第2章 直接刚度法

2.4 平衡方程组的求解

Solution of Equilibrium Equations

[Bathe]: Chapter 8

Solution of Equilibrium Equations

- Direct solution techniques $\boldsymbol{Ka} = \boldsymbol{R}$
  - Gauss elimination
  - LDL^T solution
  - Cholesky factorization
  - Frontal solution

Equations are solved using a number of steps and operations that are predetermined in an exact manner.

- Iterative solution methods
  - Gauss-Seidel Method
  - Conjugate Gradient Method with Preconditioning
    - The number of iterations required for convergence depends on *the condition number* of the matrix $\boldsymbol{K}$ and whether the *acceleration schemes* used are effective for the particular case considered
    - For large system, iterative methods can be much more effective.

Solution of Equilibrium Equations

LDL$^{\mathrm{T}}$ solution

$$\boldsymbol{K} = \boldsymbol{L}\boldsymbol{U} = \boldsymbol{L}\boldsymbol{D}\tilde{\boldsymbol{U}}$$

$$\boldsymbol{L} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & \ddots & & \\ \vdots & \vdots & \vdots & 1 & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \quad \boldsymbol{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & \ddots & \cdots & \vdots \\ & & & u_{n-1,n-1} & u_{n-1,n} \\ & & & & u_{nn} \end{bmatrix} \quad \boldsymbol{D} = \begin{bmatrix} u_{11} & & & \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{bmatrix}$$

$$\boldsymbol{K}^{\mathrm{T}} = \tilde{\boldsymbol{U}}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$$

If $\boldsymbol{K}$ is symmetric: $\tilde{\boldsymbol{U}} = \boldsymbol{L}^{\mathrm{T}}$ $\quad \boldsymbol{K} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$

? ?

$$\boldsymbol{L}^{\mathrm{T}} = \begin{bmatrix} 1 & l_{21} & l_{31} & \cdots & l_{n1} \\ & 1 & l_{32} & \cdots & l_{n2} \\ & & \ddots & \cdots & \vdots \\ & & & 1 & l_{n,n-1} \\ & & & & 1 \end{bmatrix}$$

$$\boldsymbol{K} = \boldsymbol{L}\boldsymbol{U} \qquad k_{11} = u_{11} \qquad k_{ij} = \sum_{r=1}^{i-1} l_{ir}u_{rj} + u_{ij} \quad (j = 2:n;\ i \le j)$$

$$\Downarrow$$

$$u_{ij} = k_{ij} - \sum_{r=1}^{i-1} l_{ir}u_{rj}$$

$$\boldsymbol{U} = \boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$$

$$u_{ij} = d_{ii}l_{ji}$$

$$l_{ji} = u_{ij} / d_{ii}$$

$$d_{11} = u_{11} = k_{11}$$

$$d_{jj} = u_{jj} = k_{jj} - \sum_{r=1}^{j-1} l_{jr}u_{rj}$$

Solution of Equilibrium Equations LDL$^{\mathrm{T}}$ solution

$$\boldsymbol{K}\boldsymbol{a}=\boldsymbol{R} \Longleftarrow \boldsymbol{K}=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$$

$$\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{a}=\boldsymbol{R}$$

$$\boldsymbol{L}\boldsymbol{V}=\boldsymbol{R} \Rightarrow \sum_{j=1}^{i-1} l_{ij}v_j + v_i = r_i$$

$$v_1 = r_1, \quad v_i = r_i - \sum_{j=1}^{i-1} l_{ij}v_j$$

$$\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{a}=\boldsymbol{V} \Rightarrow \boldsymbol{L}^{\mathrm{T}}\boldsymbol{a}=\boldsymbol{D}^{-1}\boldsymbol{V}=\bar{\boldsymbol{V}}$$

$$a_i + \sum_{j=i+1}^{n} l_{ji}a_j = \bar{v}_i$$

$$a_n = \bar{v}_n, \quad a_i = \bar{v}_i - \sum_{j=i+1}^{n} l_{ji}a_j$$

$$\boldsymbol{L}=\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & \ddots & & \\ \vdots & \vdots & \vdots & 1 & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$$

$$\boldsymbol{L}^{\mathrm{T}}=\begin{bmatrix} 1 & l_{21} & l_{31} & \cdots & l_{n1} \\ & 1 & l_{32} & \cdots & l_{n2} \\ & & \ddots & \cdots & \vdots \\ & & & 1 & l_{n,n-1} \\ & & & & 1 \end{bmatrix}$$

- Reduction of ***R*** can be performed at the same time as the ***K*** is decomposed or may be carried out separately afterward — **Multiple load cases**

- Computer implementation of the Gauss solution procedure
  - Use a small solution time
  - The high-speed storage requirements should be small
  - Possible for effective out-of-core solution
- *Active column solution* or *skyline (or column) reduction method*
  - LDL^T decomposition can be carried out by columns
  - Calculation of the element l_{ij} and d_{jj} in the jth column ($j = 2{:}n$)

$$d_{11} = u_{11} = k_{11}$$

$m_j + 1$

$$u_{ij} = k_{ij} - \sum_{r=1}^{i-1} l_{ir} u_{rj} = k_{ij} - \sum_{r=1}^{i-1} l_{ri} u_{rj} \qquad i = 2 : j-1$$

$\max(m_i, m_j)$

$$d_{jj} = u_{jj} = k_{jj} - \sum_{r=1}^{j-1} l_{jr} u_{rj} = k_{jj} - \sum_{r=1}^{j-1} l_{rj} u_{rj}$$

$$l_{ji} = u_{ij} / d_{ii}$$

m_j

$$l_{rj} = u_{rj} / d_{rr}$$

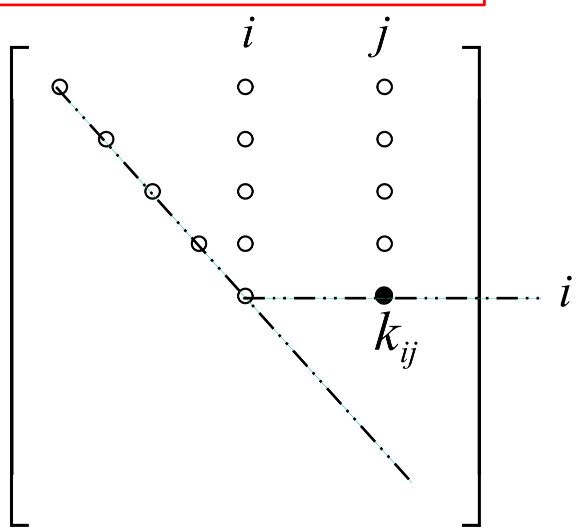
Solution of Equilibrium Equations — Active column solution

The active column solution / skyline (or column) reduction method

$$j = 2:n, \quad i = m_j + 1 : j-1$$

$$\boxed{\begin{aligned} d_{11} &= k_{11} \\ u_{ij} &= k_{ij} - \sum_{r=\mathrm{m}}^{i-1} l_{ri} u_{rj} \\ &\qquad \mathrm{m} = \max(m_i, m_j) \\ d_{jj} &= k_{jj} - \sum_{r=m_j}^{j-1} l_{rj} u_{rj} \\ l_{rj} &= u_{rj} / d_{rr} \end{aligned}}$$

$\boldsymbol{K} =$



- Storage arrangements
  - $k_{ij} \leftarrow l_{ij} \qquad i = m_j + 1, \cdots, j-1$
  - $k_{jj} = d_{jj}$
- Number of operations required

$$\frac{1}{2}\sum_{j=1}^{n}(j - m_j)^2 \approx \frac{1}{2} n m_K^2$$

- Out-of-core solution ?
  - **Block solution** — Assembled and reduced in blocks
  - **Frontal solution method** — Only those equations that are actually required for the elimination of a specific DOF are assembled, the DOF considered is statically condensed out.
  - **PARDISO**(http://www.pardiso-project.org)

Solution of Equilibrium Equations — Active column solution

```
for j = 2:n
    for i = m(j)+1:j-1
        c = 0.0;
        for r = max(m(i),m(j)):i-1
            c = c + K(r,i)*K(r,j);
        end
        K(i,j) = K(i,j) - c;
    end

    for r = m(j):j-1
        Lrj = K(r,j)/K(r,r);
        K(j,j) = K(j,j) - Lrj*K(r,j);
        K(r,j) = Lrj;

        if K(j,j) <= 0
            fprintf('Error - stiffness matrix not positive definite !\n')
            fprintf('        Nonpositive pivot for equation %d\n', n)
            fprintf('        Pivot = %f\n', K(j,j))
        end
    end
end
```

$j = 2:n, \quad i = m_j + 1 : j-1$

$$d_{11} = k_{11}$$

$$u_{ij} = k_{ij} - \sum_{r=\mathrm{m}}^{i-1} l_{ri} u_{rj}$$

$$\mathrm{m} = \max(m_i, m_j)$$

$$d_{jj} = k_{jj} - \sum_{r=m_j}^{j-1} l_{rj} u_{rj}$$

$$l_{rj} = u_{rj} / d_{rr}$$

COLSOL.m

Solution of Equilibrium Equations — Active column solution

$$\boldsymbol{LV} = \boldsymbol{R} \qquad \boldsymbol{L}^{\mathrm{T}}\boldsymbol{a} = \bar{\boldsymbol{V}} \qquad \bar{\boldsymbol{V}} = \boldsymbol{D}^{-1}\boldsymbol{V}$$

$$v_i = r_i - \sum_{j=m_i}^{i-1} l_{ji} v_j, \quad i = 2, \cdots, n \qquad a_n = \bar{v}_n, \quad a_i = \bar{v}_i - \sum_{j=i+1}^{n} l_{ij} \bar{v}_j \quad i = n-1, n-2, \cdots, 1$$

- Number of operations required: $2\sum_{i=1}^{n}(i - m_i) \approx 2nm_K$

```
% Reduce right-hand-side load vector
for i = 2:n
    for j = m(i):i-1
        R(i) = R(i) - K(j,i) * R(j)
    end
end
```

$$v_i = r_i - \sum_{j=m_i}^{i-1} l_{ji} v_j, \quad i = 2, \cdots, n$$

```
%back-substitute
for i = 1:n
    R(i) = R(i)/K(i,i);
end
```

$$\bar{\boldsymbol{V}} = \boldsymbol{D}^{-1}\boldsymbol{V}$$

```
for j = n:-1:2
    for i = m(j):j-1
        R(i) = R(i) - K(i,j)*R(j)
    end
end
```

$$a_i = \bar{v}_i - \sum_{j=i+1}^{n} l_{ij} \bar{v}_j \quad i = n-1, n-2, \cdots, 1$$

先对列循环，再对行循环 $j = n:2$, $i = j-1:1$