Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- **Linear Equation Solver**
  - Gaussian elimination
  - Condition number
  - Full/partial pivoting

Linear Equation

- **Ordinary differential equation**

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad x(0) = 0$$

Backward Euler

$$x(t_{n+1}) = (I - \Delta t \cdot A)^{-1} \cdot [x(t_n) + \Delta t \cdot B \cdot u(t_{n+1})] \quad x(t_0) = 0$$

- **Partial differential equation**

$$\rho \cdot C_p \cdot \frac{\partial T(x,y,z,t)}{\partial t} = \kappa \cdot \nabla^2 T(x,y,z,t) + f(x,y,z,t)$$

Finite Difference

$$\rho \cdot C_p \cdot \frac{\partial T_{i,j,k}}{\partial t} = f_{i,j,k} + \frac{\kappa \cdot [T_{i+1,j,k} - T_{i,j,k}]}{(\Delta x)^2} - \frac{\kappa \cdot [T_{i,j,k} - T_{i-1,j,k}]}{(\Delta x)^2} +$$

$$\frac{\kappa \cdot [T_{i,j+1,k} - T_{i,j,k}]}{(\Delta y)^2} - \frac{\kappa \cdot [T_{i,j,k} - T_{i,j-1,k}]}{(\Delta y)^2} + \frac{\kappa \cdot [T_{i,j,k+1} - T_{i,j,k}]}{(\Delta z)^2} - \frac{\kappa \cdot [T_{i,j,k} - T_{i,j,k-1}]}{(\Delta z)^2}$$

Linear Equation Solver

$$A \cdot X = B$$

- **In theory, X is equal to $A^{-1}B$**
- **In practice, explicitly inverting a matrix is never a good idea**
- **A more efficient algorithm should be applied**
  - E.g., use X = A\B in MATLAB

Gaussian Elimination

- **Step 1: convert A to an upper triangular matrix**

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \\ B \\ \\ \end{bmatrix} \Rightarrow \begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \\ Y \\ \\ \end{bmatrix}$$

- **Step 2: solve for X via backward substitution**

$$\begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \\ Y \\ \\ \end{bmatrix} \Rightarrow \begin{bmatrix} \\ X \\ \\ \end{bmatrix}$$

Gaussian Elimination

- **A simple example**

$$\underbrace{\begin{bmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} 8 \\ -11 \\ -3 \end{bmatrix}}_{\mathbf{B}}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 1/2 & 1/2 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 5 \end{bmatrix}$$

Gaussian Elimination

- **Step 1: convert A to an upper triangular matrix**

$$\begin{bmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -11 \\ -3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 0.5 & 0.5 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 5 \end{bmatrix}$$

Gaussian Elimination

- **Step 1: convert A to an upper triangular matrix**

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 0.5 & 0.5 \\ 0 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 1 \end{bmatrix}$$

Gaussian Elimination

- Step 2: solve for X via backward substitution

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 1 \end{bmatrix}$$

$-x_3 = 1$ $x_3 = -1$

$0.5x_2 + 0.5x_3 = 1$ $0.5x_2 - 0.5 = 1$ $x_2 = 3$

$2x_1 + x_2 - x_3 = 8$ $2x_1 + 3 - (-1) = 8$ $2x_1 + 4 = 8$

$x_1 = 2$

Gaussian Elimination

- Gaussian elimination is much cheaper than calculating A^{-1}

$$\begin{bmatrix} A \end{bmatrix} \cdot \begin{bmatrix} X \end{bmatrix} = \begin{bmatrix} B \end{bmatrix}$$

Gaussian elimination: solve for X where B is an N x 1 vector

$$\begin{bmatrix} A \end{bmatrix} \cdot \begin{bmatrix} A^{-1} \end{bmatrix} = \begin{bmatrix} I \end{bmatrix}$$

Matrix inverse: solve for A^{-1} where I is an N x N identity matrix

The difference between Gaussian elimination and matrix inverse becomes more significant for large matrix

Numerical Noise

- **In theory, Gaussian elimination works well if A is nonsingular, i.e.,**

$$A \cdot X = B \quad where \quad \det(A) \neq 0$$

  - A is singular if and only if det(A) = 0

- **However, round-off errors in our numerical computation can bring about problems even if det(A) is not 0**
  - Numerical noise can change the determinant value for Gaussian elimination

Numerical Noise

- **A simple example**

$$A = \begin{bmatrix} 100 & -100 \\ -100 & 100.01 \end{bmatrix}$$

$\det(A) = 100 \cdot 100.01 - 100 \cdot 100$
$= 10001 - 10000$
$= 1 \neq 0$

  - If our machine only has 3 decimal digits of precision

$$A \approx \begin{bmatrix} 100 & -100 \\ -100 & 100 \end{bmatrix}$$

$\det(A) = 100 \cdot 100 - 100 \cdot 100$
$= 0$

Condition Number

- **The "singularity" of a linear equation can be quantitatively measured by its condition number**

$$A \cdot X = B$$

- **The condition number of A is defined as:**

$$k(A) = \|A\| \cdot \|A^{-1}\|$$

  - ||•|| is the norm of a matrix

Condition Number

- **We can get different condition number values when using different matrix norm definitions**

1-norm $$\|A\|_1 = \max_{1 \le j \le N} \sum_{i=1}^{N} |a_{ij}|$$

F-norm $$\|A\|_F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} |a_{ij}|^2}$$

Inf-norm $$\|A\|_\infty = \max_{1 \le i \le N} \sum_{j=1}^{N} |a_{ij}|$$

max

max

Condition Number

- **Condition number is highly correlated to singularity**
  - Use 1-norm as an example

1-norm $$\|A\|_1 = \max_{1\le j\le N} \sum_{i=1}^{N} |a_{ij}|$$

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

$A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad K(A) = 1 \cdot 1 = 1$

$\max(1, 1) = 1$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-5} \end{bmatrix}$$

$A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 10^5 \end{bmatrix}$ $\quad K(A) = 1 \cdot 10^5 = 10^5$

$\max(1, 10^5) = 10^5$

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

$A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \infty \end{bmatrix}$ $\quad K(A) = 1 \cdot \infty = \infty$

$\max = (1, \infty) = \infty$

Condition Number

- **For the equation AX = B, the solution error is bounded by:**

$$\frac{\|\Delta X\|}{\|X\|} \le k(A) \cdot \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta B\|}{\|B\|} \right)$$ ← Condition number important

  - ΔA and ΔB: errors of A and B respectively
  - ΔX: errors of the solution X

- **Large condition number yields large solution error**
  - E.g., MATLAB will show a warning message if k(A) is more than 10^{16}~10^{17}

Simple Examples

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot X = B \qquad k\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 1$$

$B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \longrightarrow x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\Delta B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$ $\quad \Delta X = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$

$B = \begin{bmatrix} 1 \\ 1.1 \end{bmatrix}$ $\quad X = \begin{bmatrix} 1 \\ 1.1 \end{bmatrix}$

Simple Examples

$$\begin{bmatrix} 1 & 1 \\ 0.999 & 1 \end{bmatrix} \cdot X = B \qquad k\left(\begin{bmatrix} 1 & 1 \\ 0.999 & 1 \end{bmatrix}\right) = 4000$$

$$B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow X = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\Delta B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \qquad \Delta X = \begin{bmatrix} -100 \\ 100 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 1.1 \end{bmatrix} \Rightarrow X = \begin{bmatrix} -100 \\ 101 \end{bmatrix}$$

Pivoting for Accuracy

$$\frac{\|\Delta X\|}{\|X\|} \le k(A)\cdot\left(\frac{\|\Delta A\|}{\|A\|}+\frac{\|\Delta B\|}{\|B\|}\right)$$

- **This inequality only considers numerical errors in A and B**
  - It assumes that no additional error is introduced when solving the equation (e.g., during Gaussian elimination)
- **Gaussian elimination adds extra numerical errors**
  - Every intermediate step is not perfect (due to rounding)

Pivoting for Accuracy

- **When solving AX = B, we should minimize the additional numerical error introduced by the solver**
- **A general rule is to select large pivot values during Gaussian elimination**

Pivoting for Accuracy

- **Example: solve the following problem on a machine that has 3 decimal digits of precision**

$$\begin{bmatrix} 1.00e-4 & 1.00 \\ 1.00 & 1.00 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.00 \\ 2.00 \end{bmatrix}$$

- **If we directly apply Gaussian elimination w/o pivoting:**

$$\begin{bmatrix} 1e^{-4} & 1.00 \\ 0.00 & -1e^{-4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.00 \\ -1.00e^{4} \end{bmatrix}$$

$-1.00e^{4} \cdot x_2 = -1.00e4$ $\qquad x_2 = 1.00$

$1.00e^{-4} \cdot x_1 + x_2 = 1.00$ $\qquad x_1 = 0.00$

$x_1 + x_2 = 1.00$

Pivoting for Accuracy

- **If we apply Gaussian elimination w/ pivoting:**

$$\begin{bmatrix} 1.00e-4 & 1.00 \\ 1.00 & 1.00 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.00 \\ 2.00 \end{bmatrix}$$

$$\begin{bmatrix} 1.00 & 1.00 \\ 1.00e^{-4} & 1.00 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 1.00 \end{bmatrix}$$

$$\begin{bmatrix} 1.00 & 1.00 \\ 0.00 & 1.00 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 1.00 \end{bmatrix}$$

$x_2 = 1.00$

$x_1 + x_2 = 2.00$ $\qquad x_1 = 1.00$

$$\begin{cases} 1.00e^{-4} \cdot 1.00 + 1.00 \cdot 1.00 = 1.00 \\ 1.00 \cdot 1.00 + 1.00 \cdot 1.00 = 2.00 \end{cases}$$

Pivoting for Accuracy

- **Various choices of pivoting (tradeoff between accuracy and runtime)**
  - Full: Swap rows and columns to get largest magnitude on the diagonal
  - Partial: Swap to put largest magnitude from pivot row (or column) onto diagonal

Summary

- **Linear equation solver**
  - Gaussian elimination
  - Condition number
  - Full/partial pivoting

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- Linear Equation Solver
  - LU decomposition
  - Cholesky decomposition

Linear Equation Solver

- **Gaussian elimination solves a linear equation**

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \\ B \\ \\ \end{bmatrix}$$

- **Sometimes we want to repeatedly calculate the solutions for different right-hand-side vectors**

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \\ X_1 \\ \\ \end{bmatrix} = \begin{bmatrix} \\ B_1 \\ \\ \end{bmatrix} \qquad \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \\ X_2 \\ \\ \end{bmatrix} = \begin{bmatrix} \\ B_2 \\ \\ \end{bmatrix}$$

Case 1 **Case 2**

Ordinary Differential Equation Example

- **Backward Euler integration for linear ordinary differential equation with constant time step Δt**

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad x(0) = 0$$

$$\Downarrow$$

$$x(t_{n+1}) = \underbrace{(I - \Delta t \cdot A)^{-1}}_{\text{Identical @ all } t_n\text{'s}} \cdot \underbrace{[x(t_n) + \Delta t \cdot B \cdot u(t_{n+1})]}_{\text{Different @ all } t_n\text{'s}} \quad x(t_0) = 0$$

LU Factorization

- **It would be expensive to repeatedly run Gaussian elimination for many times**
  - How can we save and re-use the intermediate results?
  - LU factorization is to address this problem

$$\begin{bmatrix} & A & \end{bmatrix} \cdot \begin{bmatrix} X_1 \end{bmatrix} = \begin{bmatrix} B_1 \end{bmatrix} \qquad \begin{bmatrix} & A & \end{bmatrix} \cdot \begin{bmatrix} X_2 \end{bmatrix} = \begin{bmatrix} B_2 \end{bmatrix}$$

Case 1 **Case 2**

LU Factorization

- **Key idea:**
  - Represent A as the product of L (lower triangular) and U (upper triangular) via Gaussian-elimination-like steps
  - All diagonal elements in U are set to 1 by proper scaling

$$\begin{bmatrix} & A & \end{bmatrix} \cdot \begin{bmatrix} X \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} \qquad \begin{bmatrix} & A & \end{bmatrix} = \begin{bmatrix} & L & \end{bmatrix} \cdot \begin{bmatrix} & U & \end{bmatrix}$$

LU factorization is unchanged as long as A is unchanged (i.e., independent of the right-hand-side vector B)

LU Factorization

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{N1} & l_{N2} & \cdots & l_{NN} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \cdots & u_{1N} \\ & 1 & \cdots & u_{2N} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

$a_{11} = l_{11} \cdot 1$

$a_{21} = l_{21} \cdot 1 + l_{22} \cdot 0 = l_{21}$

$A_{N1} = l_{N1} \cdot 1 + l_{N2} \cdot 0 + \ldots 0 = l_{N1}$

LU Factorization

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{N1} & l_{N2} & \cdots & l_{NN} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & \cdots & u_{1N} \\ & 1 & \cdots & u_{2N} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

$a_{12} = l_{11} u_{12} + 0 \ldots = l_{11} u_{12}$

$a_{13} = l_{11} \cdot u_{13}$

$A_{1N} = l_{11} \cdot u_{1N}$

LU Factorization

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{N1} & l_{N2} & \cdots & l_{NN} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & \cdots & u_{1N} \\ & 1 & \cdots & u_{2N} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

$a_{22} = l_{21} u_{12} + l_{22}$

$\vdots$

$a_{N2} = l_{N1} u_{12} + l_{N2}$

Memory Storage

- **The matrix A can be iteratively replaced by L and U**
  - No additional memory is required

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} l_{11} & u_{12} & \cdots & u_{1N} \\ l_{21} & l_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & u_{N-1,N} \\ l_{N1} & l_{N2} & \cdots & l_{NN} \end{bmatrix}$$

A Simple LU Example

$$\begin{bmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ & 1 & u_{23} \\ & & 1 \end{bmatrix}$$

$2 = l_{11} \cdot 1$

$-3 = l_{21} \cdot 1$

$-2 = l_{31} \cdot 1$

$$\begin{bmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{bmatrix}$$

A Simple LU Example

$$\begin{bmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ & 1 & u_{23} \\ & & 1 \end{bmatrix}$$

$1 = l_{11} \cdot u_{12} = 2 \cdot u_{12} \qquad u_{12} = \frac{1}{2}$

$-1 = l_{11} \, u_{13} = 2 \cdot u_{13} \qquad u_{13} = \frac{-1}{2}$

$$\begin{bmatrix} 2 & \frac{1}{2} & \frac{-1}{2} \\ -3 & -1 & 2 \\ 2 & 1 & 2 \end{bmatrix}$$

A Simple LU Example

$$\begin{bmatrix} 2 & 1/2 & -1/2 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ & 1 & u_{23} \\ & & 1 \end{bmatrix}$$

$-1 = l_{21} u_{12} + l_{22} = (-3) \cdot \frac{1}{2} + l_{22} = \frac{-3}{2} + l_{22}$

$l_{22} = \frac{1}{2}$

$1 = l_{31} u_{12} + l_{32} = (-2) \cdot \frac{1}{2} + l_{32} = -1 + l_{32}$

$l_{32} = 2$

$$\begin{bmatrix} 2 & \frac{1}{2} & \frac{-1}{2} \\ -3 & \frac{1}{2} & 2 \\ -2 & 2 & 2 \end{bmatrix}$$

A Simple LU Example

$$\begin{bmatrix} 2 & 1/2 & -1/2 \\ -3 & 1/2 & 2 \\ -2 & 2 & 2 \end{bmatrix} \quad \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ & 1 & u_{23} \\ & & 1 \end{bmatrix}$$

$2 = l_{21} u_{13} + l_{22} u_{23} = (-3) \cdot \left(\frac{-1}{2}\right) + \frac{1}{2} u_{23}$

$= \frac{3}{2} + \frac{1}{2} u_{23} \qquad u_{23} = 1$

$$\begin{bmatrix} 2 & \frac{1}{2} & \frac{-1}{2} \\ -3 & \frac{1}{2} & 1 \\ -2 & 2 & 2 \end{bmatrix}$$

A Simple LU Example

$$\begin{bmatrix} 2 & 1/2 & -1/2 \\ -3 & 1/2 & 1 \\ -2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ & 1 & u_{23} \\ & & 1 \end{bmatrix}$$

$2 = l_{31} U_{13} + l_{32} U_{23} + l_{33} = (-2)(-\frac{1}{2}) + 2 \cdot 1 + l_{33}$

$= 1 + 2 + l_{33} \qquad l_{33} = -1$

$$\begin{bmatrix} 2 & \frac{1}{2} & -\frac{1}{2} \\ -3 & \frac{1}{2} & 1 \\ -2 & 2 & -1 \end{bmatrix}$$

LU Factorization

- Given L and U, solve linear equation via two steps

$$A \cdot X = B$$

$A = L \cdot U$

$L \cdot \underbrace{U \cdot X}_{V} = B$

$L \cdot V = B$

$UX = V$

LU Factorization

$$[L] \cdot [V] = [B] \quad \Rightarrow \quad [U] \cdot [X] = [V]$$

Forward substitution   **Backward substitution**

- **Only the above two steps are repeated if the right-hand-side vector B is changed**
  - LU factorization is not repeated
  - More efficient than Gaussian elimination

Cholesky Factorization

- **If the matrix A is symmetric and positive definite, Cholesky factorization is preferred over LU factorization**

$$[A] = [L] \cdot [L^T]$$

- **Cholesky factorization is cheaper than LU**
  - Only needs to find a single triangular matrix L (instead of two different matrices L and U)

Cholesky Factorization

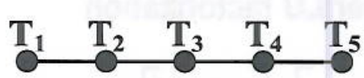
- A must be symmetric and positive definite to make Cholesky factorization applicable → $A^T = A$
- A symmetric matrix A is positive definite if

$$P^T \cdot A \cdot P > 0 \quad \text{for any real-valued vector } P \neq 0$$

- Sufficient and necessary condition for a symmetric matrix A to be positive definite:
  - All eigenvalues of A are positive

Partial Differential Equation Example

- 1-D rod discretized into 4 segments



$$\kappa \cdot \frac{\partial^2 T(x,t)}{\partial x^2} = 0$$

$$T_1 = 30 \quad T_5 = 100$$

$$\kappa \cdot (-T_{i-1} + 2T_i - T_{i+1}) = 0 \quad (2 \le i \le 4)$$

$$-30 + 2T_2 - T_3 = 0$$
$$-T_2 + 2T_3 - T_4 = 0$$
$$-T_3 + 2T_4 - 100 = 0$$

Partial Differential Equation Example

$$-30+2T_2-T_3=0$$
$$-T_2+2T_3-T_4=0$$
$$-T_3+2T_4-100=0$$

$$\Rightarrow \underbrace{\begin{bmatrix} 2 & -1 & \\ -1 & 2 & -1 \\ & -1 & 2 \end{bmatrix}}_{A} \cdot \begin{bmatrix} T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 30 \\ 0 \\ 100 \end{bmatrix}$$

- **Eigenvalues of A**

$$\lambda_1 = 3.41$$
$$\lambda_2 = 2.00 \quad \text{(A is positive definite)}$$
$$\lambda_3 = 0.58$$

Partial Differential Equation Example

- **In practice, we never calculate eigenvalues to check if a matrix is positive definite or not**
  - Eigenvalue decomposition is much more expensive than solving a linear equation

- **If we apply finite difference to discretize steady-state heat equation, the resulting linear equation is positive definite**

Partial Differential Equation Example

$$\begin{bmatrix} 2 & -1 & \\ -1 & 2 & -1 \\ & -1 & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & l_{33} \end{bmatrix}$$

$A = LL^T$
$= (-L)\cdot(-L)^T$

$2 = l_{11} \cdot l_{11}$ $\quad l_{11} = \sqrt{2}$

$-1 = l_{21} \cdot l_{11} = l_{21} \cdot \sqrt{2}$ $\quad l_{21} = -\frac{1}{\sqrt{2}} = -\sqrt{\frac{1}{2}}$

$0 = l_{31}\, l_{11} = l_{31} \cdot \sqrt{2}$ $\quad l_{31} = 0$

$$\begin{bmatrix} \sqrt{2} & -1 & \\ -\sqrt{\frac{1}{2}} & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Partial Differential Equation Example

$$\begin{bmatrix} \sqrt{2} & -1 & \\ -\sqrt{1/2} & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \quad \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & l_{33} \end{bmatrix}$$

$2 = l_{21}^2 + l_{22}^2 = \frac{1}{2} + l_{22}^2$ $\quad l_{22} = \sqrt{\frac{3}{2}}$

$-1 = l_{31}\, l_{21} + l_{32}\, l_{22} = 0 \cdot \left(-\sqrt{\tfrac{1}{2}}\right) + l_{32} \cdot \sqrt{\tfrac{3}{2}}$

$l_{32} = -\sqrt{\frac{2}{3}}$

$$\begin{bmatrix} \sqrt{2} & -1 & \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{3}{2}} & -1 \\ 0 & -\sqrt{\frac{2}{3}} & 2 \end{bmatrix}$$

Partial Differential Equation Example

$$\begin{bmatrix} \sqrt{2} & -1 & \\ -\sqrt{1/2} & \sqrt{3/2} & -1 \\ 0 & -\sqrt{2/3} & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & l_{33} \end{bmatrix}$$

$2 = l_{31}^2 + l_{32}^2 + l_{33}^2 = 0 + \frac{2}{3} + l_{33}^2 \qquad l_{33} = \sqrt{\frac{4}{3}}$

$$\begin{bmatrix} \sqrt{2} & -1 & \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{3}{2}} & -1 \\ 0 & -\sqrt{2/3} & \sqrt{4/3} \end{bmatrix}$$

Summary

- **Linear equation solver**
  - LU decomposition
  - Cholesky decomposition