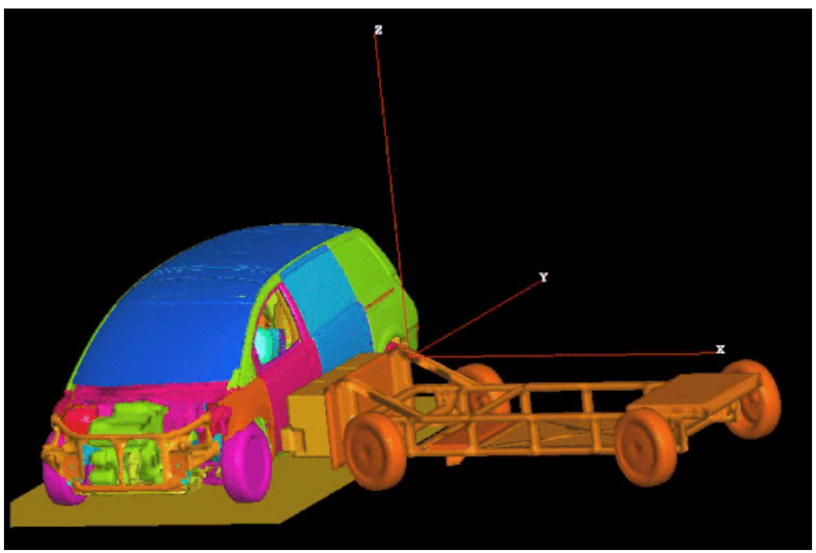
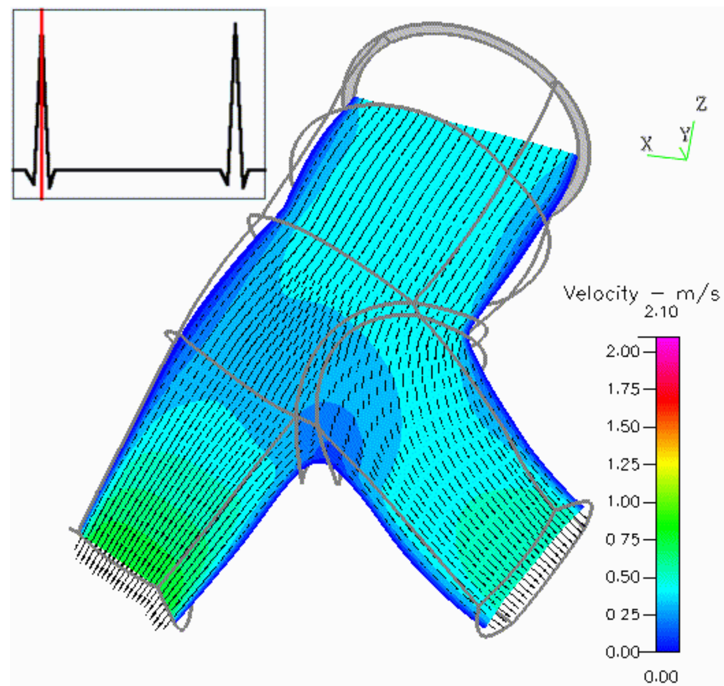
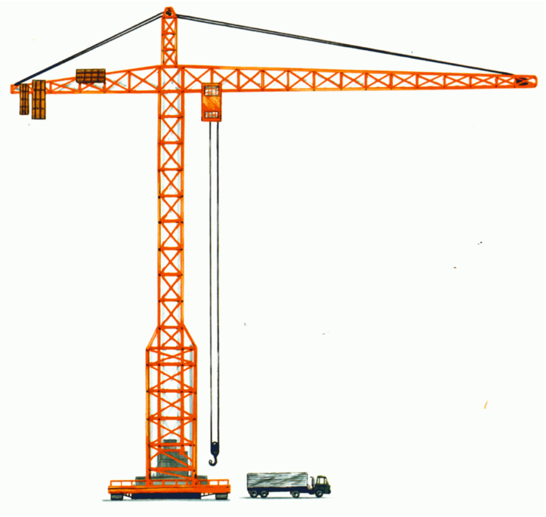
Sparse Matrices: Goals

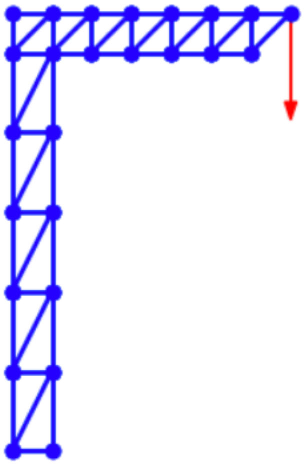
- **Perform standard matrix computations economically, i.e., without storing the zeros of the matrix.**
- For typical Finite Element and Finite Difference matrices, the number of non-zero entries is $O(n)$

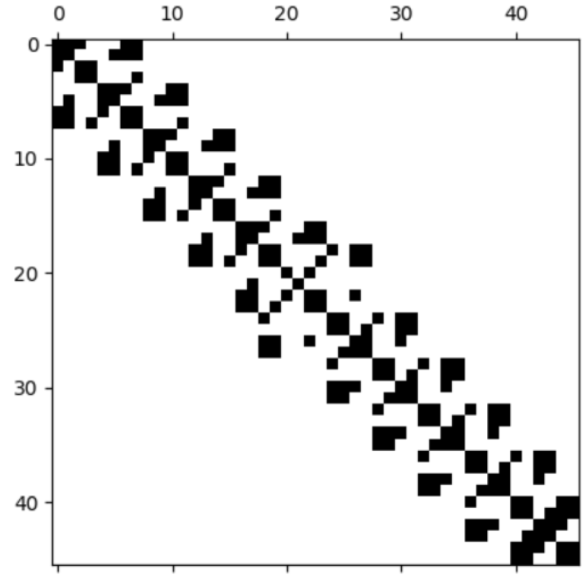




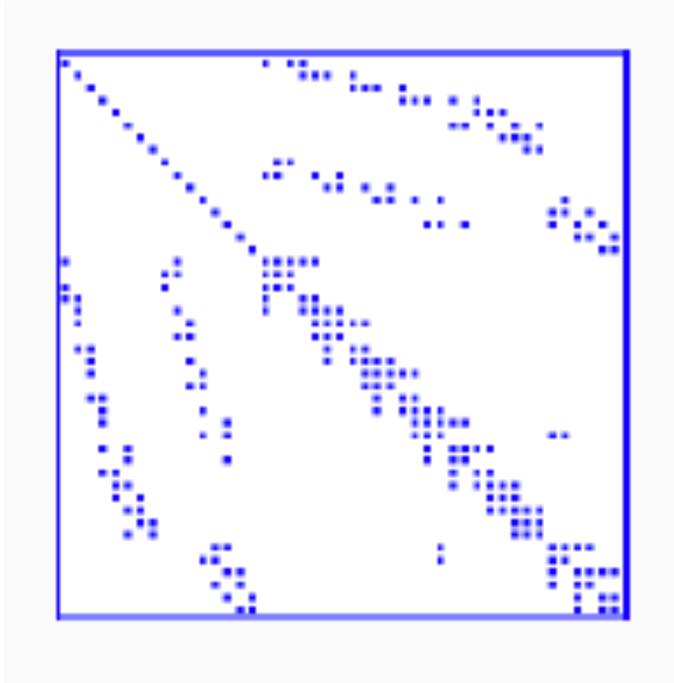
Sparse Matrices: MP example







Sparse Matrices



Some type of matrices contain many zeros.
Storing all those zero entries is wasteful!

How can we efficiently store large
matrices without storing tons of zeros?

- **Sparse matrices** (vague definition): matrix with few non-zero entries.
- For practical purposes: an $m \times n$ matrix is sparse if it has $O(\min(m, n))$ non-zero entries.
- This means roughly a constant number of non-zero entries per row and column.
- Another definition: "matrices that allow special techniques to take advantage of the large number of zero elements" (J. Wilkinson)

Sparse Matrices

EXAMPLE:

Number of operations required to add two square dense matrices:

$$O(n^2)$$

Number of operations required to add two sparse matrices $\mathbf{A}$ and $\mathbf{B}$:

$$O(\mathrm{nnz}(\mathbf{A}) + \mathrm{nnz}(\mathbf{B}))$$

where $\mathrm{nnz}(\mathbf{X}) =$ number of non-zero elements of a matrix $\mathbf{X}$

Popular Storage Structures

DNS	Dense	**ELL**	Ellpack-Itpack
BND	Linpack Banded	**DIA**	Diagonal
COO	Coordinate	**BSR**	Block Sparse Row
CSR	Compressed Sparse Row	**SSK**	Symmetric Skyline
CSC	Compressed Sparse Column	**BSR**	Nonsymmetric Skyline
MSR	Modified CSR	**JAD**	Jagged Diagonal
LIL	Linked List		

note: CSR = CRS, CCS = CSC, SSK = SKS in some references

We will focus on COO and CSR!

Dense (DNS)

$$A = \begin{bmatrix} 0. & 1.9 & 0. & -5.2 \\ 0.3 & 0. & 9.1 & 0. \\ 4.4 & 5.8 & 3.6 & 0. \\ 0. & 0. & 7.2 & 2.7 \end{bmatrix}$$

$$Ashape = (nrow, ncol)$$

$$A_{dense} = [\underbrace{0. \quad 1.9 \quad 0. \quad -5.2}_{\text{Row 0}} \quad \underbrace{0.3 \quad 0. \quad 9.1 \quad 0.}_{\text{Row 1}} \quad \underbrace{4.4 \quad 5.8 \quad 3.6 \quad 0.}_{\text{Row 2}} \quad \underbrace{0. \quad 0. \quad 7.2 \quad 2.7}_{\text{Row 3}}]$$

- Simple
- Row-wise
- Easy blocked formats
- Stores all the zeros

Coordinate (COO)

$$A = \begin{array}{cc} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} & \\ \begin{bmatrix} 0. & 1.9 & 0. & -5.2 \\ 0.3 & 0. & 9.1 & 0. \\ 4.4 & 5.8 & 3.6 & 0. \\ 0. & 0. & 7.2 & 2.7 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array}$$

data = [-5.2 0.3 9.1 1.9 5.8 7.2 3.6 4.4 2.7]

$$data = \begin{bmatrix} 1.9 & -5.2 & 0.3 & 9.1 & 4.4 & 5.8 & 3.6 & 7.2 & 2.7 \end{bmatrix}$$

$$row = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \end{bmatrix}$$

$$col = \begin{bmatrix} 1 & 3 & 0 & 2 & 0 & 1 & 2 & 2 & 3 \end{bmatrix}$$

- Simple
- Does not store the zero elements
- Not sorted
- *row* and *col*: array of integers
- *data*: array of doubles

row = [0 1 1 0 2 3 2 2 3]

col = [3 0 2 1 1 2 2 0 3]

Iclicker question

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 & 0 \\ 6 & 0 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

$$\begin{array}{lclcccccccccccccl} data & = & [& 12.0 & 9.0 & 7.0 & 5.0 & 1.0 & 2.0 & 11.0 & 3.0 & 6.0 & 4.0 & 8.0 & 10.0 &] \\ row & = & [& 4 & 2 & 2 & 1 & 0 & 0 & 3 & 1 & 2 & 1 & 2 & 3 &] \\ col & = & [& 4 & 4 & 2 & 3 & 0 & 3 & 3 & 0 & 0 & 1 & 3 & 2 &] \end{array}$$

How many integers are stored in COO format ($\boldsymbol{A}$ has dimensions $n \times n$)?

A) nnz
B) n
C) $\boxed{2\ nnz}$
D) n^2
E) $2\ n$

row $\rightarrow$ nnz $\rightarrow$ int
col $\rightarrow$ nnz $\rightarrow$ int
data $\rightarrow$ nnz $\rightarrow$ float

Iclicker question

Representing a Sparse Matrix in Coordinate (COO) Form

1 point

Consider the following matrix:

$$A = \begin{bmatrix} 0 & 0 & 1.3 \\ -1.5 & 0.2 & 0 \\ 5 & 0 & 0 \\ 0 & 0.3 & 3 \\ 0 & 0 & 0 \end{bmatrix}$$

Suppose we store one row index (a 32-bit integer), one column index (a 32-bit integer), and one data value (a 64-bit float) for each non-zero entry in A. How many bytes in total are stored? Please note that 1 byte is equal to 8 bits.

A) 56 bytes
B) 72 bytes
C) 96 bytes
D) 120 bytes
E) 144 bytes

$$\underbrace{2\,nnz * 32}_{\text{int}} + \underbrace{nnz * 64}_{\text{float}} = 12*32 + 6*64 = \frac{768}{8} = 96 \text{ bytes}$$

Compressed Sparse Row (CSR)

$$
A = \begin{array}{c}
\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 2 & 0 \\
3 & 4 & 0 & 5 & 0 \\
6 & 0 & 7 & 8 & 9 \\
0 & 0 & 10 & 11 & 0 \\
0 & 0 & 0 & 0 & 12
\end{bmatrix}
\end{array}
\begin{array}{l}
0 \longrightarrow 2 \text{ non-zero} \\
1 \longrightarrow 3 \text{ 〃} \\
2 \longrightarrow 4 \\
3 \longrightarrow 2 \\
4 \longrightarrow 1
\end{array}
$$

float(nnz)

$$
data = [\; \underbrace{1.0 \quad 2.0}_{\text{row 0}} \quad \underbrace{3.0 \quad 4.0 \quad 5.0}_{\text{row 1}} \quad \underbrace{6.0 \quad 7.0 \quad 8.0 \quad 9.0}_{\text{row 2}} \quad \underbrace{10.0 \quad 11.0}_{\text{row 3}} \quad \underbrace{12.0}_{\text{row 4}} \;]
$$

int(nnz)

$$
col = [\; 0 \quad 3 \quad 0 \quad 1 \quad 3 \quad 0 \quad 2 \quad 3 \quad 4 \quad 2 \quad 3 \quad 4 \;]
$$

float (number of rows)

$$
rowptr = [\; 0 \xrightarrow{+2} 2 \xrightarrow{+3} 5 \xrightarrow{+4} 9 \xrightarrow{+2} 11 \xrightarrow{+1} 12 \;]
$$

nnz in row 0

nnz in row 1

row 2

row 3

row 4

always zero

total number of non-zero entries

Compressed Sparse Row (CSR)

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 & 0 \\ 6 & 0 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

$$\begin{array}{lllllllllllllll} data & = & [& 1.0 & 2.0 & 3.0 & 4.0 & 5.0 & 6.0 & 7.0 & 8.0 & 9.0 & 10.0 & 11.0 & 12.0 &] \\ col & = & [& 0 & 3 & 0 & 1 & 3 & 0 & 2 & 3 & 4 & 2 & 3 & 4 &] \\ rowptr & = & [& 0 & 2 & 5 & 9 & 11 & 12 &] \end{array}$$

- Does not store the zero elements
- Fast arithmetic operations between sparse matrices, and fast matrix-vector product
- *col*: contain the column indices (array of nnz integers)
- *data*: contain the non-zero elements (array of nnz doubles)
- *rowptr*: contain the row offset (array of $n + 1$ integers)

Example - CSR format

Back to Image-Blur

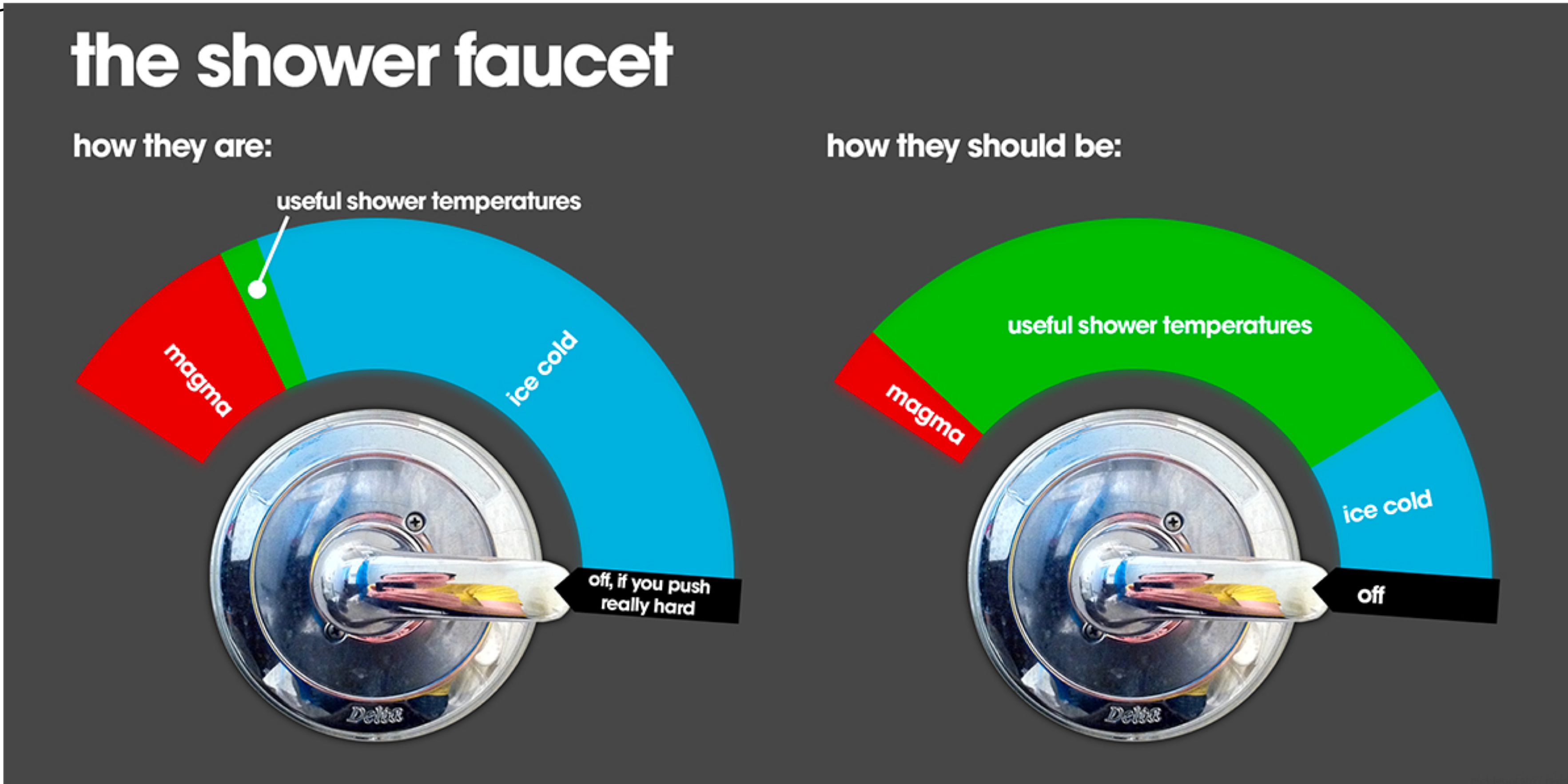
* How perturbation "slowly" creeps in and affects the perturbation in the output

$$A = \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0. & 1.9 & 0. & -5.2 \\ 0. & 0. & 0. & 0. \\ 4.4 & 5.8 & 3.6 & 0. \\ 0. & 0. & 7.2 & 2.7 \end{bmatrix} \end{array} \begin{array}{l} \\ \to 2 \\ \to 0 \\ \to 3 \\ \to 2 \end{array}$$

data = [1.9 -5.2 4.4 5.8 3.6 7.2 2.7]

col = [1 3 0 1 2 2 3]

rowptr = [0 2 2 5 7]


the shower faucet
how they are:
useful shower temperatures
magma
ice cold
off, if you push really hard
how they should be:
useful shower temperatures
magma
ice cold
off


WHAT IT LOOKS LIKE
WHAT IT FEELS LIKE

Numerical experiments

Input has uncertainties:

- Errors due to representation with finite precision
- Error in the sampling

Once you select your numerical method , how much error should you expect to see in your **output?**

Is your method sensitive to errors (perturbation) in the input?

Demo "HilbertMatrix-ConditionNumber"

Hilbert Matrix

$$\begin{pmatrix} & & \\ & \underset{\approx}{A} & \\ & & \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \underset{\sim}{b} \\ \\ \\ \end{pmatrix} \qquad \underset{\approx}{A}\,\underset{\sim}{x} = \underset{\sim}{b}$$

$\hookrightarrow$ this is the exact solution $\underset{\sim}{x}$

→ generate random $\underset{\approx}{A}$, use mat-vec to get $\underset{\sim}{b}$

→ "undo" button

→ $A\hat{x} = b \longrightarrow \hat{x}$ is the approx solution (solve!)

→ error $= \|\hat{x} - x\|_2$

Sensitivity of Solutions of Linear Systems

Suppose we start with a non-singular system of linear equations $\boldsymbol{A\,x} = \boldsymbol{b}$.

We change the right-hand side vector $\boldsymbol{b}$ (input) by a small amount $\Delta\boldsymbol{b}$.

How much the solution $\boldsymbol{x}$ (output) changes, i.e., how large is $\Delta\boldsymbol{x}$?

$$\frac{\text{Output Relative error}}{\text{Input Relative error}} = \frac{\|\Delta\boldsymbol{x}\|/\|\boldsymbol{x}\|}{\|\Delta\boldsymbol{b}\|/\|\boldsymbol{b}\|} = \frac{\|\Delta\boldsymbol{x}\|\;\|\boldsymbol{b}\|}{\|\Delta\boldsymbol{b}\|\;\|\boldsymbol{x}\|}$$

$\cancel{Ax} + A\Delta x = \cancel{b} + \Delta b$

$$\boldsymbol{A\,\hat{x}} = \boldsymbol{\hat{b}} \;\rightarrow\; \boldsymbol{A\,\hat{x}} = \boldsymbol{A}(\boldsymbol{x} + \Delta\boldsymbol{x}) = (\boldsymbol{b} + \Delta\boldsymbol{b}) \;\rightarrow\; \boldsymbol{A}\,\Delta\boldsymbol{x} = \Delta\boldsymbol{b}$$

$$\frac{\text{Output Relative error}}{\text{Input Relative error}} = \frac{\overbrace{\|\boldsymbol{A}^{-1}\,\Delta\boldsymbol{b}\|}^{\Delta x}\;\overbrace{\|\boldsymbol{A\,x}\|}^{b}}{\|\Delta\boldsymbol{b}\|\;\|\boldsymbol{x}\|} \le \frac{\|\boldsymbol{A}^{-1}\|\,\cancel{\|\Delta\boldsymbol{b}\|}\;\|\boldsymbol{A}\|\,\cancel{\|\boldsymbol{x}\|}}{\cancel{\|\Delta\boldsymbol{b}\|}\;\cancel{\|\boldsymbol{x}\|}}$$

$= \|A^{-1}\|\,\|A\|$

$$\frac{\|\Delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le \|\boldsymbol{A}^{-1}\|\;\|\boldsymbol{A}\|\;\frac{\|\Delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}$$

Sensitivity of Solutions of Linear Systems

We can also add a perturbation to the matrix $\boldsymbol{A}$ (input) by a small amount $\boldsymbol{E}$, such that

$$(\boldsymbol{A} + \boldsymbol{E})\,\hat{\boldsymbol{x}} = \boldsymbol{b}$$

and in a similar way obtain:

$$\frac{\|\Delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{A}\|\,\frac{\|\boldsymbol{E}\|}{\|\boldsymbol{A}\|}$$

Condition number

Demo "HilbertMatrix-ConditionNumber"

get the condition number.

The condition number is a measure of sensitivity of solving a linear system of equations to variations in the input.

→ High cond(A)!

→ seems to correlate well

The condition number of a matrix $\boldsymbol{A}$:

$$cond(\boldsymbol{A}) = \|\boldsymbol{A}^{-1}\| \, \|\boldsymbol{A}\|$$

Recall that the induced matrix norm is given by

$$\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|$$

And since the condition number is relative to a given norm, we should be precise and for example write:

$$cond_2(\boldsymbol{A}) \text{ or } cond_\infty(\boldsymbol{A})$$

Iclicker question

$$\frac{\|\Delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq cond(\boldsymbol{A}) \frac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|}$$

Give an example of a matrix that is very well-conditioned (i.e., has a condition number that is good for computation). Select the best possible condition number(s) of a matrix?

$cond(A) = \|A\|\|A^{-1}\| > 0$

~~$A)\ cond(\boldsymbol{A}) < 0$~~

$B)\ cond(\boldsymbol{A}) = 0$

$C)\ 0 < cond(\boldsymbol{A}) < 1$

why not these answers?

$D)\ cond(\boldsymbol{A}) = 1$

~~$E)\ cond(\boldsymbol{A}) =$ large numbers~~

this will amplify perturbations in the output

Condition number

small

$$\frac{||\Delta \boldsymbol{x}\,||}{||\boldsymbol{x}||} \le cond(\boldsymbol{A}) \frac{||\Delta \boldsymbol{b}||}{||\boldsymbol{b}||}$$

Small condition numbers mean not a lot of error amplification. Small condition numbers are good!

The identity matrix should be well-conditioned:

$cond(\underline{\underline{I}}) = ||\underline{\underline{I}}||\,||\underline{\underline{I}}^{-1}|| = 1$

$$||\boldsymbol{I}|| = \max_{||\boldsymbol{x}||=1} ||\boldsymbol{I}\,\boldsymbol{x}|| = 1$$

It turns out that this is the smallest possible condition number:

$\boxed{cond(A) \ge 1}$

$$cond(\boldsymbol{A}) = ||\boldsymbol{A}^{-1}||\,||\boldsymbol{A}|| \ge ||\boldsymbol{A}^{-1}\boldsymbol{A}|| = ||\boldsymbol{I}|| = 1$$

If $\boldsymbol{A}^{-1}$ does not exist, then $cond(\boldsymbol{A}) = \infty$ (by convention)

(SINGULAR)

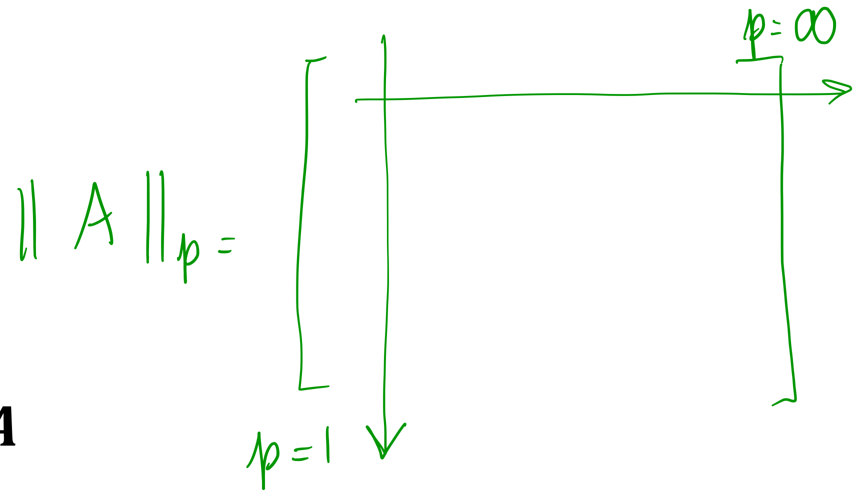
Recall Induced Matrix Norms

$$\|\boldsymbol{A}\|_1 = \max_j \sum_{i=1}^{n} |A_{ij}|$$

Maximum absolute column sum of the matrix $\boldsymbol{A}$

$$\|\boldsymbol{A}\|_\infty = \max_i \sum_{j=1}^{n} |A_{ij}|$$

Maximum absolute row sum of the matrix $\boldsymbol{A}$



$$\|\boldsymbol{A}\|_2 = \max_k \sigma_k$$

σ_k are the singular value of the matrix $\boldsymbol{A}$

Iclicker question

Condition Number of a Diagonal Matrix

What is the 2-norm-based condition number of the diagonal matrix

$$A = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 13 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}?$$

A) 1
B) *50*
C) 100
D) 200

$$\|A\|_2 = \max_k \sigma_k = \max(100, 13, 0.5) = 100$$

$$\|A^{-1}\|_2 = \max_k \sigma_k = \max\left(\frac{1}{100}, \frac{1}{13}, \frac{1}{0.5}\right) = 2$$

$$\text{cond}_2(A) = \|A\| \, \|A^{-1}\|_2 = 200$$

in general $\text{cond}_2(D) = \frac{D_{max}}{D_{min}}$

“Little c” demo

Discuss what happens when c is ”close” to zero
What are the eigenvalues of triangular matrices?

only the first part

Remarks:
The need for pivoting does not depend on whether the matrix is singular.
A non-singular matrix always has a solution.
A singular matrix may not have a solution, or may have infinitely many solutions.

About condition numbers

1. For any matrix $\boldsymbol{A}$, $cond(\boldsymbol{A}) \geq 1$

2. For the identity matrix $\boldsymbol{I}$, $cond(\boldsymbol{I}) = 1$

3. For any matrix $\boldsymbol{A}$ and a nonzero scalar γ, $cond(\gamma \boldsymbol{A}) = cond(\boldsymbol{A})$

4. For any diagonal matrix $\boldsymbol{D}$, $cond(\boldsymbol{D}) = \frac{\boldsymbol{max}|d_i|}{\boldsymbol{min}|d_i|}$

5. The condition number is a measure of how close a matrix is to being singular: a matrix with large condition number is nearly singular, whereas a matrix with a condition number close to 1 is far from being singular

6. The determinant of a matrix is NOT a good indicator is a matrix is near singularity if $det(A) = 0 \longrightarrow$ singular

Residual versus error

Our goal is to find the solution $\boldsymbol{x}$ to the linear system of equations $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$

Let us recall the solution of the perturbed problem

exact

$$\hat{\boldsymbol{x}} = (\boldsymbol{x} + \Delta\boldsymbol{x})$$

approx

which could be the solution of

$$\boldsymbol{A}\,\hat{\boldsymbol{x}} = (\boldsymbol{b} + \Delta\boldsymbol{b}), \qquad (\boldsymbol{A} + \boldsymbol{E})\hat{\boldsymbol{x}} = \boldsymbol{b}, \qquad (\boldsymbol{A} + \boldsymbol{E})\,\hat{\boldsymbol{x}} = (\boldsymbol{b} + \Delta\boldsymbol{b})$$

And the **error vector** as

$$\boldsymbol{e} = \Delta\boldsymbol{x} = \hat{\boldsymbol{x}} - \boldsymbol{x}$$

but we usually don't have $\underset{\sim}{x}$!!

We can write the **residual vector** as

$$\boxed{\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\,\hat{\boldsymbol{x}}}$$

known quantity

BACK TO HILBERT

Residual versus error

Our goal is to find the solution $\boldsymbol{x}$ to the linear system of equations $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$

It can be shown that the residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\,\hat{\boldsymbol{x}}$ satisfies the following:

$$\boxed{\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{A}\|\,\|\hat{\boldsymbol{x}}\|} \leq c\,\epsilon_m}$$

Where c is large without pivoting and small with partial pivoting.

Therefore, solving the system of equations with partial pivoting yields **small relative residual regardless of conditioning of the system**.

Residual versus error

Relative residual: $\boxed{\dfrac{||\boldsymbol{r}||}{||\boldsymbol{A}||\,||\boldsymbol{x}||}}$ **(How well the solution satisfies the problem)**

This is what I can measure

?

Relative error: $\dfrac{||\Delta\boldsymbol{x}||}{||\boldsymbol{x}||}$ **(How close the approximated solution is from the exact one)**

This is what I am interested in!

When solving a system of linear equations via LU with partial pivoting, the relative residual is guaranteed to be small!

For well-conditioned matrices, small relative residual implies small relative error.

$$\boxed{\frac{||\Delta\boldsymbol{x}||}{||\boldsymbol{x}||} \le cond(\boldsymbol{A})\,\frac{||\boldsymbol{r}||}{||\boldsymbol{A}||\,||\boldsymbol{x}||}}$$

When solving $Ax=b$ with LU + partial pivoting, the error will be small for well-conditioned A!

Rule of thumb for conditioning

Suppose we want to find the solution $\boldsymbol{x}$ to the linear system of equations $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ using LU factorization with partial pivoting and backward/forward substitutions.

Suppose we compute the solution $\widehat{\boldsymbol{x}}$.

If the entries in $\boldsymbol{A}$ and $\boldsymbol{b}$ are accurate to S decimal digits,

and $cond(\boldsymbol{A}) = \mathbf{10}^{\boldsymbol{W}}$,

then the elements of the solution vector $\widehat{\boldsymbol{x}}$ will be accurate to about

$$S - W$$

decimal digits

Iclicker question

DEMO

Matrix Conditioning: Accurate digits

1 point

Let's say we want to solve the following linear system:

$$Ax = b$$

Assuming you are working with IEEE double precision floating point numbers, how many digits of accuracy will your answer have if $\kappa(A) = 1000$?

A) 3
B) 10
C) 13
D) 16
E) 32

$\underline{\underline{A}}, \underline{b} \rightarrow$ 16 digits of accuracy

$\text{cond}(\underline{\underline{A}}) = 10^3$

Hence $\underset{\sim}{x}$ will have $16-3=13$ digits of accuracy.