Solution Technique (Chapter 11)

- Inversion
- Direct Method
  - Cramer's Rule
  - Gauss Elimination
  - LU factorization, Cholesky's method
- Indirect Method
  - Inverse Iteration
    - Jacobi method : total steps
    - Gauss-Seidel method : single step

Inversion

$$\underline{K}\underline{U} = \underline{P}, \qquad \underline{U} = \underline{K}^{-1}\underline{P}$$

Inefficient method since it requires the evaluation of a number of determinants of high order in calculation of $\underline{K}^{-1}$

Gauss Elimination

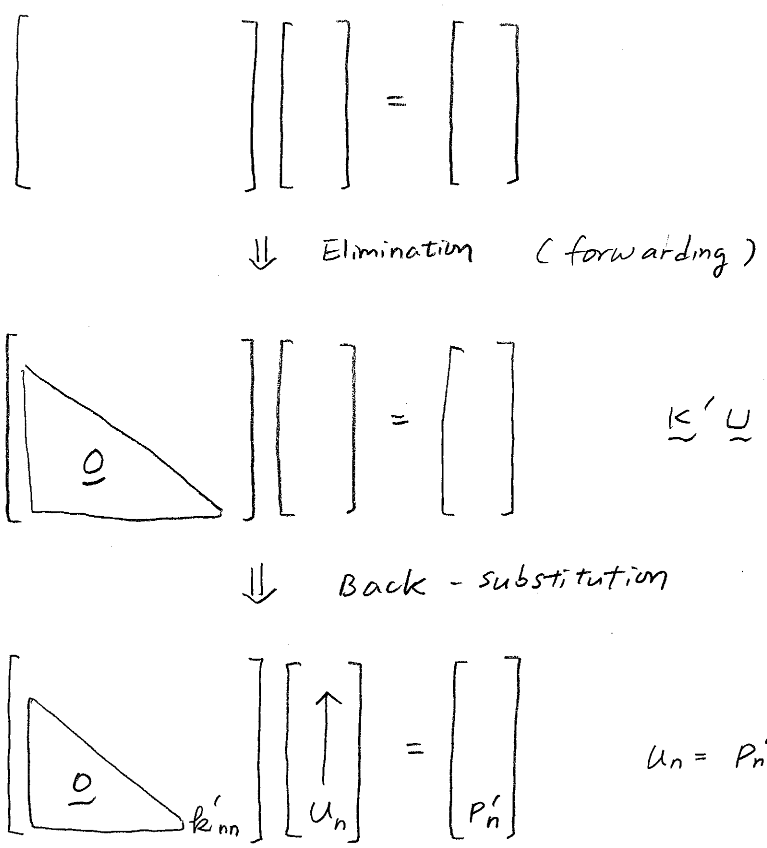
Characteristics of matrix calculation does not change even after row- and column-calculations.

A systematic procedure for making a triangular matrix

Or for eliminating variables one by one

1) forwarding
2) backwarding (back-substitution)

$$\underset{\sim}{K}\,\underset{\sim}{U} = \underset{\sim}{P}$$



$$\underset{\sim}{K}'\,\underset{\sim}{U} = \underset{\sim}{P}'$$

$$u_n = p_n' / k_{nn}'$$

LU - Factorization

$\underset{\sim}{K}\,\underset{\sim}{V} = \underset{\sim}{P}$ $\underset{\sim}{V}$ = unknown displacement

$\underset{\sim}{K} = \underset{\sim}{L}\,\underset{\sim}{U}$ $\underset{\sim}{L}$ = Lower triangular matrix

$\underset{\sim}{U}$ = upper triangular matrix

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$k_{11} = u_{11}$, $k_{12} = u_{12}$, $k_{13} = u_{13}$

$k_{21} = m_{21} u_{11}$, $k_{22} = m_{21} u_{12} + u_{22}$, etc

$\underset{\sim}{K}\,\underset{\sim}{V} = \underset{\sim}{P} \Rightarrow \underset{\sim}{L}\,\underset{\sim}{U}\,\underset{\sim}{V} = \underset{\sim}{P}$ set $\underset{\sim}{U}\,\underset{\sim}{V} = \underset{\sim}{Y}$

$\underset{\sim}{L}\,\underset{\sim}{Y} = \underset{\sim}{P}$ solve $\underset{\sim}{Y}$

$\underset{\sim}{U}\,\underset{\sim}{V} = \underset{\sim}{Y}$ solve $\underset{\sim}{V}$

cholesky's method

If $\underset{\sim}{K}$ is symmetric and positive definite ($\underset{\sim}{X}^T \underset{\sim}{K}\,\underset{\sim}{X} > 0$),

$\underset{\sim}{U} = \underset{\sim}{L}^T$ $\underset{\sim}{K} = \underset{\sim}{L}\,\underset{\sim}{L}^T$

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

$$\underset{\sim}{K}\,\underset{\sim}{V} = \underset{\sim}{P} \Rightarrow \underset{\sim}{L}\,\underset{\sim}{L}^T\,\underset{\sim}{V} = \underset{\sim}{P} \quad \text{set} \quad \underset{\sim}{L}^T\underset{\sim}{V} = \underset{\sim}{Y}$$

$$\underset{\sim}{L}\,\underset{\sim}{Y} = \underset{\sim}{P} \quad \text{solve } \underset{\sim}{Y}$$

$$\underset{\sim}{L}^T\underset{\sim}{V} = \underset{\sim}{Y} \quad \text{solve } \underset{\sim}{V}$$

Iterative Method

By using iterative method, rapid convergence can be achived,

when 1) matrix has large main diagonal entries

2) matrix are sparse, that is, have very many zeros

$$\underset{\sim}{K}\,\underset{\sim}{u} = \underset{\sim}{P} \quad \Rightarrow \quad \underset{\sim}{K}'\,\underset{\sim}{u} = \underset{\sim}{P}'$$

$\underset{\sim}{K}'$: all diagonal terms are 1's.

$$\underbrace{\begin{bmatrix} 1 & -0.25 & -0.25 & 0 \\ -0.25 & 1 & 0 & -0.25 \\ -0.25 & 0 & 1 & -0.25 \\ 0 & -0.25 & -0.25 & 1 \end{bmatrix}}_{\underset{\sim}{K}'} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_{\underset{\sim}{u}} = \underbrace{\begin{bmatrix} 50 \\ 50 \\ 25 \\ 25 \end{bmatrix}}_{\underset{\sim}{P}'}$$

Gauss - Seidel Iteration Method (Successive correction)

Assuming $\underset{\sim}{u}^{(0)} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}$

$$u_1^{(1)} = \qquad + 0.25\,u_2^{(0)} + 0.25\,u_3^{(0)} \qquad + 50 = 100$$

$$u_2^{(1)} = 0.25\,u_1^{(1)} \qquad + 0.25\,u_4^{(0)} + 50 = 100$$

$$u_3^{(1)} = 0.25\,u_1^{(1)} \qquad + 0.25\,u_4^{(0)} + 25 = 75$$

$$u_4^{(1)} = \qquad 0.25\,u_2^{(1)} + 0.25\,u_3^{(1)} \qquad \sout{+25} = 68.75$$

iterative calculation is required until $\underset{\sim}{u}^{(0)} \parallel \underset{\sim}{u}^{(1)}$

$$\underset{\sim}{u}^{(0)} \cdot \underset{\sim}{u}^{(1)} = \|\underset{\sim}{u}^{(0)}\|\,\|\underset{\sim}{u}^{(1)}\| \cdot \cos\theta$$

if $\underset{\sim}{u}^{(0)} \parallel \underset{\sim}{u}^{(1)}$, $\cos\theta \rightarrow 1.0$

$$\underset{\sim}{K} = \underset{\sim}{I} + \underset{\sim}{L} + \underset{\sim}{U}$$

$$\underset{\sim}{K}\,\underset{\sim}{u} = (\underset{\sim}{I} + \underset{\sim}{L} + \underset{\sim}{U})\,\underset{\sim}{u} = \underset{\sim}{P}$$

$$\underset{\sim}{u}^{(m+1)} = \underset{\sim}{P} - \underset{\sim}{L}\,\underset{\sim}{u}^{(m+1)} - \underset{\sim}{U}\,\underset{\sim}{u}^{(m)}$$

$$(\underset{\sim}{I} + \underset{\sim}{L})\,\underset{\sim}{u}^{(m+1)} = \underset{\sim}{P} - \underset{\sim}{U}\,\underset{\sim}{u}^{(m)}$$

$$\underset{\sim}{u}^{(m+1)} = (\underset{\sim}{I} + \underset{\sim}{L})^{-1}\underset{\sim}{P} - (\underset{\sim}{I} + \underset{\sim}{L})^{-1}\underset{\sim}{U}\,\underset{\sim}{u}^{(m)}$$

Jacobi Iteration Method - (Simultaneous correction)

$$\underset{\sim}{K}\,\underset{\sim}{u} = \underset{\sim}{I}\,\underset{\sim}{u} + (\underset{\sim}{K} - \underset{\sim}{I})\,\underset{\sim}{u} = \underset{\sim}{P}$$

$$\underset{\sim}{u}^{(m+1)} = \underset{\sim}{P} + (\underset{\sim}{I} - \underset{\sim}{K})\,\underset{\sim}{u}^{(m)}$$

An **LU-factorization** of a given square matrix A is of the form

(2) $$\boxed{\mathbf{A} = \mathbf{LU}}$$

where L is lower triangular and U is upper triangular. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 8 & 5 \end{bmatrix} = \mathbf{LU} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & -7 \end{bmatrix}.$$

It can be proved that for any nonsingular matrix (see Sec. 6.7) the rows can be reordered so that the resulting matrix A has an LU-factorization (2) in which L turns out to be the matrix of the *multipliers* m_{jk} of the Gauss elimination, with main diagonal 1, · · · , 1, and U is the matrix of the triangular system at the end of the Gauss elimination. (See Ref. [E3], pp. 155–156, listed in Appendix 1.)

The *crucial idea* now is that L and U in (2) can be computed directly, without solving simultaneous equations (thus, without using the Gauss elimination). As a count shows, this needs about $n^3/3$ operations, about half as many as the Gauss elimination, which needs about $2n^3/3$ (see Sec. 18.1). And once we have (2), we can use it for solving $\mathbf{Ax} = \mathbf{b}$ in two steps, involving only about n^2 operations, simply by noting that $\mathbf{Ax} = \mathbf{LUx} = \mathbf{b}$ may be written

(3) $$\boxed{\text{(a)}\quad \mathbf{Ly} = \mathbf{b} \qquad \text{where} \qquad \text{(b)}\quad \mathbf{Ux} = \mathbf{y}}$$

and solving first (3a) for y and then (3b) for x. This is called **Doolittle's method.** Both systems (3a) and (3b) are triangular, so their solution is the same as back substitution in the Gauss elimination.

A similar method, **Crout's method,** is obtained from (2) if U (instead of L) is required to have main diagonal 1, · · · , 1. In either case the factorization (2) is unique.

Doolittle's method

Solve the system in Example 1 of Sec. 18.1 by Doolittle's method.

Solution. The decomposition (2) is obtained from

$$\mathbf{A} = [a_{jk}] = \begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

by determining the m_{jk} and u_{jk}, using matrix multiplication. By going through A row by row we get successively

$a_{11} = 3 = u_{11}$	$a_{12} = 5 = u_{12}$	$a_{13} = 2 = u_{13}$
$a_{21} = 0 = m_{21}u_{11}$	$a_{22} = 8 = m_{21}u_{12} + u_{22}$	$a_{23} = 2 = m_{21}u_{13} + u_{23}$
$m_{21} = 0$	$u_{22} = 8$	$u_{23} = 2$
$a_{31} = 6 = m_{31}u_{11}$	$a_{32} = 2 = m_{31}u_{12} + m_{32}u_{22}$	$a_{33} = 8 = m_{31}u_{13} + m_{32}u_{23} + u_{33}$
$= m_{31} \cdot 3$	$= 2 \cdot 5 + m_{32} \cdot 8$	$= 2 \cdot 2 - 1 \cdot 2 + u_{33}$
$m_{31} = 2$	$m_{32} = -1$	$u_{33} = 6$

(4)
$$\begin{aligned}
u_{1k} &= a_{1k} && k = 1, \cdots, n \\
m_{j1} &= \frac{a_{j1}}{u_{11}} && j = 2, \cdots, n \\
u_{jk} &= a_{jk} - \sum_{s=1}^{j-1} m_{js}u_{sk} && k = j, \cdots, n; \quad j \geqq 2 \\
m_{jk} &= \frac{1}{u_{kk}}\left(a_{jk} - \sum_{s=1}^{k-1} m_{js}u_{sk}\right) && j = k+1, \cdots, n; \quad k \geqq 2.
\end{aligned}$$

Cholesky's Method

For a *symmetric, positive definite* matrix **A** (thus $\mathbf{A} = \mathbf{A}^T$, $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$) we can in (2) even choose $\mathbf{U} = \mathbf{L}^T$, thus $u_{jk} = m_{kj}$ (but impose no conditions on the main

The popular method of solving $\mathbf{Ax} = \mathbf{b}$ based on this factorization $\mathbf{A} = \mathbf{LL}^T$ is called **Cholesky's method.** In terms of the entries of $\mathbf{L} = [l_{jk}]$ the formulas for the factorization are

$$l_{11} = \sqrt{a_{11}}$$

$$l_{j1} = \frac{a_{j1}}{l_{11}} \qquad j = 2, \cdots, n$$

(6) $$l_{jj} = \sqrt{a_{jj} - \sum_{s=1}^{j-1} l_{js}^2} \qquad j = 2, \cdots, n$$

$$l_{pj} = \frac{1}{l_{jj}}\left(a_{pj} - \sum_{s=1}^{j-1} l_{js}l_{ps}\right) \qquad p = j+1, \cdots, n; \quad j \geqq 2.$$

If **A** is symmetric but not positive definite, this method could still be applied, but then leads to a *complex* matrix **L**, so that it becomes impractical.

Cholesky's method

Solve by Cholesky's method:

$$\begin{aligned} 4x_1 + 2x_2 + 14x_3 &= 14 \\ 2x_1 + 17x_2 - 5x_3 &= -101 \\ 14x_1 - 5x_2 + 83x_3 &= 155. \end{aligned}$$

Solution. From (6) or from the form of the factorization

$$\begin{bmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

we compute, in the given order,

$$l_{11} = \sqrt{a_{11}} = 2 \qquad l_{21} = \frac{a_{21}}{l_{11}} = \frac{2}{2} = 1 \qquad l_{31} = \frac{a_{31}}{l_{11}} = \frac{14}{2} = 7$$

$$l_{22} = \sqrt{a_{22} - l_{21}^2} = \sqrt{17 - 1} = 4$$

$$l_{32} = \frac{1}{l_{22}}(a_{32} - l_{31}l_{21}) = \frac{1}{4}(-5 - 7 \cdot 1) = -3$$

$$l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2} = \sqrt{83 - 7^2 - (-3)^2} = 5.$$

Gauss–Seidel Iteration Method

This is an iterative method of great practical importance, which we can simply explain in terms of an example.

Gauss–Seidel iteration

We consider the linear system

$$
\begin{aligned}
x_1 - 0.25x_2 - 0.25x_3 \qquad\qquad &= 50 \\
-0.25x_1 + x_2 \qquad\qquad - 0.25x_4 &= 50 \\
-0.25x_1 \qquad\qquad + x_3 - 0.25x_4 &= 25 \\
- 0.25x_2 - 0.25x_3 + x_4 &= 25.
\end{aligned} \tag{1}
$$

(Equations of this form arise in the numerical solution of partial differential equations and in spline interpolation.) We write the system in the form

$$
\begin{aligned}
x_1 &= \qquad 0.25x_2 + 0.25x_3 \qquad + 50 \\
x_2 &= 0.25x_1 \qquad\qquad + 0.25x_4 + 50 \\
x_3 &= 0.25x_1 \qquad\qquad + 0.25x_4 + 25 \\
x_4 &= \qquad 0.25x_2 + 0.25x_3 \qquad + 25.
\end{aligned} \tag{2}
$$

We use these equations for iteration, that is, we start from a (possibly poor) approximation to the solution, say, $x_1^{(0)} = 100$, $x_2^{(0)} = 100$, $x_3^{(0)} = 100$, $x_4^{(0)} = 100$, and compute from (2) a presumably better approximation

Use "old" values

("New" values here not yet available)

$$
\begin{aligned}
x_1^{(1)} &= \qquad 0.25x_2^{(0)} + 0.25x_3^{(0)} \qquad + 50.00 = 100.00 \\
x_2^{(1)} &= 0.25x_1^{(1)} \qquad\qquad + 0.25x_4^{(0)} + 50.00 = 100.00 \\
x_3^{(1)} &= 0.25x_1^{(1)} \qquad\qquad + 0.25x_4^{(0)} + 25.00 = 75.00 \\
x_4^{(1)} &= \qquad 0.25x_2^{(1)} + 0.25x_3^{(1)} \qquad + 25.00 = 68.75.
\end{aligned} \tag{3}
$$

Use "new" values

We see that these equations are obtained from (2) by substituting on the right the ***most recent*** approximations. In fact, corresponding elements replace previous ones as soon as they have been computed, so that in the second and third equations we use $x_1^{(1)}$ (not $x_1^{(0)}$), and in the last equation of (3) we use $x_2^{(1)}$ and $x_3^{(1)}$ (not $x_2^{(0)}$ and $x_3^{(0)}$). The next step yields

$$
\begin{aligned}
x_1^{(2)} &= \qquad 0.25x_2^{(1)} + 0.25x_3^{(1)} \qquad + 50.00 = 93.75 \\
x_2^{(2)} &= 0.25x_1^{(2)} \qquad\qquad + 0.25x_4^{(1)} + 50.00 = 90.62 \\
x_3^{(2)} &= 0.25x_1^{(2)} \qquad\qquad + 0.25x_4^{(1)} + 25.00 = 65.62 \\
x_4^{(2)} &= \qquad 0.25x_2^{(2)} + 0.25x_3^{(2)} \qquad + 25.00 = 64.06.
\end{aligned}
$$

In practice, one would do further steps and obtain a more accurate approximate solution.

The reader may show that the exact solution is $x_1 = x_2 = 87.5$, $x_3 = x_4 = 62.5$.

To obtain an algorithm for the Gauss–Seidel iteration, let us derive the general formulas for this iteration.

We assume that $a_{jj} = 1$ for $j = 1, \cdots, n$. (Note that this can be achieved if we can rearrange the equations so that no diagonal coefficient is zero; then we may divide each equation by the corresponding diagonal coefficient.) We now write

$$\text{(4)} \qquad \mathbf{A} = \mathbf{I} + \mathbf{L} + \mathbf{U} \qquad (a_{jj} = 1)$$

where $\mathbf{I}$ is the $n \times n$ unit matrix and $\mathbf{L}$ and $\mathbf{U}$ are respectively lower and upper triangular matrices with zero main diagonals. If we substitute (4) into $\mathbf{Ax} = \mathbf{b}$, we have

$$\mathbf{Ax} = (\mathbf{I} + \mathbf{L} + \mathbf{U})\mathbf{x} = \mathbf{b}.$$

Taking $\mathbf{Lx}$ and $\mathbf{Ux}$ to the right, we obtain, since $\mathbf{Ix} = \mathbf{x}$,

$$\text{(5)} \qquad \mathbf{x} = \mathbf{b} - \mathbf{Lx} - \mathbf{Ux}.$$

Remembering from our computation in Example 1 that below the main diagonal we took "new" approximations and above the main diagonal "old" approximations, we obtain from (5) the desired iteration formulas

$$\text{(6)} \qquad \boxed{\mathbf{x}^{(m+1)} = \mathbf{b} - \mathbf{Lx}^{(m+1)} - \mathbf{Ux}^{(m)}} \qquad (a_{jj} = 1)$$

where $\mathbf{x}^{(m)} = [x_j^{(m)}]$ is the mth approximation and $\mathbf{x}^{(m+1)} = [x_j^{(m+1)}]$ is the $(m + 1)$st approximation. In components this gives the formula in line 1 in Table 18.2. The matrix

Jacobi Iteration

The Gauss–Seidel iteration is a method of **successive corrections** because we replace approximations by corresponding new ones as soon as the latter have been computed. A method is called a method of **simultaneous corrections** if no component of an approximation $\mathbf{x}^{(m)}$ is used until *all* the components of $\mathbf{x}^{(m)}$ have been computed. A method of this type is the **Jacobi iteration**, which is similar to the Gauss–Seidel iteration but involves *not* using improved values until a step has been completed and then replacing $\mathbf{x}^{(m)}$ by $\mathbf{x}^{(m+1)}$ at once, directly before the beginning of the next cycle. Hence, if we write $\mathbf{Ax} = \mathbf{b}$ (*with* $a_{jj} = 1$ *as before!*) in the form $\mathbf{x} = \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}$, the Jacobi iteration in matrix notation is

$$\text{(13)} \qquad \boxed{\mathbf{x}^{(m+1)} = \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}^{(m)}} \qquad (a_{jj} = 1).$$

This method converges for every choice of $\mathbf{x}^{(0)}$ if and only if the spectral radius of $\mathbf{I} - \mathbf{A}$ is less than 1. It has recently gained greater practical interest since on parallel processors all n equations can be solved simultaneously at each iteration step.

Modification of equations considering the support conditions

$$\underset{\sim}{K}\,\underset{\sim}{u} = \underset{\sim}{P}$$

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & & \ddots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix}$$

$$\left[\begin{array}{c|c} \underset{\sim}{k}_{ff} & \underset{\sim}{k}_{fs} \\ \hline \underset{\sim}{k}_{sf} & \underset{\sim}{k}_{ss} \end{array}\right] \left[\begin{array}{c} \underset{\sim}{u}_f \\ \hline \underset{\sim}{u}_s \end{array}\right] = \left[\begin{array}{c} \underset{\sim}{P}_f \\ \hline \underset{\sim}{P}_s \end{array}\right]$$

f : free

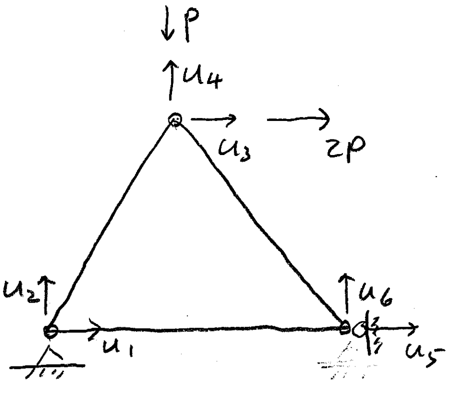
s : support

$\underset{\sim}{u}_f$ = unknown displacements

$\underset{\sim}{P}_f$ = known (applied) forces

$\underset{\sim}{u}_s$ = known (or given) displacements (boundary displacements)

$\underset{\sim}{P}_s$ = unknown forces (reactions)



$$\underset{\sim}{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ u_3 \\ u_4 \\ 0 \\ u_6 \end{bmatrix} \Rightarrow \underset{\sim}{u}_f \; (u_3, u_4, u_6)$$

$$\underset{\sim}{P} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ 2P \\ -P \\ P_5 \\ 0 \end{bmatrix} \Rightarrow \underset{\sim}{P}_f \; (2P, -P, 0)$$

$$\underset{\sim}{k}_{ff}\,\underset{\sim}{u}_f + \underset{\sim}{k}_{fs}\,\underset{\sim}{u}_s = \underset{\sim}{P}_f$$

$$\underset{\sim}{k}_{ff}\,\underset{\sim}{u}_f = \underset{\sim}{P}_f - \underset{\sim}{k}_{fs}\,\underset{\sim}{u}_s \quad \Rightarrow \text{ solve } \underset{\sim}{u}_f$$

$$\underset{\sim}{k}_{sf}\,\underset{\sim}{u}_f + \underset{\sim}{k}_{ss}\,\underset{\sim}{u}_s = \underset{\sim}{P}_s \quad \Rightarrow \text{ solve } \underset{\sim}{P}_s$$

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ & & k_{33} & k_{34} & k_{35} & k_{36} \\ & \text{sym.} & & k_{44} & k_{45} & k_{46} \\ & & & & k_{55} & k_{56} \\ & & & & & k_{66} \end{bmatrix} \begin{bmatrix} \cancel{u_1}^{0} \\ \cancel{u_2}^{0} \\ u_3 \\ u_4 \\ \cancel{u_5}^{0} \\ u_6 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ 2P \\ -P \\ P_5 \\ 0 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} k_{33} & k_{34} & k_{36} \\ k_{43} & k_{44} & k_{46} \\ k_{63} & k_{64} & k_{66} \end{bmatrix}}_{\underset{\sim}{k}_{ff}} \underbrace{\begin{bmatrix} u_3 \\ u_4 \\ u_6 \end{bmatrix}}_{\underset{\sim}{u}_f} = \underbrace{\begin{bmatrix} 2P \\ -P \\ 0 \end{bmatrix}}_{\underset{\sim}{P}_f} - \underset{\sim}{k}_{fs}\,\cancel{\underset{\sim}{u}_s}^{\,0}$$

solve $\underset{\sim}{u}_f$

$$\underbrace{\begin{bmatrix} k_{13} & k_{14} & k_{16} \\ k_{23} & k_{24} & k_{26} \\ k_{53} & k_{54} & k_{56} \end{bmatrix}}_{\underset{\sim}{k}_{sf}} \underbrace{\begin{bmatrix} u_3 \\ u_4 \\ u_6 \end{bmatrix}}_{\underset{\sim}{u}_f} = \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ P_5 \end{bmatrix}}_{\underset{\sim}{P}_s} - \underset{\sim}{k}_{ss}\,\cancel{\underset{\sim}{u}_s}^{\,0}$$

solve $\underset{\sim}{P}_s$

Technique not disturbing the size and order of the original stiffness matrix

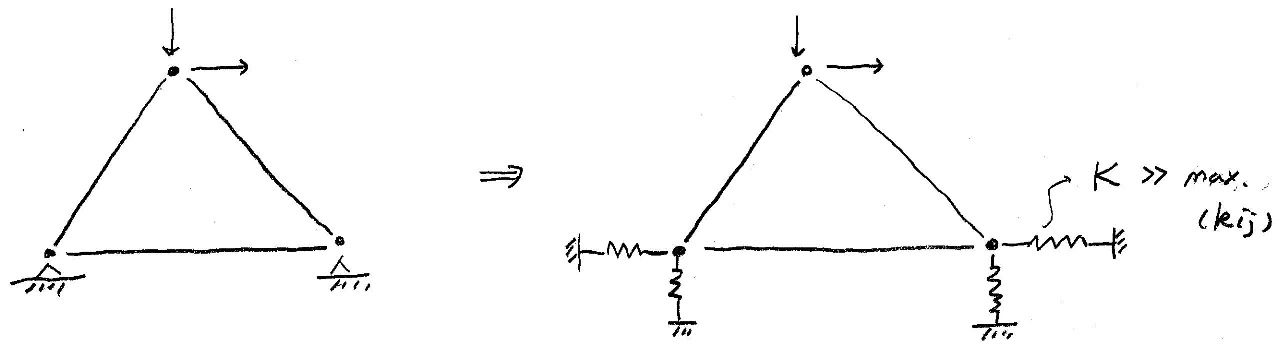
$$\begin{bmatrix} \underset{\sim}{k}_{ff} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{I} \end{bmatrix} \begin{bmatrix} \underset{\sim}{u}_f \\ \underset{\sim}{u}_s \end{bmatrix} = \begin{bmatrix} \underset{\sim}{P}_f - \underset{\sim}{k}_{fs} \underset{\sim}{u}_s \\ \underset{\sim}{u}_s \end{bmatrix}$$

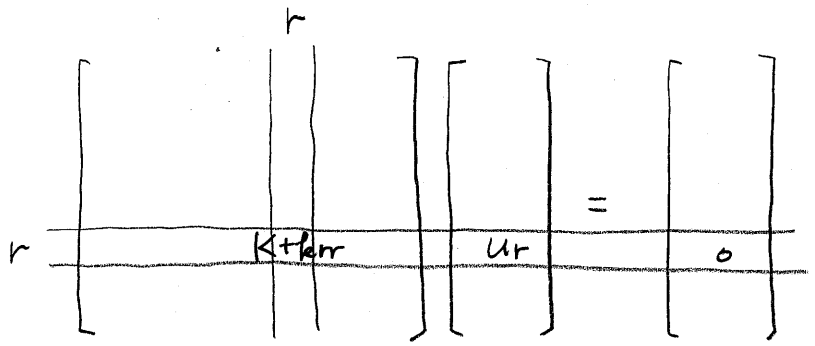
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{33} & k_{34} & 0 & k_{36} \\ 0 & 0 & k_{43} & k_{44} & 0 & k_{46} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & k_{63} & k_{64} & 0 & k_{66} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} \cancel{u_1}\ 0 \\ \cancel{u_2}\ 0 \\ 2P \\ -P \\ \cancel{u_5}\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ k_{31}u_1 + k_{32}u_2 + k_{35}u_5 \\ k_{41}u_1 + k_{42}u_2 + k_{45}u \\ 0 \\ k_{61}u_1 + k_{62}u_2 + k_{65}u_5 \end{bmatrix}$$

solve $\underset{\sim}{u}_f$

$$\underset{\sim}{k}_{sf}\underset{\sim}{u}_f + \underset{\sim}{k}_{ss}\underset{\sim}{u}_s = \underset{\sim}{P}_s \quad \Rightarrow \text{solve } \underset{\sim}{P}_s$$

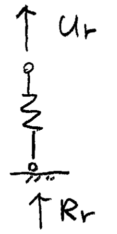
Modification of stiffness matrix with big spring





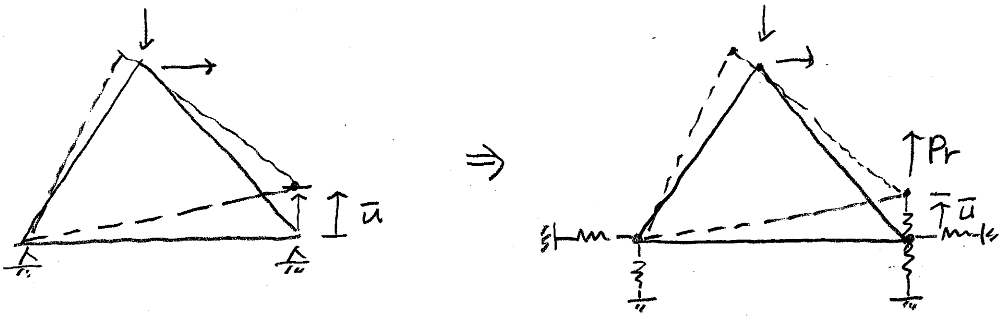
$K = 10^7 \times \max(k_{ij})$

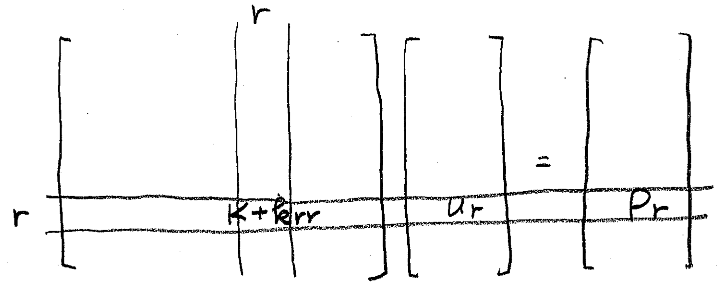
solve U_r



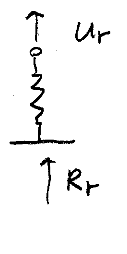
reaction $R_r = -K\,U_r$ by equilibrium

Non-zero displacement boundary condition $\bar{u}$





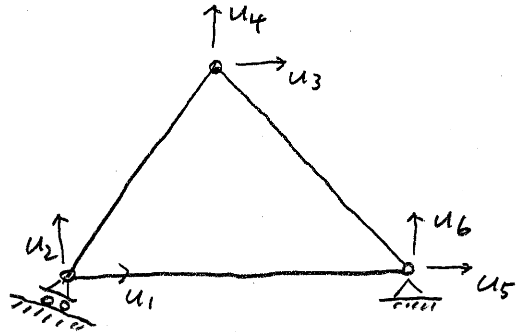
$P_r = K \cdot \bar{u}$

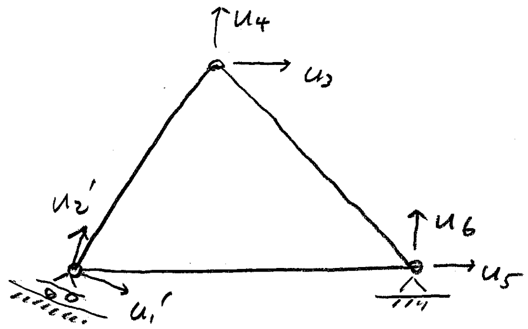


reaction $R_r = -K\,U_r$

net reaction $R_r' = -K\,(U_r - \bar{u})$

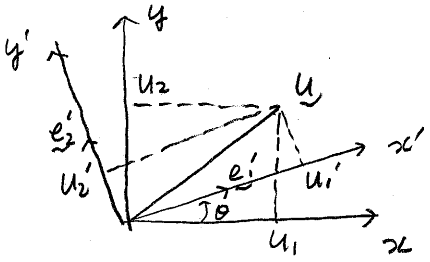
Inclined supports





$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} u_1' \\ u_2' \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}$$

u_1', u_2' ⇒ local coordinate system



$$\mathbf{e}_1' = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$ directional vector of x' axis

$$\mathbf{e}_2' = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$ directional vector of y' axis

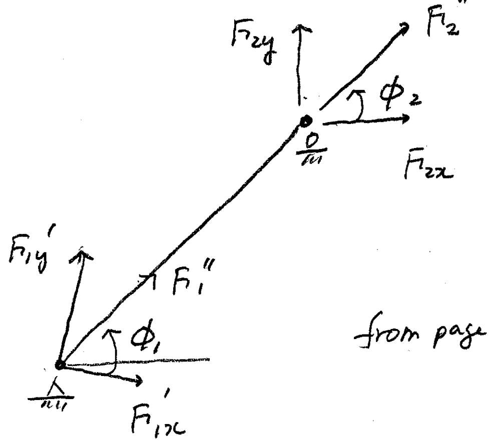
$$u_1' = \mathbf{e}_1' \cdot \mathbf{u} = \mathbf{e}_1'^T \mathbf{u}$$

$$u_2' = \mathbf{e}_2' \cdot \mathbf{u} = \mathbf{e}_2'^T \mathbf{u}$$

$$\begin{bmatrix} u_1' \\ u_2' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\mathbf{u}' = \mathbf{R}\,\mathbf{u} \qquad \mathbf{u} = \mathbf{R}^T \mathbf{u}'$$

calculation of element stiffness in terms of local axis



when $u_1' = v_1' = v_2' = 0$,

$$F_{2x} = F_2'' \cos\phi_2 = \frac{EA}{L} \cos^2\phi_2 \, u_2$$

$$F_{2y} = F_2'' \sin\phi_2 = \frac{EA}{L} \cos\phi_2 \sin\phi_2 \, u_2$$

from page 17.

$$F_{1x}' = F_1'' \cos\phi_1 = -\frac{EA}{L} \cos\phi_1 \cos\phi_2 \, u_2$$

$$F_{1y}' = F_1'' \sin\phi_1 = -\frac{EA}{L} \cos\phi_2 \sin\phi_1 \, u_2$$

$$\underset{\sim}{k} = \frac{EA}{L} \begin{bmatrix} \; & \; & -\cos\phi_1 \cos\phi_2 & \; \\ & & -\cos\phi_2 \sin\phi_1 & \\ & & \cos^2\phi_2 & \\ & & \cos\phi_2 \sin\phi_2 & \end{bmatrix}$$

Element stiffness matrix should be constructed considering the local coordinate system.

The bottom horizontal element's stiffnes can also be calulated considering the local axis.