CHAP 5. Linear Solver. $\underset{\sim}{K}\underset{\sim}{q} = \underset{\sim}{F}$.

ⓐ Gaussian elimination.

연립방정식 처럼 계수 맞추고 미지수 소거.

n이 클 때 연산 회수 $\sim O(\frac{n^3}{3})$

ⓑ Gaussian elimination with Cholesky decomposition

n이 클때. 연산 회수 $\sim O(\frac{n^3}{6})$

n = 미지수 개수

Cholesky Decomposition

If symmetric $\underset{\sim}{K}$, $\underset{\sim}{K} = U^T U$ 로 쓸수 있다.

(cholesky decomposition)

U = upper triangular matrix

$$= \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & & u_{2n} \\ 0 & 0 & u_{33} & & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

즉 $u_{ij} = 0$ if $i > j$

U^T = transpose of U

= lower triangular matrix.

$\underset{\sim}{K}\,\underset{\sim}{q} = \underset{\sim}{F}$ 에 cholesky decomposition 적용

$$\underset{\sim}{U}^T \underset{\sim}{U}\,\underset{\sim}{q} = \underset{\sim}{F} \qquad (1)$$

set $\underset{\sim}{U}\,\underset{\sim}{q} = \underset{\sim}{Y}$ (2)

Then $\underset{\sim}{U}^T \underset{\sim}{Y} = \underset{\sim}{F}$

$$\begin{bmatrix} u_{11} & 0 & \cdots & 0 \\ u_{12} & u_{22} & & 0 \\ \vdots & \vdots & & \\ u_{1n} & u_{2n} & \cdots & u_{nn} \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{Bmatrix} \qquad (3)$$

식(3)은 y_1 부터 차례로 y_n 까지 계산 가능
(Forward substitution)

식 (2) 에서 $\underset{\sim}{U}\,\underset{\sim}{q} = \underset{\sim}{Y}$

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & u_{nn} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{Bmatrix}$$

q_n 부터 차례로 q_1 까지 계산 가능.

backward substitution

$\underset{\sim}{U}$를 구하는 방법.

$\underset{\sim}{U}^T \underset{\sim}{U} = \underset{\sim}{K}$ 에서

$$\begin{bmatrix} u_{11} & 0 & 0 & \cdots & 0 \\ u_{12} & u_{22} & 0 & \cdots & 0 \\ u_{13} & u_{23} & u_{33} & \cdots & 0 \\ \vdots & & & & \\ u_{1n} & u_{2n} & u_{3n} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & & u_{2n} \\ 0 & 0 & u_{33} & & u_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & u_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots & K_{1n} \\ & K_{22} & K_{23} & & K_{2n} \\ & & K_{33} & & K_{3n} \\ & \text{SYM} & & & \vdots \\ & & & & K_{nn} \end{bmatrix}$$

(1) Row-wise evaluation of u_{ij} $(i \le j)$

ⓐ 1st row.

$$u_{11} u_{11} = K_{11} \longrightarrow u_{11} = \sqrt{K_{11}}$$

$$u_{11} u_{12} = K_{12} \longrightarrow u_{12} = K_{12}/u_{11}$$

$$\vdots$$

$$u_{11} u_{1n} = K_{1n} \longrightarrow u_{1n} = K_{1n}/u_{11}$$

ⓑ 2nd row.

$$u_{12}^2 + u_{22}^2 = K_{22} \longrightarrow u_{22} = \sqrt{K_{22} - u_{12}^2}$$

$$u_{12}u_{13} + u_{22}u_{23} = K_{23} \rightarrow u_{23} = (K_{23} - u_{12}u_{13})/u_{22}$$

Therefore,

diagonal term

$$u_{ii} = \sqrt{K_{ii} - \sum_{l=1}^{i-1} u_{li}^2} \qquad i = 1 \sim n$$

off-diagonal term

$$u_{ij} = \frac{1}{u_{ii}}\left(K_{ij} - \sum_{l=1}^{i-1} u_{li}u_{lj}\right), \quad j = i+1 \sim n$$

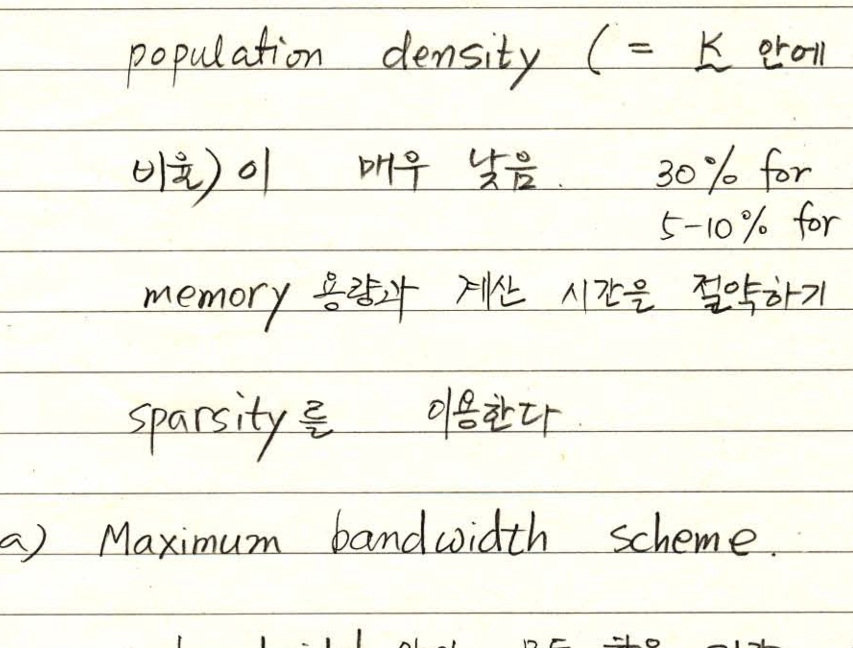
(NOTE) global $\underset{\sim}{K}$ matrix 는 sparse matrix

population density (= $\underset{\sim}{K}$ 안에 nonzero term의 비율)이 매우 낮음. 30% for 2-D
5-10% for 3-D

memory 용량과 계산 시간을 절약하기 위해 $\underset{\sim}{K}$ 의 sparsity를 이용한다.

(a) Maximum bandwidth scheme.

max bandwidth 안의 모든 항을 저장, 연산.



$\underset{\sim}{K}$ $(n \times n)$ $\qquad$ $n \times m$ matrix

Another approach.

$$\underset{\sim}{K} = \underset{\sim}{U}^T \underset{\sim}{D} \underset{\sim}{U}$$

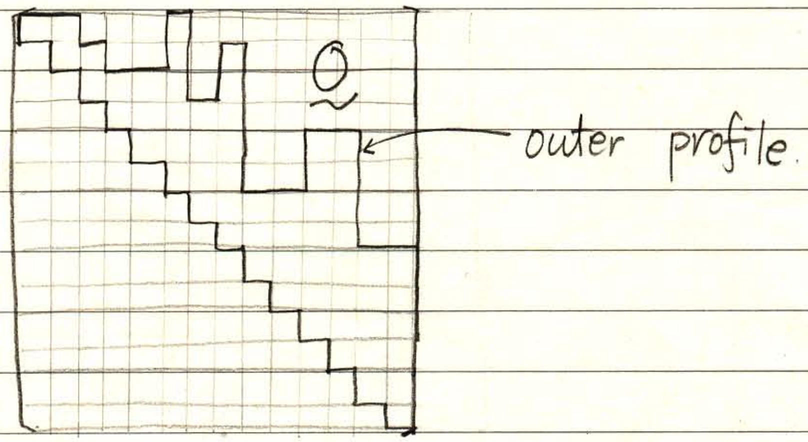
$$\begin{bmatrix} 1 & & & & \\ u_{12} & 1 & & \underset{\sim}{0} & \\ u_{13} & u_{23} & 1 & & \\ \vdots & & & \ddots & \\ u_{1n} & u_{2n} & \cdots & & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & & \\ & d_2 & & \underset{\sim}{0} & \\ & & d_3 & & \\ & \underset{\sim}{0} & & \ddots & \\ & & & & d_n \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & \cdots & u_{1n} \\ & 1 & u_{23} & \cdots & u_{2n} \\ & & 1 & \cdots & u_{3n} \\ & \underset{\sim}{0} & & & \vdots \\ & & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ & K_{22} & \cdots & K_{2n} \\ & & \ddots & \vdots \\ \text{SYM} & & & K_{nn} \end{bmatrix}.$$

(NOTE) Max. bandwidth scheme 은 small scale 문제에 적합.

(b) skyline method.

K의 각 column 마다 첫번째 nonzero 항부터 diagonal 항까지 저장, variable bandwidth 방법의 일종.



skyline profile 내부에 아직 많은 zero terms or "windows"가 있음, medium to large scale 문제에 적합.

(c) Hypermatrix scheme.

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & \cdots & K_{1N} \\ & K_{22} & \cdots & \cdots & K_{2N} \\ & & & & \vdots \\ \text{SYM} & & & & \vdots \\ & & & & K_{NN} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{Bmatrix}$$

K를 NXN의 submatrix로 나눈다

많은 submatrix가 zero matrix

submatrix를 backup storage (disk)에 저장.

large scale 문제에 적당.

$K = U^T U$ 에서 U도 hypermatrix

$$U = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1N} \\ & U_{22} & & U_{2N} \\ & & \ddots & \vdots \\ 0 & & & U_{NN} \end{bmatrix}$$

submatrix level 에서 cholesky decomposition.

for $i = 1$ to N.

$$U_{ii}^T U_{ii} = K_{ii} - \sum_{l=1}^{i-1} U_{li}^T U_{li}$$

cholesky decomposition을 이용 U_{ii}

for $j = i+1$ to N.

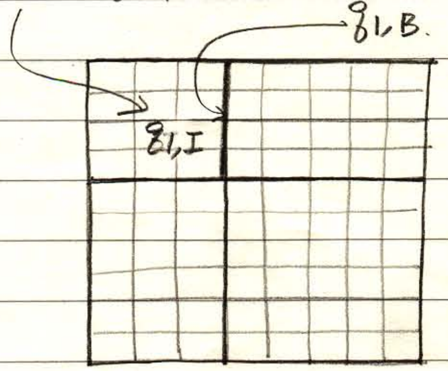
$$U_{ii}^T U_{ij} = \left(K_{ij} - \sum_{l=1}^{i-1} U_{li}^T U_{lj} \right)$$

forward substitution 이용 U_{ij}

U를 구한 뒤 submatrix 단위에서 forward substitution, backward substitution 하여 해를 구한다.

(d) substructure (or subelement) Technique.

Substructure 1



구조물을 작은 단위의 substructure 로 나눈다.

$$\delta\Pi = \sum_{i=1}^{N} \delta\Pi_i$$

i = substructure number

N = total number of substructure

large scale 문제에 적합.

set $\underset{\sim}{q}_{i,I}$ = substructure i 의 내부 dof vector

$\underset{\sim}{q}_{i,B}$ = substructure i 의 경계에 있는 dof vector.

$$\underset{\sim}{q}_i = \begin{Bmatrix} \underset{\sim}{q}_{i,I} \\ \underset{\sim}{q}_{i,B} \end{Bmatrix}$$

Then $\delta\Pi_i = \delta \underset{\sim}{q}_i^T (\underset{\sim}{K}^i \underset{\sim}{q}_i - \underset{\sim}{F}_i)$

$$= \lfloor \delta\underset{\sim}{q}_{i,I} \;\; \delta\underset{\sim}{q}_{i,B} \rfloor \left(\begin{bmatrix} K^i_{II} & K^i_{IB} \\ K^i_{BI} & K^i_{BB} \end{bmatrix} \begin{Bmatrix} \underset{\sim}{q}_{i,I} \\ \underset{\sim}{q}_{i,B} \end{Bmatrix} - \begin{Bmatrix} \underset{\sim}{F}_{i,I} \\ \underset{\sim}{F}_{i,B} \end{Bmatrix} \right)$$

여기서 $\underset{\sim}{K}^i_{BI} = (\underset{\sim}{K}^i_{IB})^T$

Since $\delta \underset{\sim}{q}_{i,I}$ 와 $\delta \underset{\sim}{q}_{i,B}$ 가 임의 값,

$$\underset{\sim}{K}_{II}^{i} \underset{\sim}{q}_{i,I} + \underset{\sim}{K}_{IB}^{i} \underset{\sim}{q}_{i,B} - \underset{\sim}{F}_{i,I} = 0 \qquad (a)$$

$$\underset{\sim}{K}_{BI}^{i} \underset{\sim}{q}_{i,I} + \underset{\sim}{K}_{BB}^{i} \underset{\sim}{q}_{i,B} - \underset{\sim}{F}_{i,B} = 0. \qquad (b)$$

식 (a) 에서 $\underset{\sim}{q}_{i,I} = \left(\underset{\sim}{K}_{II}^{i}\right)^{-1} \left(\underset{\sim}{F}_{i,I} - \underset{\sim}{K}_{IB}^{i} \underset{\sim}{q}_{i,B} \right) \quad (c)$

식 (c)를 (b) 에 대입

$$\left(\underset{\sim}{K}_{BB}^{i} - \underset{\sim}{K}_{BI}^{i} \underset{\sim}{K}_{II}^{i^{-1}} \underset{\sim}{K}_{IB}^{i} \right) \underset{\sim}{q}_{i,B} = \left(\underset{\sim}{F}_{i,B} - \underset{\sim}{K}_{BI}^{i} \underset{\sim}{K}_{II}^{i^{-1}} \underset{\sim}{F}_{i,I} \right) \qquad (d)$$

식 (d) 를 assemble 하여 global stiffness & load vector for $\underset{\sim}{q}_B$

$\underset{\sim}{q}_B$ 계산후 식 (c) 에서 $q_{i,I}$ 계산.

식 (d) 의 size 는 원래 문제보다 작음.

병렬 계산에 활용

(e) Iterative (Indirect) Equation Solver

$\underset{\sim}{K}\,\underset{\sim}{q} = \underset{\sim}{F}$ 에서

residual $\Delta\underset{\sim}{F} = \underset{\sim}{F} - \underset{\sim}{K}\,\underset{\sim}{q}$

if $\underset{\sim}{q}$ is exact solution, Then $\Delta\underset{\sim}{F} = 0$

if $\underset{\sim}{q}^1$ is initially guessed solution,

$\Delta\underset{\sim}{F}^1 = \underset{\sim}{F} - \underset{\sim}{K}\,\underset{\sim}{q}^1$

$\underset{\sim}{K}\,\Delta\underset{\sim}{q}^1 = \Delta\underset{\sim}{F}^1 \longrightarrow$ solve for $\Delta\underset{\sim}{q}^1$

$\underset{\sim}{q}^2 = \underset{\sim}{q}^1 + \Delta\underset{\sim}{q}^1$

$\underset{\sim}{K}\,\Delta\underset{\sim}{q}^2 = \Delta\underset{\sim}{F}^2 = \underset{\sim}{F} - \underset{\sim}{K}\,\underset{\sim}{q}^2$

$\longrightarrow$ solve for $\Delta\underset{\sim}{q}^2$

$\underset{\sim}{q}^3 = \underset{\sim}{q}^2 + \Delta\underset{\sim}{q}^2$

$\vdots$

repeat unitil $e = \dfrac{(\underset{\sim}{q}^k)^T \Delta\underset{\sim}{F}^k}{(\underset{\sim}{q}^k)^T \underset{\sim}{F}} <$ tolerance

$= \dfrac{\text{work done by residual force}}{\text{work done by applied force}}$

* Similar to modified Newton-Raphson method (constant stiffness)
* Ill-conditioned equations converges slowly

Modified Version of iterative solver.

$$\underset{3\times3}{\underset{\sim}{K}}\,\underset{\sim}{q} = \underset{\sim}{F}$$

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}.$$

q_i^0 = initial guess of q_i

(ex) $q_i^0 = 0 \quad (i=1, \cdots n)$.

$$q_1^1 = (F_1 - K_{12} q_2^0 - K_{13} q_3^0)/K_{11}$$

$$q_2^1 = (F_2 - K_{21} q_1^1 - K_{23} q_3^0)/K_{22}$$

$$q_3^1 = (F_3 - K_{31} q_1^1 - K_{32} q_2^1)/K_{33}$$

$$\vdots$$

$$q_1^K = (F_1 - K_{12} q_2^{K-1} - K_{13} q_3^{K-1})/K_{11}$$

$$q_2^K = (F_2 - K_{21} q_1^K - K_{23} q_3^{K-1})/K_{22} \qquad (1)$$

$$q_3^K = (F_3 - K_{31} q_1^K - K_{32} q_2^K)/K_{33}$$

Gauss - Seidel Iteration

Introduce overrelaxation factor β

$$q_i^K \xRightarrow{\text{replace}} \beta q_i^K + (1-\beta) q_i^{K-1}$$

$$= q_i^{K-1} + \beta (q_i^K - q_i^{K-1})$$

Then in general form

$$q_i^K = q_i^{K-1} + \frac{\beta}{K_{ii}} \underbrace{\left(F_i - \sum_{j=1}^{i-1} K_{ij} q_j^K - \sum_{j=i}^{n} K_{ij} q_j^{K-1} \right)}_{\text{"residual force}}$$

If $\underset{\sim}{K}$ is positive definite and stable structure, converges when $0 < \beta < 2$.

If $\beta = 1$, same as (1) ; Gauss-Seidel iteration

If $\beta > 1$, then SOR (successive overrelaxation)

Optimum value $\beta \simeq 1.6$ (problem dependent)

Iterative method

advantages

1. easy to program
2. 정확도가 중요치 않을 때 경제적 (few iteration)
3. nonlinear 문제 또는 반복해석 설계에 유리 ($\because$ good initial solution)

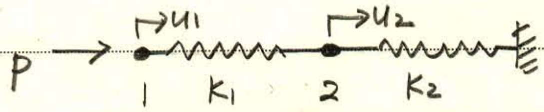
disadvantages

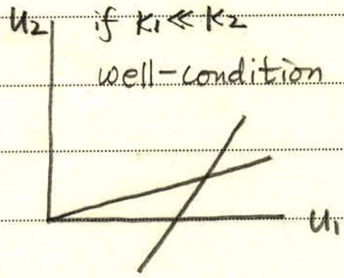
1. ill-condition 문제 ; 수렴 느림.
2. 계산시간 예측이 어려움.

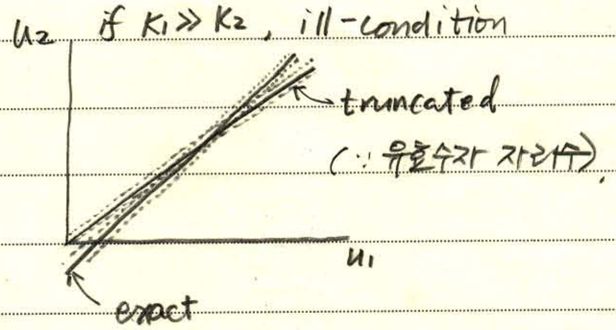
3. symmetric K의 장점이 있음 (direct method에서 sym K는 계산량이 $\frac{1}{2}$, choleskian decomposition)

Ill-conditioned problem due to truncation error

$$\begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1+k_2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} P \\ 0 \end{Bmatrix} \quad \text{or} \quad \begin{aligned} k_1u_1 - k_1u_2 &= P \quad (1) \\ -k_1u_1 + (k_1+k_2)u_2 &= 0 \quad (2) \end{aligned}$$







$(1)+(2) \quad [(k_1+k_2) - k_1]\,u_2 = P \qquad (3)$

analytically correct result $= \quad k_2u_2 = P$

if $k_1 \gg k_2$, $\quad k_1 = 1.000000$, $\quad k_2 = 4.444444 \times 10^{-6}$

if 7 digits is stored

식 (3) $\Rightarrow \quad 1.000004 - 1.000000 = 4 \times 10^{-6}$

if 6 digits is stored

식 (3) $\Rightarrow \quad 1.00000 - 1.00000 = 0 \quad$: rigid body motion singular prob.

(예) ill-conditioned

rigid body supported by flexible region