System of Linear Equations

$$Ax = b$$

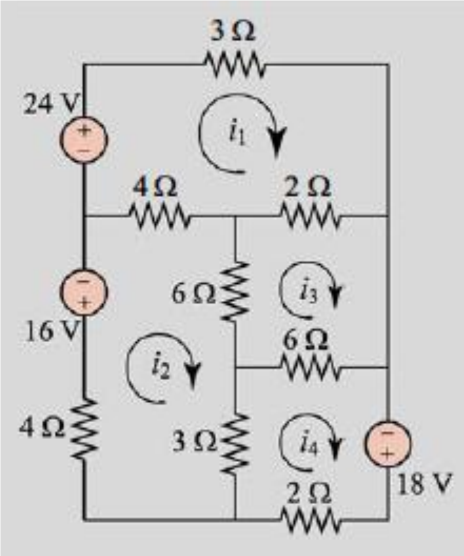
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Scope

- We've all solved three equations three unknown problems
  - What if we have a hundred or million equations and unknowns?
- A solution to a system of linear equations finds the the vector ***x*** (causes) that led to the vector of ***b*** (effects) under the action of a linear operator ***A***
  - The solution may or may not exist
  - If there is a solution, it may not be unique
- General case:
  - A is an $m \times n$ matrix, *x* is an *n*-vector, *b* is an *m*-vector
  - m>n, overdetermined system
    - Linear least square, data fitting
  - m<n, underdetermined system
    - No solution, or infinite number of solutions
    - Incomplete, noisy data (e.g. image reconstruction)
- In this chapter we will focus on m=n (square systems)

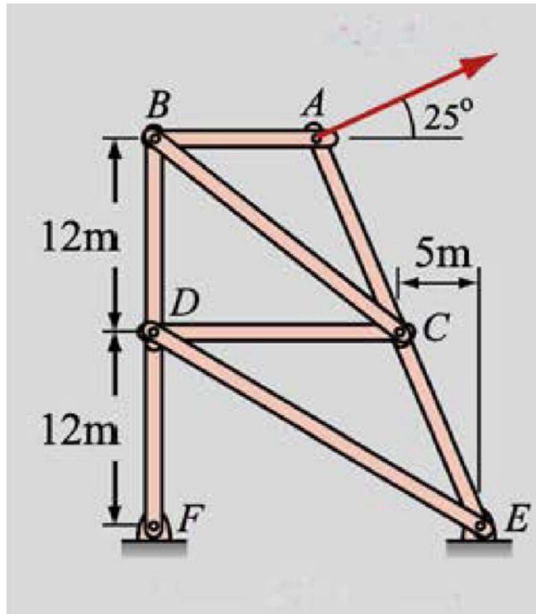
Engineering Applications

Electric circuit analysis via loop method (KVL)



$$
\begin{aligned}
9i_1 - 4i_2 - 2i_3 &= 24 \\
-4i_1 + 17i_2 - 6i_3 - 3i_4 &= -16 \\
-2i_1 - 6i_2 + 14i_3 - 6i_4 &= 0 \\
-3i_2 - 6i_3 + 11i_4 &= 18
\end{aligned}
$$

Static Equilibrium in 8-member truss



$$0.9231F_{AC} = 1690$$

$$-F_{AB} - 0.3846F_{AC} = 3625$$

$$F_{AB} - 0.7809F_{BC} = 0$$

$$0.6247F_{BC} - F_{BD} = 0$$

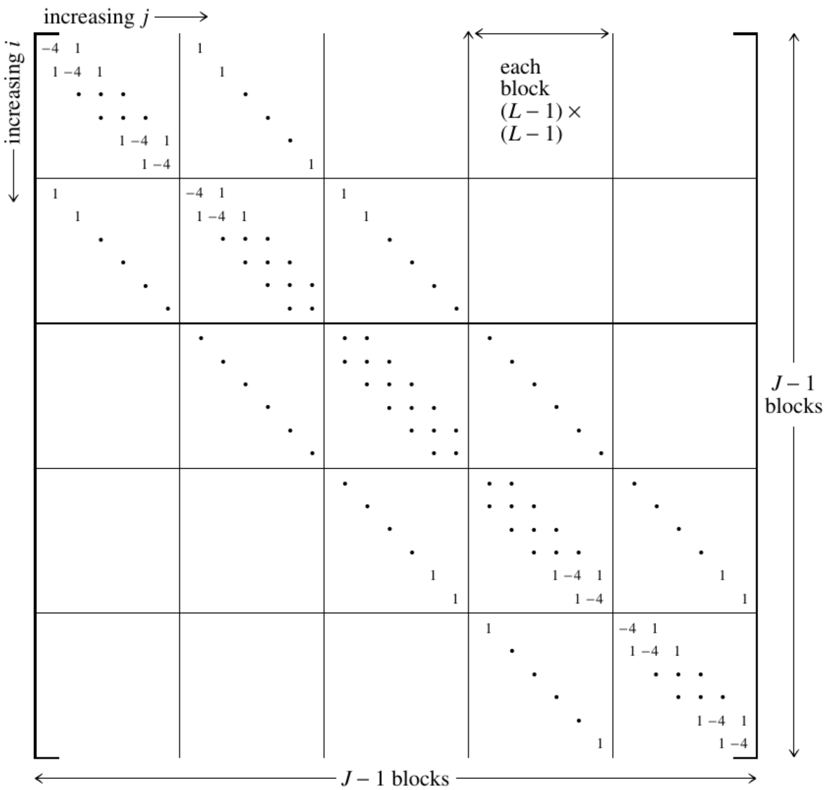
$$F_{CD} + 0.8575F_{DE} = 0$$

$$F_{BD} - 0.5145F_{DE} - F_{DF} = 0$$

$$0.3846F_{CE} - 0.3846F_{AC} - 0.7809F_{BC} - F_{CD} = 0$$

$$0.9231F_{AC} + 0.6247F_{BC} - 0.9231F_{CE} = 0$$

Solving linear PDEs



Poisson's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$$

Finite difference method

$$x_j = x_0 + j\Delta, \qquad j = 0, 1, \ldots, J$$
$$y_l = y_0 + l\Delta, \qquad l = 0, 1, \ldots, L$$

Apply second order accurate central differencing formula

$$\Rightarrow \quad u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{j,l} = \Delta^2 \rho_{j,l}$$

$$\Rightarrow \quad \mathbf{A} \cdot \mathbf{u} = \mathbf{b}$$

Existence and Uniqueness

- An $n \times n$ matrix $\boldsymbol{A}$ is nonsingular if it satisfies *any* one of the following
    - $\boldsymbol{A}$ has an inverse $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$ (identity matrix)
    - $\det(\boldsymbol{A}) \neq 0$
    - rank($\boldsymbol{A}$) =n
    - For any nonzero vector $\boldsymbol{z} \neq 0$, $\boldsymbol{A}\boldsymbol{z} \neq 0$
- Otherwise → the matrix is singular

- The rank of a matrix is the maximum number of linearly independent rows or columns
    - There are methods to determine the rank of a matrix
    - If an $n \times n$ matrix $\boldsymbol{A}$ has a rank lower than *n*, it is not invertible, meaning A is singular.

Sensitivity and Conditioning of $Ax = b$

- How sensitive is solution **x** to changes/perturbations in **A** and **b**?
- We need a notion of "size" for a matrix A and a vector b
  - Hence the concept of "norms" $\|\cdot\|$
  - Size in the sense of magnitude or distance and not number of rows or columns

Common Vector Norms

$$\mathbf{1-norm}: \ \|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

$$\mathbf{2-norm}: \ \|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}} = \sqrt{\mathbf{x}^T\mathbf{x}}$$

$$\boldsymbol{\infty-norm}: \ \|\mathbf{x}\|_\infty = \max_{i=1\ldots n} |x_i|$$

Properties of vector norms

1. $\|\mathbf{x}\| \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$
2. $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = 0$
3. $\|\alpha \mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$ for any $\mathbf{x} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$
4. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ (the triangle inequality)

$$\|\mathbf{x}\| - \|\boldsymbol{y}\| \leq \|\mathbf{x} - \mathbf{y}\|$$

$$\|\mathbf{x}\|_1 \geq \|\mathbf{x}\|_2 \geq \|\mathbf{x}\|_\infty$$

$$\|\mathbf{x}\|_1 \leq \sqrt{n}\, \|\mathbf{x}\|_2, \qquad \|\mathbf{x}\|_2 \leq \sqrt{n}\, \|\mathbf{x}\|_\infty, \qquad \|\mathbf{x}\|_1 \leq n\, \|\mathbf{x}\|_\infty$$

Unit vector: $\|\mathbf{x}\| = 1$

A *unit vector* can be obtained from any nonzero vector as $\dfrac{\mathbf{u}}{\|\mathbf{u}\|}$

Example: $u = [1\ \ -5\ \ 3]$, $\mathrm{v} = \begin{bmatrix} -2 \\ 2 \\ 0 \end{bmatrix}$

Shape of vector doesn't matter

1-norm

$$\|u\|_1 = |1| + |-5| + |3| = 9, \quad \|v\|_1 = |-2| + |2| + 0 = 4$$

2-norm

$$\|u\|_2 = (|1|^2 + |-5|^2 + |3|^2)^{1/2} = \sqrt{35},$$

$$\|v\|_2 = (|-2|^2 + |2|^2 + 0)^{1/2} = \sqrt{8} = 2\sqrt{2}$$

∞-norm

$$\|u\|_\infty = \max(|1|, |-5|, |3|) = 5$$

Matrix Norms

$A: m \times n \; matrix$

$\mathbf{1-norm}$: $\|A\|_1 = \max_j \sum_{i=1}^{m} |a_{ij}|$ Maximum absolute column sum

$\boldsymbol{\infty - norm}$: $\|\boldsymbol{A}\|_{\mathbf{1}} = \mathbf{\max_{\boldsymbol{i}}} \sum_{\boldsymbol{j}=1}^{\boldsymbol{n}} |\boldsymbol{a_{ij}}|$ Maximum absolute row sum

2-norm for a matrix is not easy to compute. Need to compute the largest singular value of the matrix

Example: $\boldsymbol{A} = \begin{bmatrix} 1 & 3 \\ -5 & 8 \end{bmatrix}$

$$\|\boldsymbol{A}\|_1 = \max(1+5, 3+8) = 11$$
$$\|\boldsymbol{A}\|_\infty = \max(1+3, 5+8) = 13$$

Properties of Matrix Norms

For any $n \times n$ matrix $\mathbf{A}$ and induced matrix norm,

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|, \qquad \textit{for any } \mathbf{x} \in \mathbb{R}^n,$$

$$\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|, \qquad \textit{for any } \mathbf{B} \in \mathbb{R}^{n \times n},$$

$$\|\mathbf{A}^k\| \leq \|\mathbf{A}\|^k, \qquad \textit{for any integer } k \geq 0.$$

$$\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$$

$\|\gamma \mathbf{A}\| \leq |\gamma| \cdot \|\mathbf{A}\|$ for any scalar γ

Matrix Condition Number

- Consider a nonsingular square matrix **A**
- Condition number with respect to a given norm
    - $cond(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$
- Condition number would be infinity if A is singular
- Condition number bounds the ratio of the relative change in the solution of a linear system to a given relative change in the input data
- Condition number is a measure of how close a matrix is to being a singular matrix
- Computing the inverse of a matrix is more expensive than solving a linear system governed by the same matrix
    - Condition number is estimated within an order of magnitude in practice!

Matrix condition number

- We want to solve $\boldsymbol{Ax} = \boldsymbol{b}$
- We want to analyze how robust our answers are to perturbations of $\boldsymbol{A}$ and $\boldsymbol{b}$
- To measure what happens to the sizes of vectors and matrices, use norms
- We know that $\left|\left|\boldsymbol{b}\right|\right| = ||\boldsymbol{Ax}|| \leq ||\boldsymbol{A}||\ ||\boldsymbol{x}||$
- We also know that $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$,
  - $||\boldsymbol{x}|| = ||\boldsymbol{A}^{-1}\boldsymbol{b}|| \leq \left|\left|\boldsymbol{A}^{-1}\right|\right|\ ||\boldsymbol{b}||$
- If we change $\boldsymbol{b}$ to $\boldsymbol{b} + \boldsymbol{d}$, then the solution changes somewhat to $\boldsymbol{x} + \boldsymbol{h}$, say, such that $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{h}) = \boldsymbol{b} + \boldsymbol{d}$
- But, using the original equation $\boldsymbol{Ax} = \boldsymbol{b}$, the first term on each side cancels

Matrix condition number

- We then have $\boldsymbol{Ah} = \boldsymbol{d}$
- Then $\boldsymbol{h} = \boldsymbol{A}^{-1}\boldsymbol{d}$
- Compute norms and use property of norms
  - $||\boldsymbol{h}|| = \|\boldsymbol{A}^{-1}\boldsymbol{d}\| \leq \|\boldsymbol{A}^{-1}\| \cdot ||\boldsymbol{d}||$
- Let's now look at the relative change in the solution $||\boldsymbol{h}||/||\boldsymbol{x}||$ compared to the relative change in the right-hand side $||\boldsymbol{d}||/||\boldsymbol{b}||$
- Taking the ratio of those two

$$\frac{\frac{\|\mathbf{h}\|}{\|\mathbf{x}\|}}{\frac{\|\mathbf{d}\|}{\|\mathbf{b}\|}} = \frac{\|\mathbf{h}\| \cdot \|\mathbf{b}\|}{\|\mathbf{d}\| \cdot \|\mathbf{x}\|} \leq \frac{\left(\|\mathbf{A}^{-1}\|\,\|\mathbf{d}\|\right)\left(\|\mathbf{A}\|\,\|\mathbf{x}\|\right)}{\|\mathbf{d}\|\,\|\mathbf{x}\|} = \|\mathbf{A}^{-1}\|\,\|\mathbf{A}\|$$

- The last quantity is the condition number

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

Choose d such that $\frac{\|\boldsymbol{h}\|}{||\boldsymbol{d}||}$ is as large as possible to have a reasonable estimate of $||\boldsymbol{A}^{-1}||$

Estimating the condition number

$\text{cond}(\boldsymbol{A}) = \left|\left|\boldsymbol{A}^{-\mathbf{1}}\right|\right| \cdot ||\boldsymbol{A}\,||$

- There are different approaches to compute the norm $||\cdot||$
  - 2-norm is common (use SVD)
- Condition number of a matrix is the ratio of the largest singular value of that matrix to the smallest singular value.
  - $\text{cond}(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}}$
- 2-norm might not be feasible for large matrices
  - Use an estimate of the condition number

Triangular Systems

- Let's begin with simple systems to understand algorithms for solving $\boldsymbol{Ax} = \boldsymbol{b}$
- Consider *lower triangular* systems where elements above a_{ii} are zero
- Said another way, $a_{ij} = 0, j > i$
- Solve by starting with $x_1 = \frac{8}{4} = 2$
- Continue with $x_2 = \frac{5-(3)(2)}{-1} = 1$
- Then find x_3, then x_4, to get solution
- This is called *forward substitution*

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 3 & -1 & 0 & 0 \\ -1 & 0 & 3 & 0 \\ 1 & -1 & -1 & 2 \end{bmatrix} \mathrm{x} = \begin{bmatrix} 8 \\ 5 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathrm{x} = \begin{bmatrix} 2 \\ 1 \\ 2/3 \\ 1/3 \end{bmatrix}$$

Triangular Systems

- For the more general system $\boldsymbol{Lx} = \boldsymbol{b}$ where $\boldsymbol{L}$ is lower triangular, we can still do forward substitution
- L_{ij} are elements of $\boldsymbol{L}$
- No trouble here unless $L_{ii} = 0$, then we can divide by it
- Theorem 2.1: If at least one of the $L_{ii} = 0$, then $\boldsymbol{L}$ is singular
- How to implement in MATLAB?

$$
\begin{aligned}
x_1 &= \frac{b_1}{L_{11}} \\
x_2 &= \frac{b_2 - L_{21}x_1}{L_{22}} \\
x_3 &= \frac{b_3 - L_{31}x_1 - L_{32}x_2}{L_{33}} \\
x_4 &= \frac{b_4 - L_{41}x_1 - L_{42}x_2 - L_{43}x_3}{L_{44}}.
\end{aligned}
$$

Upper Triangular Systems

- For the upper triangular system $\boldsymbol{Ux} = \boldsymbol{b}$ where $\boldsymbol{U}$ has $U_{ij} = 0, j < i$
- That is elements below the main diagonal are zero this time
- Solve for last variable first this time and work backward
- No trouble here unless $U_{ii} = 0$, then we cannot divide by it
- Theorem 2.1: If at least one of the $U_{ii} = 0$, then $\boldsymbol{U}$ is singular

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} & U_{14} \\ 0 & U_{22} & U_{23} & U_{24} \\ 0 & 0 & U_{33} & U_{34} \\ 0 & 0 & 0 & U_{44} \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$x_4 = \frac{b_4}{U_{44}}$$

$$x_3 = \frac{b_3 - U_{34}x_4}{U_{33}}$$

$$x_2 = \frac{b_2 - U_{23}x_3 - U_{24}x_4}{U_{22}}$$

$$x_1 = \frac{b_1 - U_{12}x_2 - U_{13}x_3 - U_{14}x_4}{U_{11}}$$

Gauss Elimination Method

- Gauss Elimination is a direct method to solve linear systems of equations.
- It converts the system into an upper triangular matrix.
- Solutions are then obtained using back substitution.
- Computational complexity is $\mathcal{O}(n^3)$
- Fast iterative solvers exist
  - Commonly used in practice when *n* is large
- Avoid solving a linear system by inverting A
  - $x = A^{-1}b$
  - Inverting A is costlier than G.E. and also less accurate!

Pivoting in Gauss Elimination

- Gauss elimination is often implemented with pivoting
  - Leading diagonal entry of the unreduced portion of the matrix can be zero
- Solution: Interchange rows
  - Does not alter the solution!
- Pivoting avoids division by zero or very small numbers.
- Pivoting improves numerical stability.
- Two types: Partial Pivoting and Complete Pivoting.
  - Partial Pivoting: Swap rows to get the largest element in the pivot position.
    - Commonly adopted
    - Numerical stability is more than adequate in practice
  - Complete Pivoting: Swap both rows and columns.
    - Superior but much more expensive in terms of pivot search

Steps in Gauss Elimination with Partial Pivoting

1. Identify the pivot element
   - Choose the largest absolute value in the column
   - Rounding errors will not be amplified
   - Multipliers will never exceed 1 in magnitude
   - Leads to a numerically stable implementation of G.E. method
2. Swap rows to bring the pivot element to the diagonal.
3. Eliminate lower elements in the column.
4. Repeat for the next column.
5. Perform back substitution to find solutions.

Permitted Transformations

- Linear system, it can be written as matrix product:
  - **T1**: Multiplication of row/column with scalar
  - **T2**: Interchanging rows/columns
  - **T3**: Add/subtract scalar multiple of current row/column to another row/column, replace current row/column

Pivot coefficient

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

T3 on rows 1 & 2 $\Longrightarrow$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$m_{21} = a_{21} / a_{11}$$

$$\begin{array}{rl} & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \\ - & m_{21}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) = m_{21}b_1 \end{array}$$

$$0 + \underbrace{(a_{22} - m_{21}a_{12})}_{a'_{22}}x_2 + \underbrace{(a_{23} - m_{21}a_{13})}_{a'_{23}}x_3 + \underbrace{(a_{24} - m_{21}a_{14})}_{a'_{24}}x_4 = \underbrace{b_2 - m_{21}b_1}_{b'_2}$$

T3 on rows 1 & 3 $\Rightarrow$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b_4 \end{bmatrix}$$

T3 on rows 1 & 4 $\Rightarrow$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Now continue with a_{22}' as pivot to eliminate second column of rows 3 & 4

T3 on rows 2 & 3 rows 2 & 4 $\Rightarrow$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b''_4 \end{bmatrix}$$

T3 on rows 3 & 4 $\Rightarrow$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$$

Transformed equation into $Ux=b'''$, U is upper triangular

Complexity analysis for $n \times n$ matrix A

- In total *(n-1)+(n-2)+…+1=n(n-1)/2* **T3** transformations (Adding multiple of a lower row to the pivot row)
  - Each T3 transformation requires $\mathcal{O}(n)$ multiplications & additions

- Total $\mathcal{O}(n^3)$ arithmetic operations for Gauss Elimination to solve for **x**
  - Backsubstitution only needs $\mathcal{O}(n^2)$ operations
  - Close to optimal for **<u>general</u>** matrices
  - Special matrices (very sparse) can be solved faster
    - e.g. Thomas algorithm for tri-diagonal systems

Example

Solve the system of equations:

$$2x + 3y + z = 1$$

$$4x + y - 2z = -2$$

$$-2x + 2y + z = 7$$

using Gauss Elimination with partial pivoting.

Step 1: Form the Augmented Matrix

The augmented matrix is:

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 1 \\ 4 & 1 & -2 & -2 \\ -2 & 2 & 1 & 7 \end{array}\right]$$

Step 2: Partial Pivoting and Row Swap

[2 3 1 | 1]
[4 1 -2 | -2]
[-2 2 1 | 7]

In column 1, the largest absolute value is 4 (Row 2).

Swap Row 1 and Row 2 to make it the pivot row.

New matrix:

[4 1 -2 | -2]
[2 3 1 | 1]
[-2 2 1 | 7]

Step 3: Eliminate Elements Below Pivot

Eliminate elements below the pivot in column 1.

R2 = R2 - (1/2) * R1

R3 = R3 + (1/2) * R1

New matrix:

[4 1 -2 | -2]

[0 2 2 | 2]

[0 2 0 | 6]

Step 4: Continue to Upper Triangular Form

[4 1 -2 | -2]
[0 2 2 | 2]
[0 2 0 | 6]

- Pivot on column 2. Largest is in Row 2 (no swap needed).
- Eliminate below the pivot: R3 = R3 - R2

New matrix:

[4 1 -2 | -2]
[0 2 2 | 2]
[0 0 -2 | 4]

Step 5: Back Substitution

Start from the last row:

- Row 3: $-2z = 4 \rightarrow z = -2$
- Row 2: $2y + 2(-2) = 2 \rightarrow y = 3$
- Row 1: $4x + 1(3) - 2(-2) = -2 \rightarrow x = -1$

Solution:

$x = -1,$

$y = 3,$

$z = -2.$

LU factorization

- Those multipliers that we used to create zero in the columns can go right into the lower triangular matrix L!

$$L = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix}$$

- And, finally, we have

$$LU = A$$

- This shows that A can factored or factorized into a lower triangular L and upper triangular U

LU Factorization

- Given $\mathbf{Ax} = \mathbf{b}$
- Factor $\mathbf{LU} = \mathbf{A}$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}.$$

L: Lower triangular matrix

U: upper triangular matrix

- Observe $(\mathbf{LU})\mathbf{x} = \mathbf{b} \rightarrow \mathbf{L}(\mathbf{Ux}) = \mathbf{b} \rightarrow \mathbf{Ux} = \mathbf{z} \rightarrow \mathbf{Lz} = \mathbf{b}$
- Solve $\mathbf{Lz} = \mathbf{b}$ for $\mathbf{z}$ by forward substitution
- Solve $\mathbf{Ux} = \mathbf{z}$ for x by backward substution

Note that this algorithm is NOT stable

Prone to division by zero because there is no row swapping

How to Exploit LU Factorization

- Consider $\mathbf{Ax} = \mathbf{b}$
- If **A** is not changing, it can be factored into L and U triangular matrices once and stored!
- Complexity of Gauss elimination is $\mathcal{O}(n^3)$
  - Factorization step is $\mathcal{O}(n^3/3)$
    - Can be done once if A is not changing
  - Subsequent triangular solutions is $\mathcal{O}(n^2)$
    - Substantial savings for large problems.

Gauss-Jordan Elimination

- More expensive than Gauss Elimination
- Motivated by the fact that diagonal linear systems are easier to solve
- Variation of the G.E. method such that the matrix A is transformed into a diagonal form rather than a triangular form
- The last stage is amenable to parallelization
    - No back substitution, which is serial