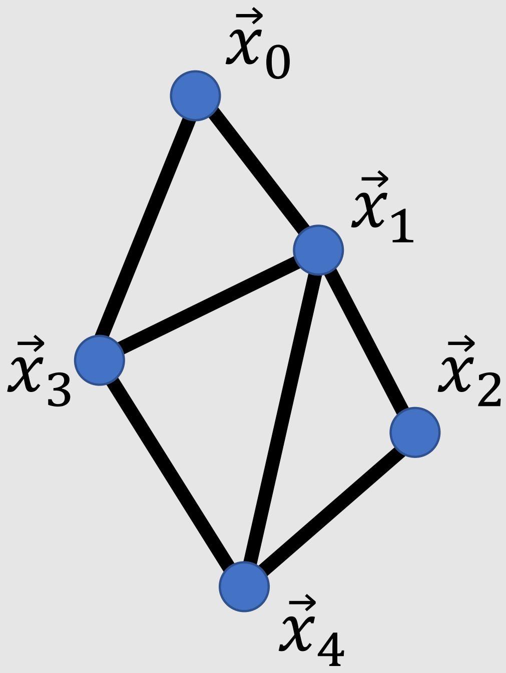


$$L = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & 0 & -1 \\ -1 & -1 & 0 & 3 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}$$

valence: # of connected points

Graph Laplacian Matrix as Constraints

- $L\vec{x} = 0$ means all the vertices are average of connected ones

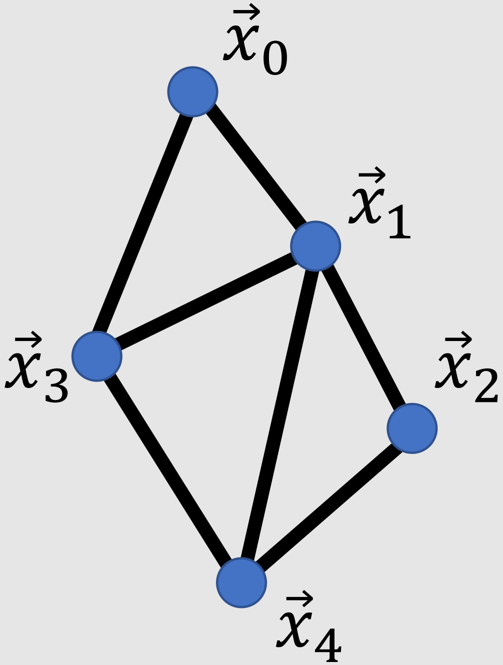


$$L\vec{x} = 0$$

$$\Longrightarrow \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & 0 & -1 \\ -1 & -1 & 0 & 3 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = 0$$

Graph Laplacian Matrix as Optimization

- $L\vec{x} = 0$ means sum of square difference is minimized



$$W = \frac{1}{2}\sum_{e \subset \mathcal{E}} \left\| x_{e_1} - x_{e_2} \right\|^2$$

$$= \frac{1}{2}\vec{x}^T L \vec{x}$$

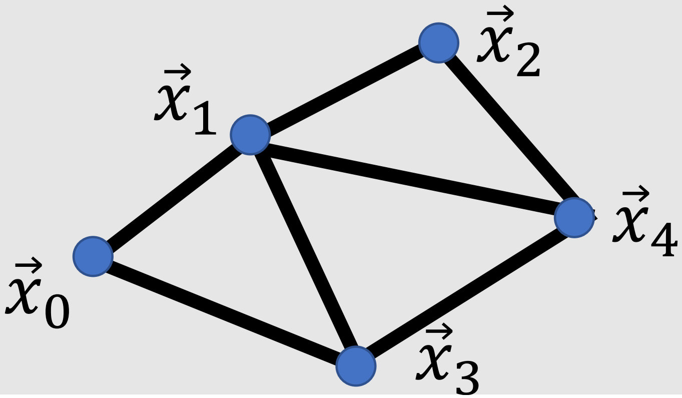
W is minimized $\rightarrow \frac{\partial W}{\partial \vec{x}} = L\vec{x} = 0$

Laplacian in Continuous Setting

Discrete Setting

- Difference on edge

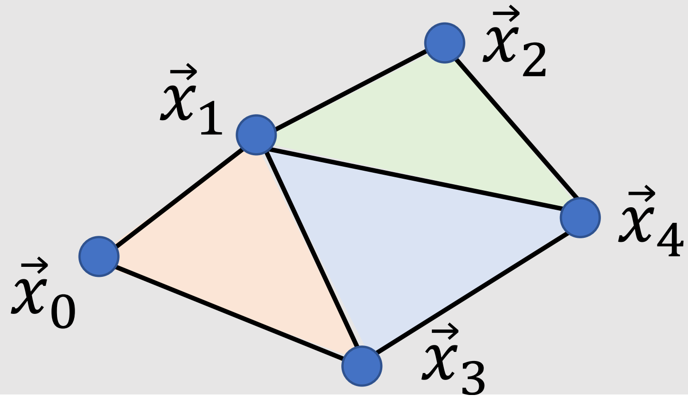
$$W = \frac{1}{2}\sum_{e\subset\mathcal{E}}\left\|x_{e_1} - x_{e_2}\right\|^2$$
$$= \frac{1}{2}\vec{x}^T L\vec{x}$$



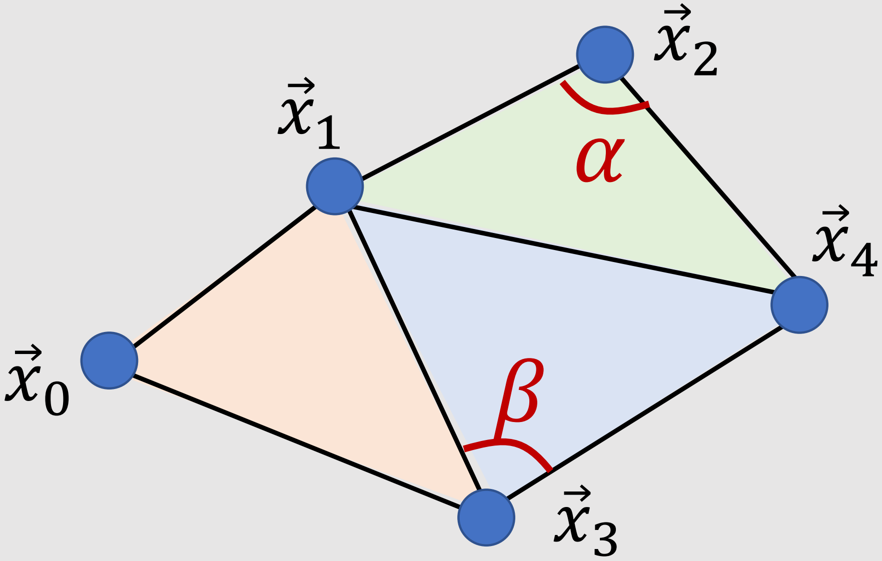
Continuous Setting

- Magnitude of gradient inside triangle

$$W = \frac{1}{2}\sum_{t\subset\mathcal{T}}\int_{\Omega_t}\|\nabla x\|^2\, d\Omega$$
$$= \frac{1}{2}\vec{x}^T \bar{L}\vec{x}$$



Cotangent Laplacian



Continuous Setting

- Magnitude of gradient inside triangle

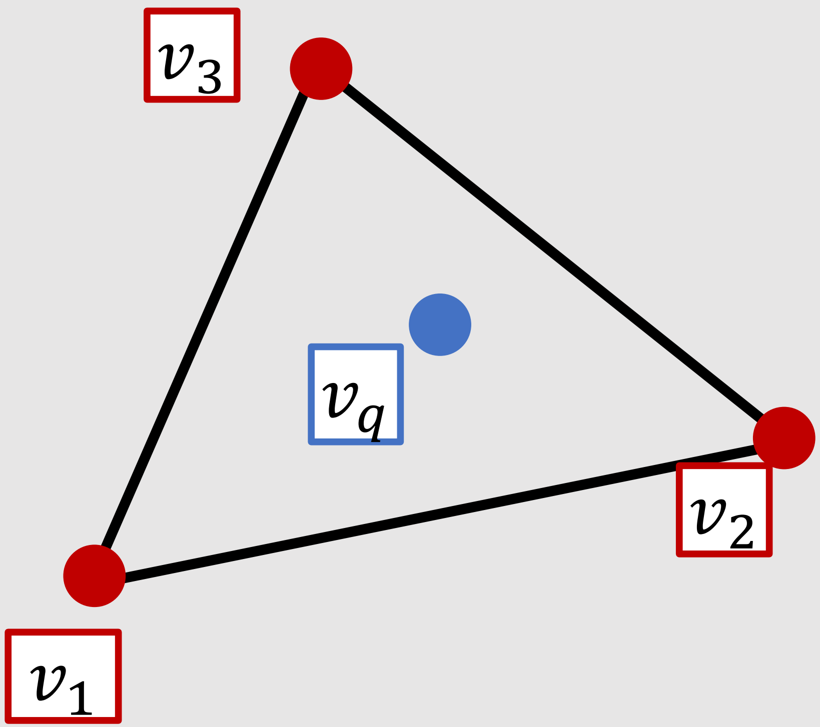
$$W = \frac{1}{2} \sum_{t \subset \mathcal{T}} \int_{\Omega_t} \|\nabla x\|^2 \, d\Omega$$
$$= \frac{1}{2} \vec{x}^T \bar{L} \vec{x}$$

$$\begin{bmatrix} l_{00} & l_{01} & 0 & l_{03} & 0 \\ l_{10} & l_{11} & l_{12} & l_{13} & l_{14} \\ 0 & l_{21} & l_{22} & 0 & l_{24} \\ l_{30} & l_{31} & l_{32} & l_{33} & l_{34} \\ 0 & l_{41} & l_{42} & l_{43} & 0 \end{bmatrix}$$

$$l_{14} = -\frac{1}{2}(\cot \alpha + \cot \beta)$$

$$l_{11} = -(l_{10} + l_{12} + l_{13} + l_{14})$$

Interpolation Using Barycentric Coordinate



Position from coordinate :

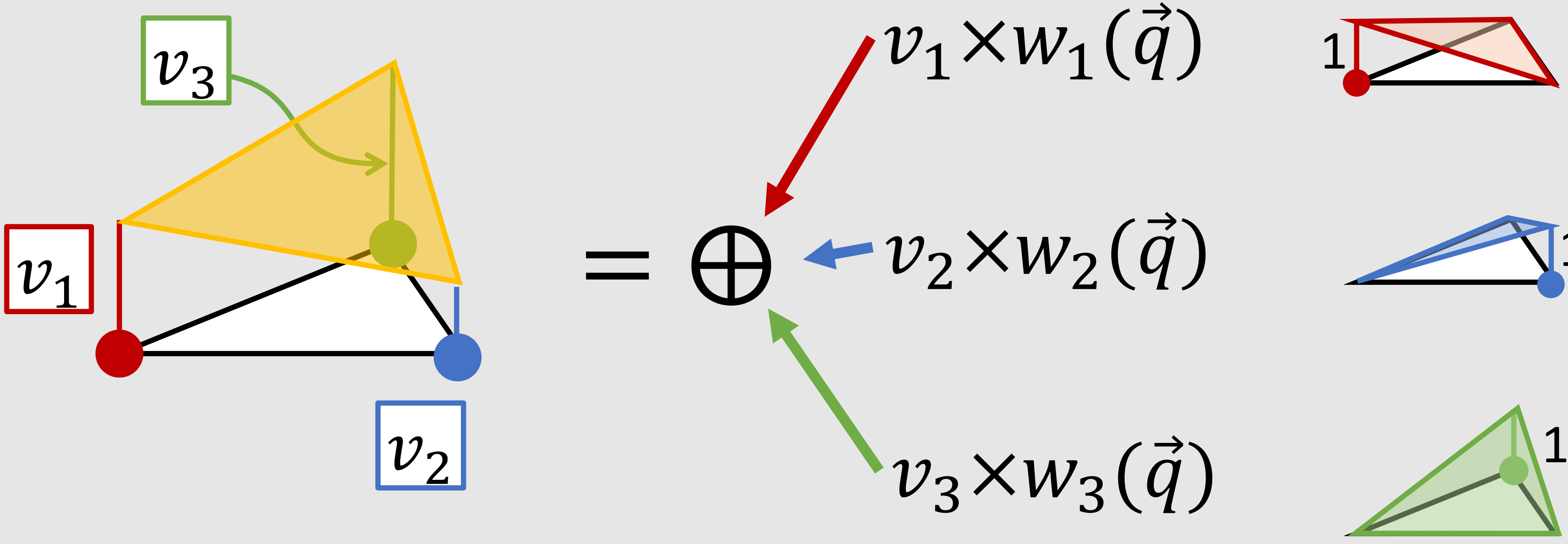
$$\vec{q} = w_1 \vec{p}_1 + w_2 \vec{p}_2 + w_3 \vec{p}_3$$

Value from coordinate:

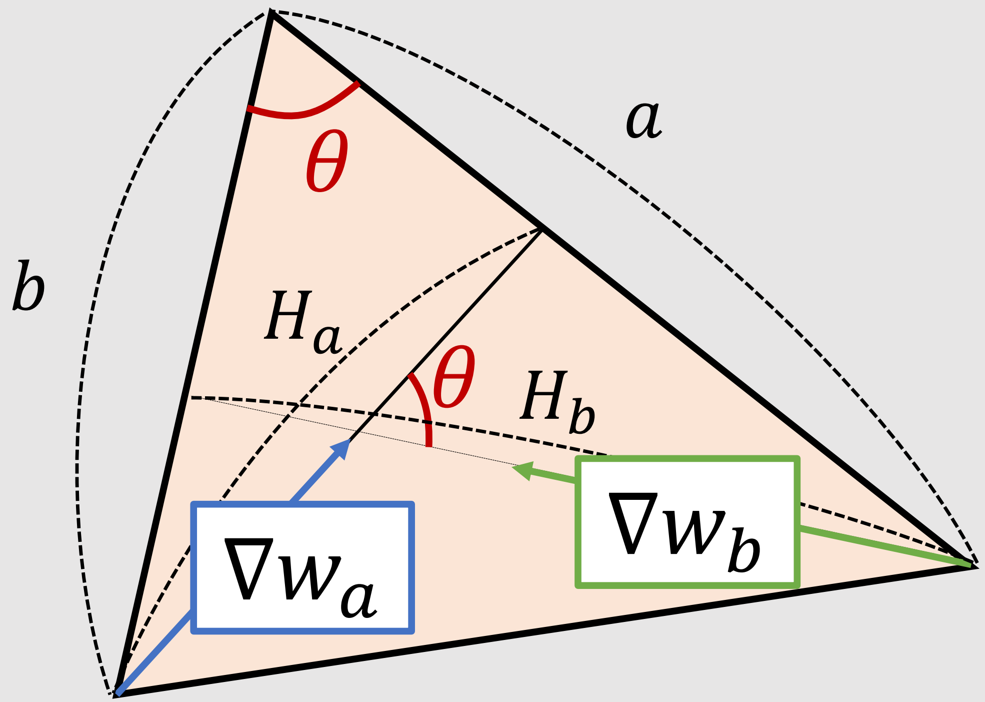
$$v_q = w_1 v_1 + w_2 v_2 + w_3 v_3$$

Interpolation Using Barycentric Coordinate

- Interpolated value is linear w.r.t coordinate



Why Cotangent?



$$Area = \frac{1}{2} ab \sin\theta$$

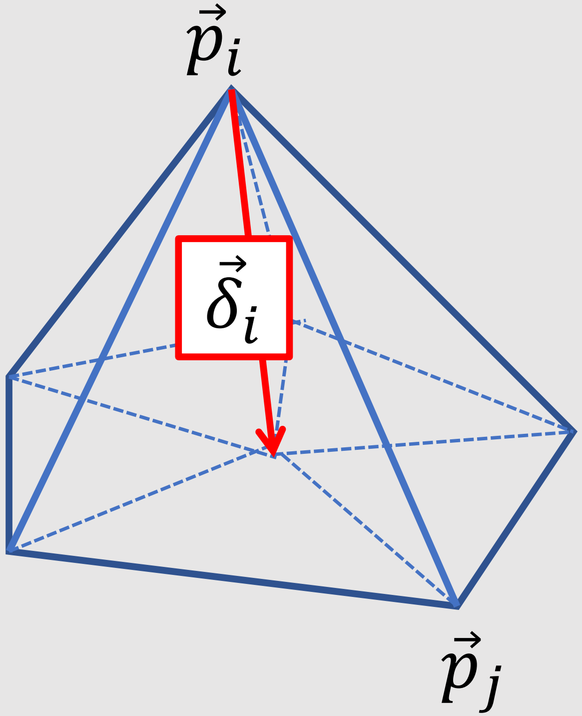
$$\|\nabla w_a\| = H_a = \frac{a}{2Area}$$

$$\|\nabla w_b\| = H_b = \frac{b}{2Area}$$

$$l_{ij} = \int_{\Omega_t} \nabla w_a \cdot \nabla w_b \, d\Omega = Area \|\nabla w_a\| \|\nabla w_b\| \cos\theta = \frac{\cos\theta}{2\sin\theta} = \frac{\cot\theta}{2}$$

Laplacian on Mesh: Differential Coordinate

- Differential Coordinate Δ encode position of vertices



Differential Coordinate

$$\vec{\delta}_i = \vec{p}_i - \sum_{j \in N_i} l_{ij}\vec{p}_j = \sum_{k \in V} l_{ik}\vec{p}_k$$

$$\Delta = LP$$

$$\Delta = \begin{bmatrix} \delta_{1x} & \delta_{1y} & \delta_{1z} \\ \delta_{2x} & \delta_{2y} & \delta_{2z} \\ \delta_{3x} & \delta_{3y} & \delta_{3z} \\ & \vdots & \end{bmatrix}$$

$$P = \begin{bmatrix} p_{1x} & p_{1y} & p_{1z} \\ p_{2x} & p_{2y} & p_{2z} \\ p_{3x} & p_{3y} & p_{3z} \\ & \vdots & \end{bmatrix}$$

What is Coordinate?

- Coordinate is a parameterization of positions

$$\Delta = LP$$

positions

parameters

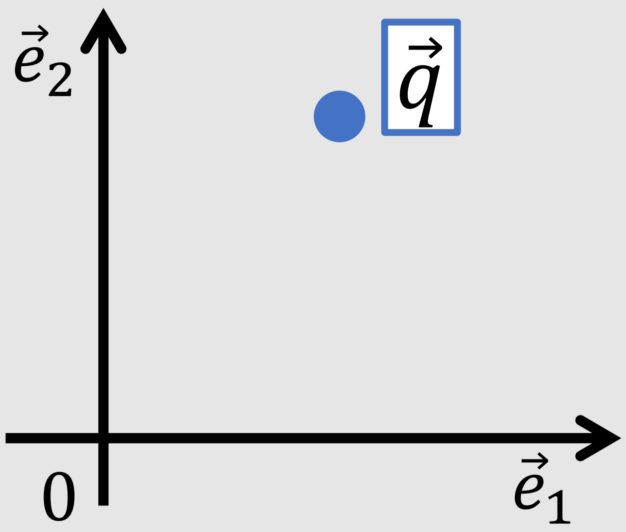
P

Δ

$$P = L^{-1}\Delta$$

Cartesian Coordinate (デカルト座標系)

- Origin and orthonomal basis vectors



Given coordinate (w_1, w_2) :

$$\vec{q} = w_1\, \vec{e}_1 + w_2 \vec{e}_2$$

Given position $\vec{q}$

$$w_1 = \vec{q} \cdot \vec{e}_1, \qquad w_2 = \vec{q} \cdot \vec{e}_2$$

René Descartes

Laplacian Mesh Deformation

- Smooth deformation by minimizing diff. of differential coord.

$$W(P) = \frac{1}{2}\left\| LP - LP_{ref} \right\|_F^2 + K(P)$$

some constraints

$$P_{def} = \operatorname*{argmin}_P W(P) \Rightarrow \left.\frac{\partial W(P)}{\partial P}\right|_{P_{def}} = 0$$