ENGR 21:
Computer Engineering Fundamentals

Lecture 16
Thursday, October 30, 2025

Summary of LU Decomposition as a technique for solving Ax=b

- It is possible to write the matrix A as the product of a lower-triangular matrix L and an upper triangular matrix U.

(Doolittle's method for finding an L and a U)

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- To find the elements of U, use Gaussian elimination on A. The end result is U.
- To find the elements of L, note that the multiplier that eliminated the element A_{pq} is equal to the element L_{pq} of L.

What do we do with L and U ?

$$Ax = b \implies \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \underbrace{\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\text{Call this vector y}} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

1. Solve this for y

$$\begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \longrightarrow \text{forward substitution}$$

2. Solve this for x

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \longrightarrow \text{backward substitution.}$$

Iterative Method:
The Gauss-Seidel Method for $\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

General idea: 3 equations, 3 unknowns.

$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 = b_1 \implies x_1 = \frac{1}{A_{11}}(b_1 - A_{12}\underline{x_2} - A_{13}\underline{x_3})$$

$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 = b_2 \implies x_2 = \frac{1}{A_{22}}(b_2 - A_{21}\underline{x_1} - A_{23}\underline{x_3})$$

$$A_{31}x_1 + A_{32}x_2 + A_{33}x_3 = b_3 \implies x_3 = \frac{1}{A_{33}}(b_3 - A_{31}\underline{x_1} - A_{32}\underline{x_2})$$

What's the problem? right-hand side still has unknowns.

↳ solution : guess! [but only for right side]

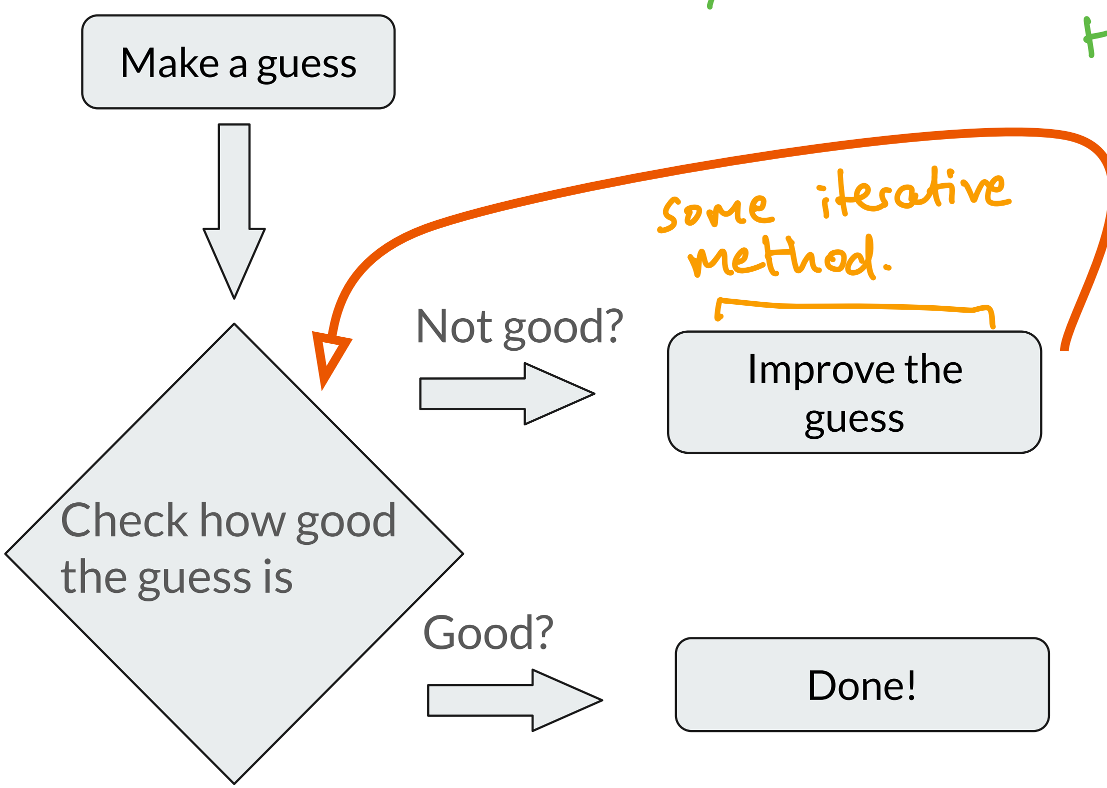
The iterative process of solving Ax=b (for any iterative method)

Start with a guess

Quit when good enough

$$\mathbf{x}^{(0)} \to \mathbf{x}^{(1)} \to \mathbf{x}^{(2)} \to \mathbf{x}^{(3)} \to \mathbf{x}^{(4)} \to \mathbf{x}^{(5)} \to \mathbf{x}^{(6)} \to \dots$$

$\longrightarrow$ only works if $\vec{x}^{(3)}$ is a better guess than $\vec{x}^{(2)}$ and, generally if $\vec{x}^{(n+1)}$ is a better guess than $\vec{x}^{(n)}$



How do you check 'good enough'

$$\vec{x}^{(6)} \longrightarrow \vec{x}^{(7)}$$

1. Verify that difference between $\vec{x}^{(7)}$ and $\vec{x}^{(6)}$ is very small.
2. Verify that equation $A\vec{x} = \vec{b}$ is approximately satisfied by $\vec{x}^{(7)}$

Checking how good a guess is: the Residual

Given a vector x, how do we quantify how well it satisfies Ax = b?

$$\boxed{\vec{r} \equiv A\vec{x}^{(k)} - \vec{b}}$$

Residual at the k^{th} guess

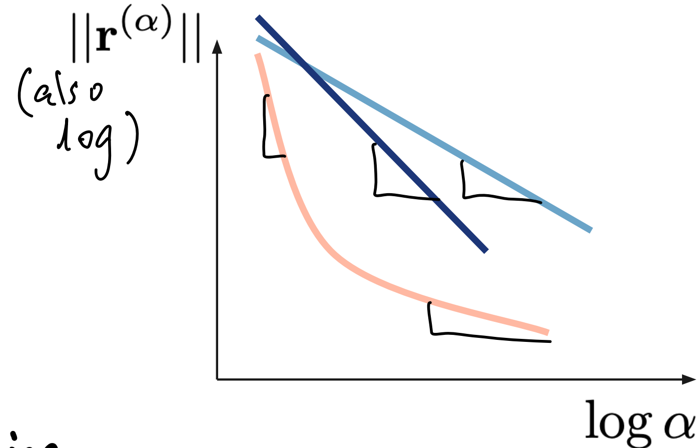
if $\vec{r} = \vec{0}$
then $\vec{x}^{(k)}$ exactly satisfies $A\vec{x} = \vec{b}$
if $\vec{r}$ is close to zero, $\vec{r} \approx \vec{0}$
then $\vec{x}^{(k)}$ approximately satisfies $A\vec{x} = \vec{b}$

$$\|\vec{a}\| = \sqrt{\sum_{i=1}^{N} a_i^2}$$

in practice, for iterative methods of solving $A\vec{x} = \vec{b}$,
we check : if $\left(\|\vec{r}\| < \text{some small number called 'tolerance' e.g. } 10^{-6}\text{, etc.} \right)$, done !

Quantifying the performance of an iterative scheme for solving Ax = b

Observe the decay of the norm of the residual after α iterations



Slope of this graph contains information about how quickly residual is decaying to zero, i.e how quickly $\vec{x}^{(\alpha)}$ is approaching the true solution $\vec{x} = A^{-1}\vec{b}$

α: step number or iteration number.

$$||\mathbf{a}|| \equiv \sqrt{\sum_{i=1}^{i=N} a_i^2}$$

$\vec{x}^{(0)} = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix}$ can be anything.

The Gauss-Seidel method, an iterative technique for solving Ax = b

check if $\|\vec{r}\| > \epsilon$

while [....]

for [....] loop over rows

for [....] build summation by looping over columns

if [...], check $i \neq j$

$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 = b_1 \Longrightarrow x_1 = \frac{1}{A_{11}}(b_1 - A_{12}x_2 - A_{13}x_3)$$

$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 = b_2 \Longrightarrow x_2 = \frac{1}{A_{22}}(b_2 - A_{21}x_1 - A_{23}x_3)$$

$$A_{31}x_1 + A_{32}x_2 + A_{33}x_3 = b_3 \Longrightarrow x_3 = \frac{1}{A_{33}}(b_3 - A_{31}x_1 - A_{32}x_2)$$

The "update formula":

$$x_i = \frac{1}{A_{ii}}\left[b_i - \sum_{j=1,\neq i}^{N} A_{ij}x_j\right]$$

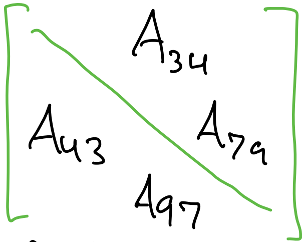
Other Iterative Methods

- Conjugate Gradient algorithm (see Resources page for code) → State of the art.
- `np.linalg.solve(A,b)` → container for vast library of methods known to numpy.
- Many others

Convergence of Iterative Methods

- If each subsequent iteration gets closer to true value → Method is convergent
- Iterative methods are useless if they are not convergent!
- Is Gauss-Seidel convergent?

↳ 2 conditions



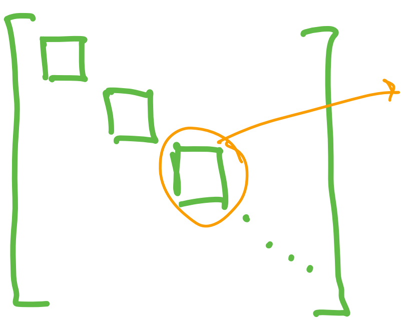
if $A_{ij} = A_{ji}$ then A is symmetric

(1) A is $\underbrace{\text{symmetric}}$ $\underbrace{\text{positive-definite}}_{\text{all eigenvalues of } A \text{ are strictly positive}}$

OR

(2) A is diagonally dominant, i.e.

for each row, magnitude of the term on the diagonal is greater than sum of magnitudes of all off-diagonal terms in that row.



An NxN matrix has N eigenvalues.

if either condition is met, Gauss-Siedel is convergent.