ME 704

Computational Methods in Thermal and Fluids Engineering

(Solution of Linear Equations-2)

Kannan Iyer
Kiyer@iitb.ac.in

Department of Mechanical Engineering
Indian Institute of Technology, Bombay

Review-I

- Began the Solution of Linear equations
  - The motivation was from several engineering applications
  - Understood the definitions of diagonal, tri-diagonal, upper triangular and lower triangular matrices
  - Understood the logic for solution of equations when the coefficient matrix is either diagonal, upper triangular or lower triangular

Review-II

- Began with the direct solution of linear equations
  - Studied Gauss Elimination
  - Understood that Pivoting improves accuracy.
  - Studied Gauss Jordan method and understood that it can be used to compute inverse

L-U Decomposition

- If the problem has to be repeated with several source vectors for the same coefficient vector, L-U decomposition is recommended

$$[A]=[L][U]$$

- Such a decomposition speeds up calculation
- In general two methods are available
  - Crout's Decomposition
  - Dolittle's Decomposition

16:37 5/22

Crout's Decomposition

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{array}{c|ccc} & 1 & 3 & 5 \\ \hline 2 & a_{11} & a_{12} & a_{13} \\ 4 & a_{21} & a_{22} & a_{23} \\ & a_{31} & a_{32} & a_{33} \end{array}$$

- ❑ $a_{11} = l_{11}$, $a_{21} = l_{21}$, $a_{31} = l_{31}$
- ❑ $a_{12} = l_{11} u_{12}$, $a_{13} = l_{11} u_{13}$
- ❑ $a_{22} = l_{21} u_{12} + l_{22}$, $a_{32} = l_{31} u_{12} + l_{32}$
- ❑ $a_{23} = l_{21} u_{13} + l_{22} u_{23}$
- ❑ $a_{33} = l_{31} u_{13} + l_{32} u_{23} + l_{33}$

16:37 6/22

Logic for Crout's Method

$l_{i1} = a_{i1}$, for $i = 1, n$

$u_{1j} = a_{1j} / l_{11}$, for $j = 2,n$

for $j = 2 , n-1$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \quad (\text{ for } i = j, n)$$

$$u_{ji} = \left(a_{ji} - \sum_{k=1}^{j-1} l_{jk} u_{ki} \right) / l_{jj} \quad (\text{ for } i = j+1, n)$$

$$l_{nn} = a_{nn} - \sum_{k=1}^{n} l_{nk} u_{kn}$$

16:37 7/22

Solution for Crout's Method

$$[A]\{x\} = \{b\} \Rightarrow [L][U]\{x\} = \{b\}$$

- ❑ Introducing $[U]\{x\} = \{d\}$ 1
- ❑ This implies $[L]\{d\} = \{b\}$ 2
- ❑ Since [L] and {b} are known, {d} can be found from Eq.(2)by forward sweep
- ❑ Once {d} is found out, {x} can be found from Eq. (1) by backward sweep

16:37 8/22

Comments on Crout's Method

- ❑ $M_{crout} = M_{Gauss}$
- ❑ But, back substitution $M = n^2 - n$
- ❑ Therefore for a large set one may save substantial effort (n^3 vs n^2)
- ❑ It is possible to store the coefficients of [L] and [U] in [A] itself as [A] is no longer required. This saves memory
- ❑ Other decompositions are similar

Comments on Crout's Method-2

- It is possible to store L and U in A itself and conserve memory and logic written accordingly

$$\begin{bmatrix} l_{11} & u_{21} & u_{13} \\ l_{21} & l_{22} & u_{23} \\ l_{31} & l_{32} & u_{33} \end{bmatrix}$$

- Indices have to be carefully addressed
- Since memory is cheap, this no longer may be required

Tridiagonal Matrix Solution

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{Bmatrix}$$

Gauss Elimination

$$\Rightarrow \begin{bmatrix} 1 & a^*_{12} & 0 & 0 \\ 0 & 1 & a^*_{23} & 0 \\ 0 & 0 & 1 & a^*_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} b^*_1 \\ b^*_2 \\ b^*_3 \\ b^*_4 \end{Bmatrix}$$

Logic

$$a^*_{12} = a_{12} / a_{11} \qquad b^*_1 = b_1 / a_{11}$$

Gauss operation for row 2 would imply

$$a'_{22} = a_{22} - a^*_{12} a_{21} \qquad b'_2 = b_2 - b^*_1 a_{21}$$

- To make the diagonal = 1, we need to divide the row by the RHS of a'_{22}

$$\Rightarrow a^*_{23} = \frac{a_{23} - 0}{a_{22} - a^*_{12} a_{21}} \qquad b^*_2 = \frac{b'_2}{a_{22} - a^*_{12} a_{21}}$$

Thomas Algorithm (TDMA)-I

$$a^*_{12} = a_{12} / a_{11} \qquad b^*_1 = b_1 / a_{11}$$

For I = 2 to N

$$\Rightarrow a^*_{i,i+1} = \frac{a_{i,i+1}}{a_{i,i} - a^*_{i-1,i} a_{i,i-1}}$$

Skip for I = N

$$b^*_i = \frac{b_i - b^*_{i-1} a_{i,i-1}}{a_{i,i} - a^*_{i-1,i} a_{i,i-1}}$$

12/15

16:37

Thomas Algorithm (TDMA)-II

13/22

Back Substitution

$$x_N = b_N^*$$

$$x_i = b_i^* - x_{i+1} a_{i,i+1}^*$$ For I = N-1, N-2, …,1

- It is possible to store A as (N,3) to conserve memory and logic written accordingly

$$a_{i,i-1} = a_{i,1} \quad a_{i,i} = a_{i,2} \quad a_{i,i+1} = a_{i,3}$$

16:37

Closing Remarks on Direct Solvers

14/22

- These methods are subject to error propagation
- The error propagation can be indicated by a term called condition number
- Ill conditioned systems are difficult to solve
- Several specialised methods exist
- Refer your book and advanced Linear Algebra Texts.
- This exposure is sufficient for most general problems

16:37

Iterative Methods-I

15/22

- For large systems, which are sparse Iterative methods are most widely used
- These naturally occur during the solution of ODE's and PDE's.
- These methods do not suffer from propagation of round-off errors
- The set of equations have to be diagonal dominant to obtain convergence
- This is generally a limitation but where they are used, it can be achieved by some techniques

16:37

16/22

Iterative methods-II

- Consider

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

$$\Rightarrow \begin{aligned} x_1 &= (b_1 - (a_{12}x_2 + a_{13}x_3))/a_{11} \\ x_2 &= (b_2 - (a_{21}x_1 + a_{23}x_3))/a_{22} \\ x_3 &= (b_3 - (a_{31}x_1 + a_{32}x_2))/a_{33} \end{aligned}$$

- One can start with a guess and iterate

Jacobi Iteration

$$x_1^N = \left(b_1 - \left(a_{12}x_2^{N-1} + a_{13}x_3^{N-1}\right)\right)/\,a_{11}$$
$$x_2^N = \left(b_2 - \left(a_{21}x_1^{N-1} + a_{23}x_3^{N-1}\right)\right)/\,a_{22}$$
$$x_3^N = \left(b_3 - \left(a_{31}x_1^{N-1} + a_{32}x_2^{N-1}\right)\right)/\,a_{33}$$

- By adding and subtracting x_i^{N-1} on both sides

$$x_1^N = x_1^{N-1} + \left(b_1 - \left(a_{11}x_1^{N-1} + a_{12}x_2^{N-1} + a_{13}x_3^{N-1}\right)\right)/\,a_{11}$$
$$x_2^N = x_2^{N-1} + \left(b_2 - \left(a_{21}x_1^{N-1} + a_{22}x_2^{N-1} + a_{23}x_3^{N-1}\right)\right)/\,a_{22}$$
$$x_3^N = x_3^{N-1} + \left(b_3 - \left(a_{31}x_1^{N-1} + a_{32}x_2^{N-1} + a_{33}x_3^{N-1}\right)\right)/\,a_{33}$$

Gauss-Siedel Iteration

- Here new values are used as soon as they are available

$$x_1^N = x_1^{N-1} + \left(b_1 - \left(a_{11}x_1^{N-1} + a_{12}x_2^{N-1} + a_{13}x_3^{N-1}\right)\right)/\,a_{11}$$
$$x_2^N = x_2^{N-1} + \left(b_2 - \left(a_{21}x_1^{N} + a_{22}x_2^{N-1} + a_{23}x_3^{N-1}\right)\right)/\,a_{22}$$
$$x_3^N = x_3^{N-1} + \left(b_3 - \left(a_{31}x_1^{N} + a_{32}x_2^{N} + a_{33}x_3^{N-1}\right)\right)/\,a_{33}$$

- Where Jacobi converges, Gauss-Siedel converges faster

Successive Over Relaxation (SOR)

- Sufficient Condition for convergence for both Jacobi and Gauss-Siedel Iterations is

$$|a_{ii}| \geq \sum_{j=1}^{N} |a_{ij}| \quad \text{for i = 1, N}$$

- Where the above criteria is satisfied it is possible to accelerate it further by introducing over-relaxation factor
- The value of the over-relaxation factor lies between 1-2. Optimum values are available for some specific form of the coefficient matrix. In general it should be found by trials

Logic for SOR

- For i = 1, N

$$x_i = x_i + \omega\left(b_i - \left(\sum_{j=1}^{N} a_{ij}x_j\right)\right)/\,a_{ii}$$

16:37

Norms of Vectors

- p norm of a vector is defined as

$$\|x\|_p = \left[\sum_{j=1}^{N} |x_j|^p \right]^{\frac{1}{p}}$$

$\|x\|_1 \Rightarrow$ Sum of absolute values of components

$\|x\|_\infty \Rightarrow$ Absolute value of the largest component

$\|x\|_2 \Rightarrow$ Euclidean Norm

16:37

Termination Criterion

$$\left\| x^{N+1} - x^N \right\|_\infty \le \varepsilon$$

$$\frac{\left\| x^{N+1} - x^N \right\|_\infty}{\left\| x^N \right\|_\infty} \le \varepsilon$$

$$\left\| r^N \right\|_\infty \le \varepsilon$$