

Matrix Data Structure

Commutative, Associative & Distributive Laws

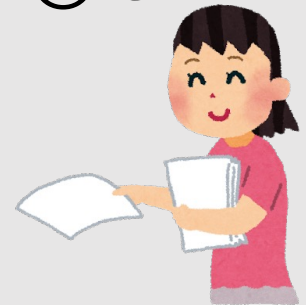
Associative law

$$A \odot (B \odot C) = (A \odot B) \odot C$$



Distributive law

$$A \odot (B + C) = A \odot B + A \odot C$$



Commutative law

$$A \odot B = B \odot A$$



Matrices don't (always) commute

Useful Property of Associative Law



Associative law for matrix: $A(BC) = (AB)C$



$$E \left(D \left(C \left(B \left(Ax \right) \right) \right) \right) = \underbrace{(EDCBA)}_K x$$

Precompute $K = EDCBA$ to efficiently compute Kx for various x

Quiz: What is an Operator without Associative Law?

$$A \odot (B \odot C) \neq (A \odot B) \odot C$$



Gradient Operator **Distributes** over Addition

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

gradient

$$\nabla W = \nabla W_1 + \nabla W_2 + \dots = \sum \nabla W_i$$

hessian

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

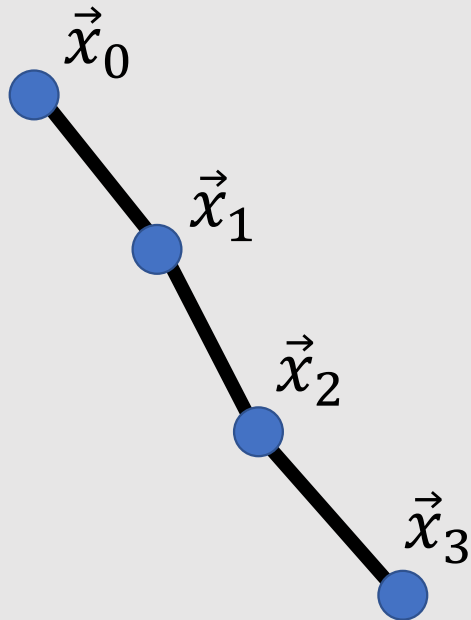
Sparsity of a Hessian Matrix: Polyline

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

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

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

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

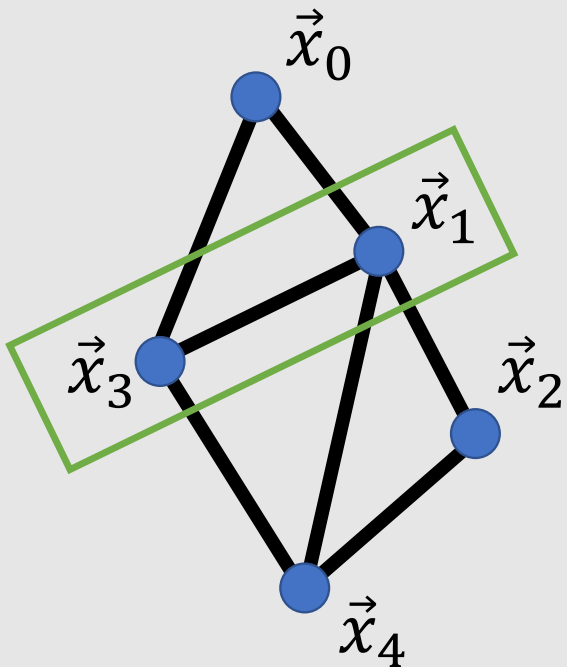


$$\begin{bmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix}$$

band matrix

Sparsity of a Hessian Matrix: Edges

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

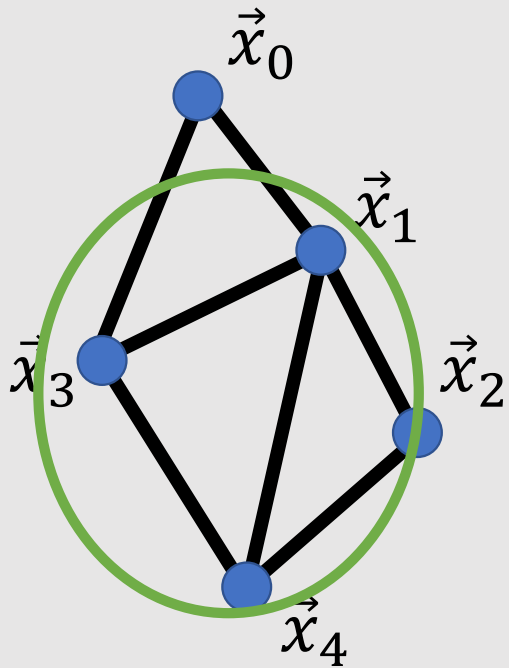


$$\begin{bmatrix} * & * & 0 & * & 0 \\ * & * & * & * & * \\ 0 & * & * & 0 & * \\ * & * & 0 & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

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

Sparsity of a Hessian Matrix: Triangles

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

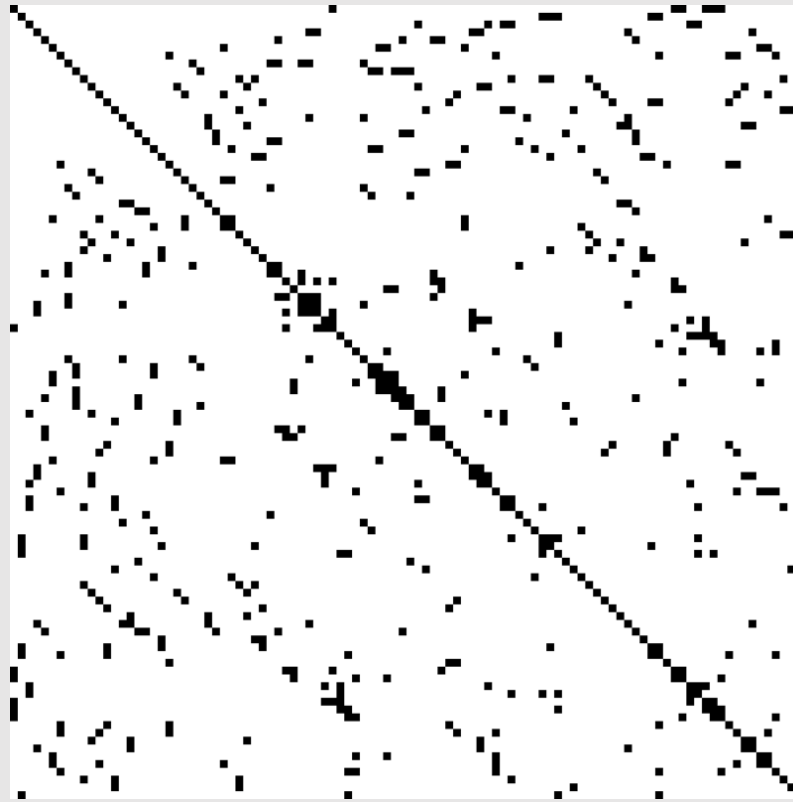


$$\begin{bmatrix} * & * & 0 & * & 0 \\ * & * & * & * & * \\ 0 & * & * & 0 & * \\ * & * & 0 & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

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

Sparse Matrix is Common in Optimization

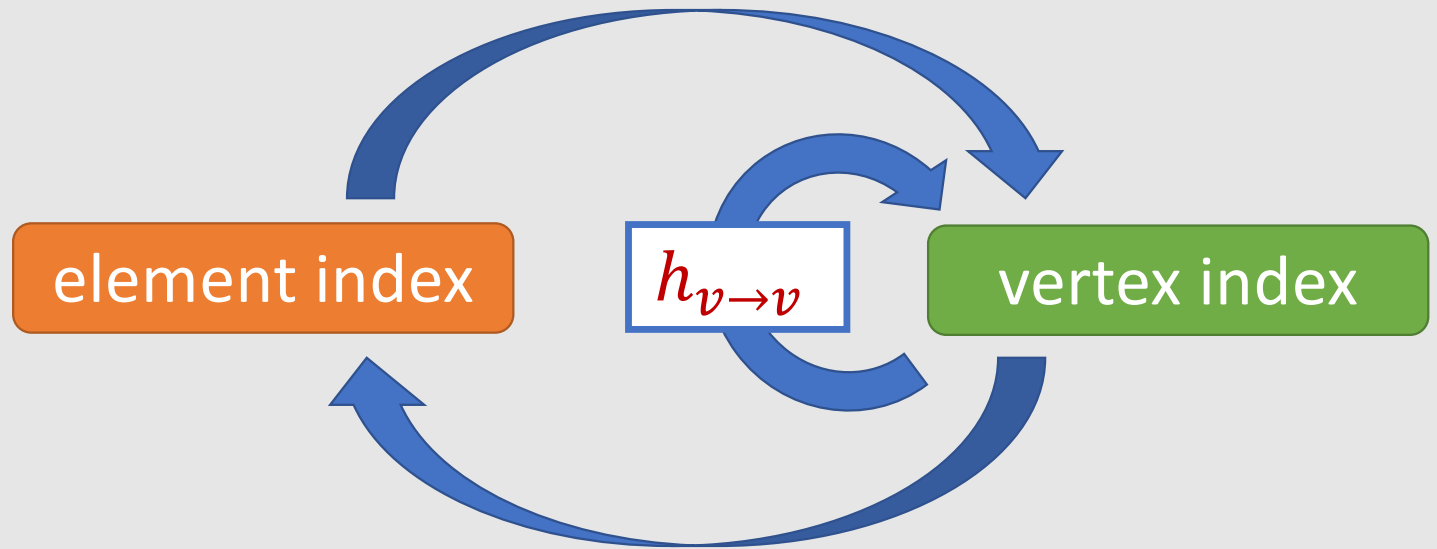
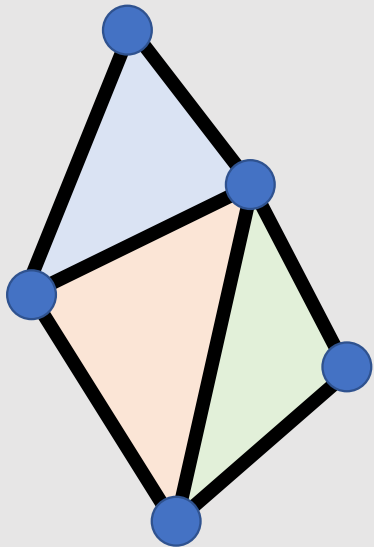
- FEM, FDM produce a very sparse matrix (e.g., 30 entries per row)



(Wikipedia: Sparse Matrix)

Constructing One-Ring Neighborhood Graph

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



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

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

Coordinate (COO) Data Structure

We are triplet!



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

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



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

Compressed Sparse Row (**CSR**) Data Structure

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

