Numerical Analysis – Linear Equations(I)

Hanyang University

Jong-Il Park

Linear equations

- *N* unknowns, *M* equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2N}x_N &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3N}x_N &= b_3 \\
\cdots \quad & \quad \cdots \\
a_{M1}x_1 + a_{M2}x_2 + a_{M3}x_3 + \cdots + a_{MN}x_N &= b_M
\end{aligned}
$$

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

coefficient matrix

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ & \cdots & & \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_M \end{bmatrix}$$

Approach

- Direct methods
  - Gauss elimination
  - Gauss-Jordan elimination
  - LU decomposition
  - (Singular value decomposition)
  - …

- Iterative methods
  - Jacobi iteration
  - Gauss-Seidel iteration
  - …

Basic properties of matrices(I)

- Definition
  - element
  - row
  - column
  - row matrix, column matrix
  - square matrix
  - order= *MxN* (*M* rows, *N* columns)
  - diagonal matrix
  - identity matrix : *I*
  - upper/lower triangular matrix
  - tri-diagonal matrix
  - transposed matrix: A^t
  - symmetric matrix: $A = A^t$
  - orthogonal matrix: $A^t A = I$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & \cdots & & \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix}$$

Basic properties of matrices(II)

- Diagonal dominance

if one of the rows satisfies the following:

$$| a_{ii} | \geq \sum_{j=1, j \neq i}^{n} | a_{ij} |$$

- Transpose facts

(i) $(A^t)^t = A$.

(ii) $(A + B)^t = A^t + B^t$.

(iii) $(AB)^t = B^t A^t$.

(iv) If A^{-1} exists, $(A^{-1})^t = (A^t)^{-1}$

Basic properties of matrices(III)

- Matrix multiplication

$$\boldsymbol{AB} = [a_{ik}]_{n\times m}[b_{kj}]_{m\times r} = [c_{ij}]_{n\times r} = \boldsymbol{C}$$

$$\begin{aligned} c_{ij} &= \text{row}_i \boldsymbol{A} \ \ \text{col}_j \boldsymbol{B} \\ &= a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{im}b_{mj} \\ &= \sum_{k=1}^{m} a_{ik}b_{kj} \end{aligned}$$

Determinant

$$|\boldsymbol{A}| = \det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

$$= \sum_{j=1}^{n} a_{ij}(-1)^{i+j} M_{ij}$$

where

i : fixed

M_{ij}: minor. Determinant of $(n-1)\times(n-1)$ matrix

※ Cofactor $a_{ij} = (-1)^{i+j} M_{ij}$

Determinant facts(I)

1. Product Theorem:

$$\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$$

2. If $\boldsymbol{A}$ is upper(or lower) diagonal,

$$\det(\boldsymbol{A}) = \prod_{i=1}^{n} a_{ii}$$

3. $\det(\boldsymbol{A}^T) = \det(\boldsymbol{A})$

4. If $\boldsymbol{B}$ is obtained from $\boldsymbol{A}$ by interchanging two of its rows, then

$$\det(\boldsymbol{B}) = -\det(\boldsymbol{A})$$

Determinant facts(II)

5. For any scalar s,

$$\det(s\boldsymbol{A}) = s^n \det(\boldsymbol{A})$$

※ $\det(-\boldsymbol{A}) = (-1)^n \det(\boldsymbol{A})$

6. If $\text{row}_i(\boldsymbol{A}) = c\ \text{row}_j(\boldsymbol{A})$,
$(j \neq i,\ c$: a constant)

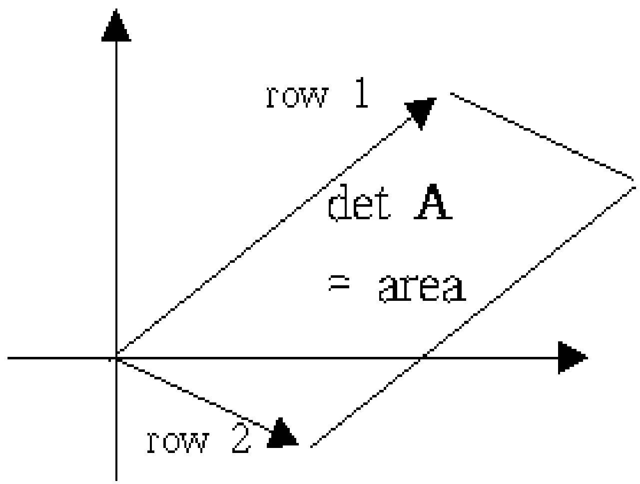
then $\det(\boldsymbol{A}) = 0$.

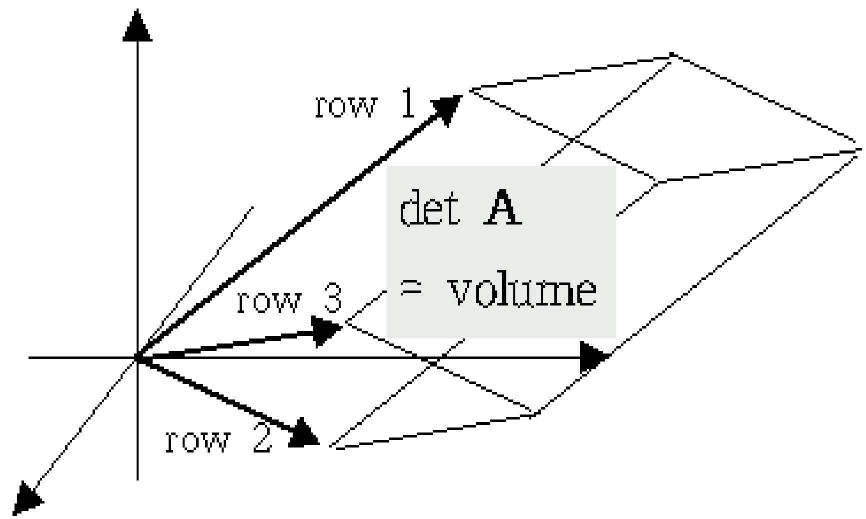
Geometrical interpretation of determinant

- Euclidean length

$$\|\text{row}_i \boldsymbol{A}\| = \sqrt{a_{i1}^2 + a_{i2}^2 + \cdots + a_{in}^2}$$

- **Volume**





- Hadamard's inequality

$$|\det \boldsymbol{A}| \le \|\text{row}_1 \boldsymbol{A}\| \; \|\text{row}_2 \boldsymbol{A}\| \cdots \|\text{row}_n \boldsymbol{A}\|$$

Over-determined/ Under-determined problem

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad \vdots \qquad\qquad &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

- Over-determined problem ($m>n$)
  - least-square estimation,
  - robust estimation etc.

- Under-determined problem ($n<m$)
  - singular value decomposition

Augmented matrix

$$[\ \boldsymbol{A} : \boldsymbol{b}\ ] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix}$$

o Elementary row operations

"without accompanying a solution change"

- multiply a constant($\neq 0$) to i-th row
- change the order of rows
- add a row multiplied by a constant($\neq 0$) to any row.

Cramer's rule

If $\det(\boldsymbol{A}) \neq 0$, then the unique solution of $\boldsymbol{Ax}=\boldsymbol{b}$ is

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{where} \quad x_j = \frac{\det(\boldsymbol{A}_j)}{\det(\boldsymbol{A})}$$

and $\boldsymbol{A}_j$ is the matrix obtained by replacing the j-th column of $\boldsymbol{A}$ by $\boldsymbol{b}$.

Triangular coefficient matrix

eg. Upper Triangle

$$
\begin{aligned}
x_1 + x_2 + 2x_3 &= -1 \\
-2x_2 + 2x_3 &= 1 \\
-4x_3 &= 4
\end{aligned}
$$

$$
[A : b] = \begin{bmatrix} 1 & 1 & 2 & : & -1 \\ 0 & -2 & 2 & : & -1 \\ 0 & 0 & -4 & : & -1 \end{bmatrix}
= \begin{bmatrix} a_{11} & a_{12} & a_{13} & : & b_1 \\ a_{21} & a_{22} & a_{23} & : & b_2 \\ a_{31} & a_{32} & a_{33} & : & b_3 \end{bmatrix}
$$

Employing **back substitution**,

$$
\begin{aligned}
x_3 &= \frac{1}{a_{33}}(b_3) &&= -1 \\
x_2 &= \frac{1}{a_{22}}(b_2 - a_{23}x_3) &&= -\frac{3}{2} \\
x_1 &= \frac{1}{a_{11}}\{b_1 - (a_{12}x_2 + a_{13}x_3)\} &&= \frac{5}{2}
\end{aligned}
$$

Substitution

Back Substitution

Upper triangular matrix

$$x_n = \frac{b_n}{a_{nn}}$$

$$x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j\right) \quad i = n-1,\ n-2,\ \cdots,2,\ 1$$

Forward Substitution

Lower triangular matrix

$$x_1 = \frac{b_1}{a_{11}}$$

$$x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j\right) \quad i = 2,\ 3,\ \cdots,\ n$$

Gauss elimination

1. **Step 1**: Gauss reduction
   - =Forward elimination
   - Coefficient matrix → upper triangular matrix

2. **Step 2**: Backward substitution

Gauss reduction

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix} \Rightarrow A' = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} & b_1^{(0)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{bmatrix}$$

Gauss reduction

Let $a_{i,n+1} = b_i$, then

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_{ik} a_{kj}^{(k-1)},$$

$$\begin{cases} i = k+1, \cdots, n \\ j = k+1, \cdots, n, n+1 \\ k = 1, 2, \cdots, n-1 \end{cases}$$

where $m_{ik} = \dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$.

Eg. Gauss elimination(I)

Example 3 Consider the linear system

$$\begin{aligned} E_1: &\quad x_1 - x_2 + 2x_3 - x_4 = -8, \\ E_2: &\quad 2x_1 - 2x_2 + 3x_3 - 3x_4 = -20, \\ E_3: &\quad x_1 + x_2 + x_3 = -2, \\ E_4: &\quad x_1 - x_2 + 4x_3 + 3x_4 = 4. \end{aligned}$$

The augmented matrix is

$$\left[\begin{array}{cccc:c} 1 & -1 & 2 & -1 & -8 \\ 2 & -2 & 3 & -3 & -20 \\ 1 & 1 & 1 & 0 & -2 \\ 1 & -1 & 4 & 3 & 4 \end{array}\right].$$

Performing the operations

$$(E_2 - 2E_1) \to (E_2), \quad (E_3 - E_1) \to (E_3), \quad \text{and} \quad (E_4 - E_1) \to (E_4),$$

we have the matrix

$$\left[\begin{array}{cccc:c} 1 & -1 & 2 & -1 & -8 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & 2 & 4 & 12 \end{array}\right].$$

Eg. Gauss elimination(II)

The element a_{22} in this matrix is zero, so the procedure cannot continue in its present form. But operations of the form $(E_i) \leftrightarrow (E_p)$ are permitted, so a search is made of the elements a_{32} and a_{42} for the first nonzero element. Since $a_{32} \neq 0$, the operation $(E_2) \leftrightarrow (E_3)$ is performed to obtain a new matrix:

$$\left[\begin{array}{rrrr:r} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 0 & 2 & 4 & 12 \end{array}\right].$$

The variable x_2 is already eliminated from E_3 and E_4, so the computations continue with the operation $(E_4 + 2E_3) \rightarrow (E_4)$, giving

$$\left[\begin{array}{rrrr:r} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 0 & 0 & 2 & 4 \end{array}\right].$$

Finally, the backward substitution is applied:

$$x_4 = \frac{4}{2} = 2, \quad x_3 = \frac{[-4 - (-1)x_4]}{-1} = 2,$$

$$x_2 = \frac{[6 - ((-1)x_3 + x_4)]}{2} = 3,$$

$$x_1 = \frac{[-8 - ((-1)x_2 + 2x_3 + (-1)x_4)]}{1} = -7.$$

Troubles in Gauss elimination

- Bad effect of round-off error in pivot coefficient

If a_{ii} is small in magnitude compared to a_{ki}, the magnitude of the multiplier

$$m_{ki} = \frac{a_{ki}}{a_{ii}}$$

will be much larger than 1. A round-off error introduced in the computation of one of the terms a_{il} is multiplied by m_{ki} when computing a_{kl}, compounding the original error.

↓

Pivoting strategy

Eg. Trouble(I)

Example 1 The linear system

$$E_1: \quad 0.003000x_1 + 59.14x_2 = 59.17,$$
$$E_2: \quad 5.291x_1 - 6.130x_2 = 46.78$$

has the solution $x_1 = 10.00$ and $x_2 = 1.000$. Suppose Gaussian elimination is performed on this system using four-digit arithmetic with rounding.

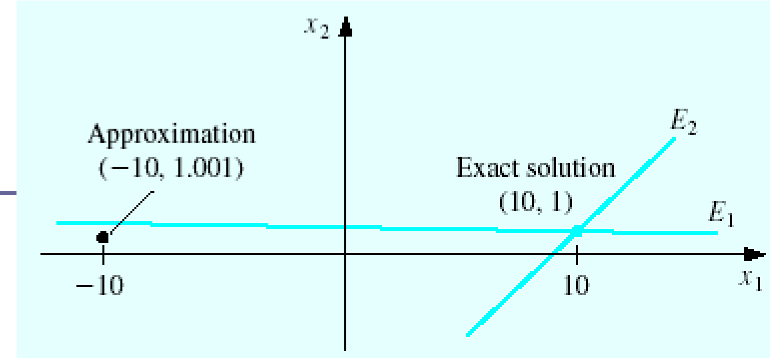
The first pivot element, $a_{11} = 0.003000$, is small, and its associated multiplier,

$$m_{21} = \frac{5.291}{0.003000} = 1763.\bar{6},$$

rounds to the large number 1764. Performing $(E_2 - m_{21}E_1) \rightarrow (E_2)$ and the appropriate rounding gives

$$0.003000x_1 + 59.14x_2 = 59.17$$
$$-104300x_2 \approx -104400$$

Eg. Trouble(II)



instead of the precise values

$$0.003000x_1 + 59.14x_2 = 59.17$$
$$-104309.37\bar{6}x_2 = -104309.37\bar{6}.$$

The disparity in the magnitudes of $m_{21}a_{13}$ and a_{23} has introduced round-off error, but the error has not yet been propagated. Backward substitution yields

$$x_2 \approx 1.001,$$

which is a close approximation to the actual value, $x_2 = 1.000$. However, because of the small pivot $a_{11} = 0.003000$,

$$x_1 \approx \frac{59.17 - (59.14)(1.001)}{0.003000} = -10.00$$

contains the small error of 0.001 multiplied by $59.14/0.003000 \approx 20000$. This ruins the approximation to the actual value $x_1 = 10.00$. (See Figure 6.1 on page 254.)

This is clearly a contrived example and the graph demonstrates why the error can so easily occur, but for only slightly larger systems it is much more difficult to predict in advance when devastating round-off error can occur.

Pivoting strategy

- To determine the smallest $p \geq i$ such that

$$|a_{pi}| = \max_{i \leq k \leq n} |a_{ki}|$$

and perform $(E_i) \leftrightarrow (E_p)$

➔ Partial pivoting

dramatic enhancement!

Effect of partial pivoting

Example 2 Reconsider the system

$$E_1: \quad 0.003000x_1 + 59.14x_2 = 59.17,$$
$$E_2: \quad 5.291x_1 - 6.130x_2 = 46.78.$$

The pivoting procedure just described results in first finding

$$\max\{|a_{11}|, |a_{21}|\} = \max\{|0.003000|, |5.291|\} = |5.291| = |a_{21}|.$$

The operation $(E_2) \leftrightarrow (E_1)$ is then performed to give the system

$$E_1: \quad 5.291x_1 - 6.130x_2 = 46.78,$$
$$E_2: \quad 0.003000x_1 + 59.14x_2 = 59.17.$$

The multiplier for this system is

$$m_{21} = \frac{a_{21}}{a_{11}} = 0.0005670,$$

and the operation $(E_2 - m_{21}E_1) \rightarrow (E_2)$ reduces the system to

$$5.291x_1 - 6.130x_2 = 46.78,$$
$$59.14x_2 \approx 59.14.$$

The four-digit answers resulting from the backward substitution are the correct values, $x_1 = 10.00$ and $x_2 = 1.000$.

Scaled partial pivoting

Although partial pivoting is sufficient for many linear systems, situations do arise when it is inadequate. For example, the linear system

$$E_1: \quad 30.00x_1 + 591400x_2 = 591700,$$
$$E_2: \quad 5.291x_1 - \quad 6.130x_2 = 46.78$$

is the same as that in Examples 1 and 2 except that all entries in the first equation have been multiplied by 10^4. Partial pivoting with four-digit arithmetic leads to the same results as obtained in Example 1 since no row interchange would be performed. A technique known as **scaled partial pivoting** is needed for this system. The first step in this procedure is to

- Scaling is to ensure that the largest element in each row has a *relative* magnitude of 1 before the comparison for row interchange is performed.

Eg. Effect of scaling

Example 3 Applying scaled partial pivoting to the system in Example 1 gives

$$s_1 = \max\{|30.00|, |591400|\} = 591400 \quad \text{and} \quad s_2 = \max\{|5.291|, |-6.130|\} = 6.130.$$

Consequently,

$$\frac{|a_{11}|}{s_1} = \frac{30.00}{591400} = 0.5073 \times 10^{-4} \quad \text{and} \quad \frac{|a_{21}|}{s_2} = \frac{5.291}{6.130} = 0.8631$$

and the interchange $(E_1) \leftrightarrow (E_2)$ is made. Applying Gaussian elimination to the new system produces the correct results: $x_1 = 10.00$ and $x_2 = 1.000$. ■

Complexity of Gauss elimination

Multiplications/divisions: $\frac{n^3}{3} + n^2 - \frac{n}{3}$.

Additions/subtractions: $\frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}$.

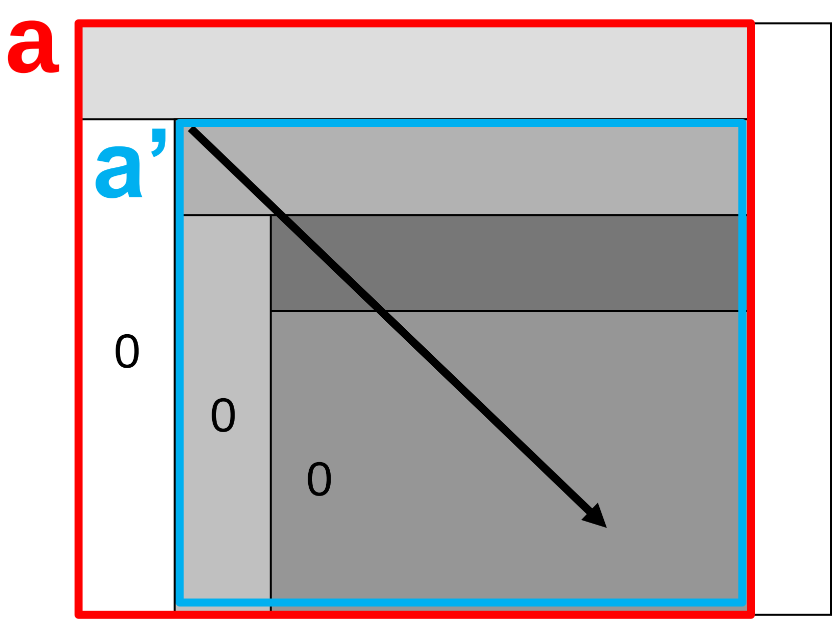
➔ Too much!

Table 6.1

n	Multiplications/Divisions	Additions/Subtractions
3	17	11
10	430	375
50	44,150	42,875
100	343,300	338,250

Summary: Gauss elimination

1) Normalization: Scaling all rows s.t. max. of a_{i*} =1, i=1,..n
2) Partial pivoting s.t. max. of a_{*1} is placed uppermost
3) Eliminating 1st element of row s.t. a_{i1} =0, i=2,..,n
4) Repeat 1)-3) for submatrix a'
5) Obtain solutions by backward substitution



Gauss-Jordan elimination

$$[\mathbf{A} \;:\; \mathbf{b}] \rightarrow [\mathbf{I} \;:\; \mathbf{x}]$$

- Eliminate until all off-diagonal elements are 0
- 50% more computation than Gauss elimination
- Good for obtaining inverse matrix

$$[\mathbf{A} \;:\; \mathbf{I}] \rightarrow [\mathbf{I} \;:\; \mathbf{A}^{-1}]$$

Eg. Obtaining inverse matrix(I)

Example 3 To determine the inverse of the matrix

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 1 & 2 \end{bmatrix},$$

let us first consider the product AB, where B is an arbitrary 3×3 matrix.

$$AB = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

$$= \begin{bmatrix} b_{11} + 2b_{21} - b_{31} & b_{12} + 2b_{22} - b_{32} & b_{13} + 2b_{23} - b_{33} \\ 2b_{11} + b_{21} & 2b_{12} + b_{22} & 2b_{13} + b_{23} \\ -b_{11} + b_{21} + 2b_{31} & -b_{12} + b_{22} + 2b_{32} & -b_{13} + b_{23} + 2b_{33} \end{bmatrix}.$$

If $B = A^{-1}$, then $AB = I$, so we must have

$$\begin{aligned} b_{11} + 2b_{21} - b_{31} &= 1, & b_{12} + 2b_{22} - b_{32} &= 0, & b_{13} + 2b_{23} - b_{33} &= 0, \\ 2b_{11} + b_{21} &= 0, & 2b_{12} + b_{22} &= 1, & 2b_{13} + b_{23} &= 0, \\ -b_{11} + b_{21} + 2b_{31} &= 0, & -b_{12} + b_{22} + 2b_{32} &= 0, & -b_{13} + b_{23} + 2b_{33} &= 1. \end{aligned}$$

Eg. Obtaining inverse matrix(II)

$$\left[\begin{array}{ccc:ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 2 & 0 & 0 & 1 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc:ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 0 & -3 & 2 & -2 & 1 & 0 \\ 0 & 3 & 1 & 1 & 0 & 1 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc:ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 0 & -3 & 2 & -2 & 1 & 0 \\ 0 & 0 & 3 & -1 & 1 & 1 \end{array}\right]$$

$$\downarrow$$

Backward substitution
For each column

$$A^{-1} = \begin{bmatrix} -\frac{2}{9} & \frac{5}{9} & -\frac{1}{9} \\ \frac{4}{9} & -\frac{1}{9} & \frac{2}{9} \\ -\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

LU decomposition

- Principle: Solving a set of linear equations based on decomposing the given coefficient matrix into a product of lower and upper triangular matrix.

$$A = LU \qquad\qquad L^{-1}$$

$$Ax = b \;\rightarrow\; LUx = b \;\rightarrow\; \cancel{L^{-1}}LUx = L^{-1}b$$

Upper triangular

$$\rightarrow\; L^{-1}b = c \;\rightarrow\; \boxed{Ux = c} \quad (1)$$

$$\downarrow L$$

Lower triangular

$$LL^{-1}b = Lc \;\rightarrow\; \boxed{Lc = b} \quad (2)$$

By solving the equations (2) and (1) successively, we get the solution x.

Various LU decompositions

- Doolittle decomposition
  - $L_{ii}=1$, for all i
- Crout decomposition
  - $U_{ii}=1$, for all i
- Cholesky decomposition
  - $L_{ii}=U_{ii}$
  - Appropriate for symmetric, positive-definite matrix

Crout decomposition

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \vdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ & \vdots & & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ & \vdots & & \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

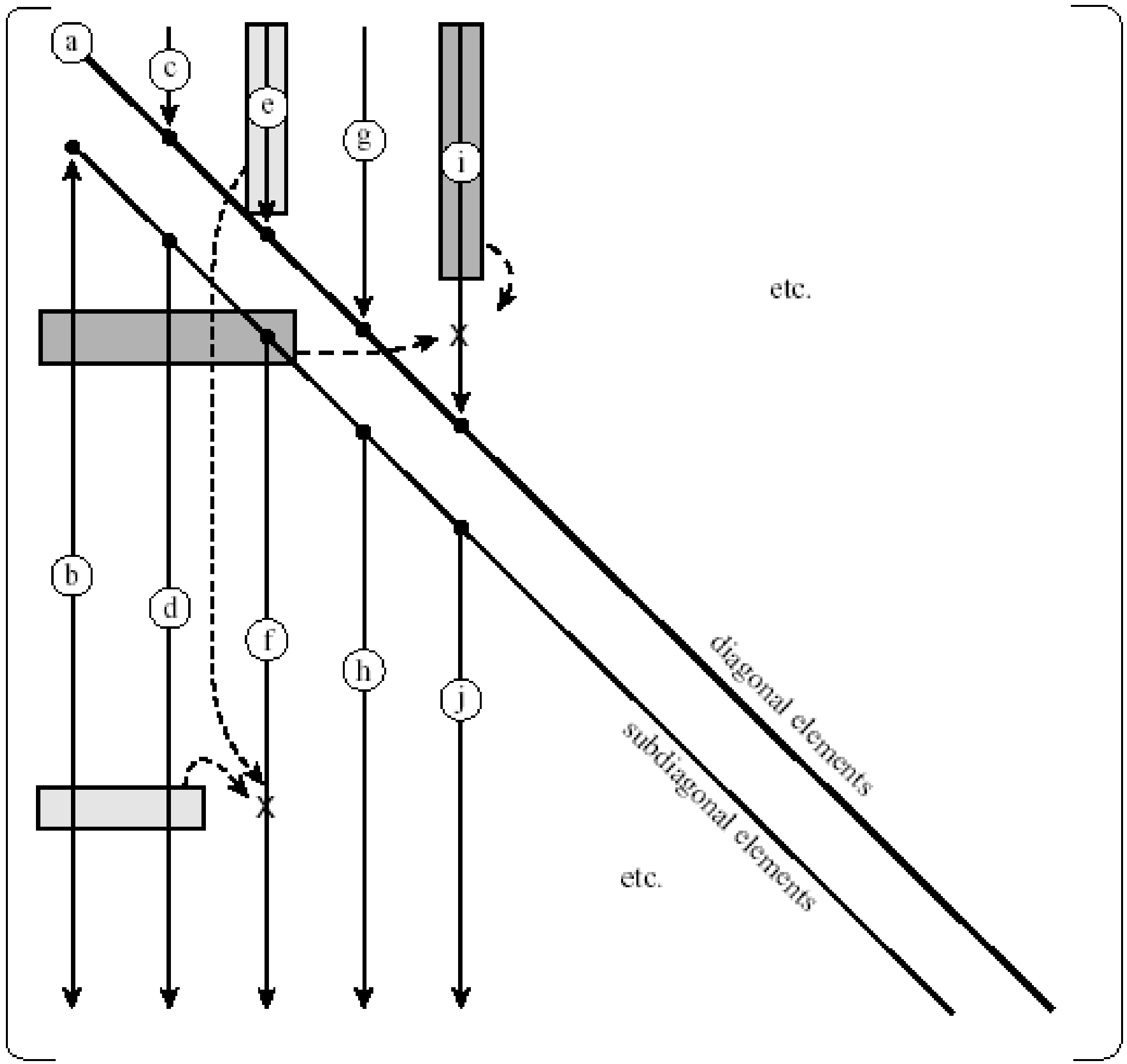
$$l_{i1} = a_{i1} \qquad (i = 1, 2, \cdots, n)$$

$$u_{1j} = \frac{a_{1j}}{l_{11}} \qquad (j = 2, 3, \cdots, n)$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \qquad (j \le i,\ \ i = 2, 3, \cdots, n)$$

$$u_{ij} = \frac{1}{l_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right] \qquad (i < j,\ \ j = 3, 4, \cdots, n)$$

Implementation of Crout method



Programming using NR in C(I)

- Solving a set of linear equations

```
float **a,*b,d;
int n,*indx;
...
ludcmp(a,n,indx,&d);
lubksb(a,n,indx,b);
```

If you subsequently want to solve a set of equations with the same **A** but a different right-hand side **b**, you repeat *only*

```
lubksb(a,n,indx,b);
```

Programming using NR in C(II)

- Obtaining inverse matrix

```
#define N ...
float **a,**y,d,*col;
int i,j,*indx;
...
ludcmp(a,N,indx,&d);                    Decompose the matrix just once.
for(j=1;j<=N;j++) {                     Find inverse by columns.
    for(i=1;i<=N;i++) col[i]=0.0;
    col[j]=1.0;
    lubksb(a,N,indx,col);
    for(i=1;i<=N;i++) y[i][j]=col[i];
}
```

Programming using NR in C(III)

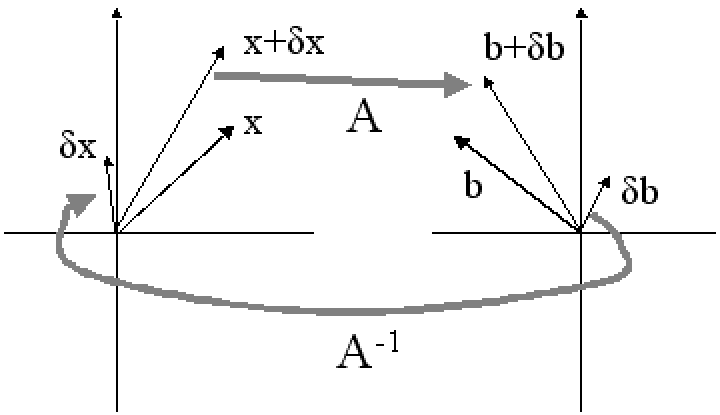
- Calculating the determinant of a matrix

```
#define N ...
float **a,d;
int j,*indx;
...
ludcmp(a,N,indx,&d);          This returns d as ±1.
for(j=1;j<=N;j++) d *= a[j][j];
```

Iterative Improvement of a Solution

- Iterative improvement
  - exact solution x
  - contaminated solution $\tilde{x} = x + \delta x$
  - wrong product $\tilde{b} = b + \delta b$
  - Algorithm derivation

$$A(x + \delta x) = b + \delta b$$



since $Ax = b$

$$A\delta x = \delta b$$

Solving the equation to get the improvement

$$\delta x = A^{-1}\delta b$$

Repeat this procedure until a convergence

Numerical Aspect of Iterative Improvement

- Numerical aspect
  - Using LU decomposition
    - Once we get a decomposition, just a substitution using δb will give the incremental correction δx
    - (Refer to mprove() in p.56 of NR in C)
  - Measure of ill-conditioning
    - A': normalized matrix
    - If A'^{-1} is large → ill-conditioned

→ Read Box 10.1

Box 10.1 Interpreting the Elements of the Matrix Inverse as a Measure of Ill-Conditioning

One method for assessing a system's condition is to scale $[A]$ so that the largest element in each row is 1 and then compute $[A]^{-1}$. If elements of $[A]^{-1}$ are several orders of magnitude greater than the elements of the original scaled matrix, it is likely that the system is ill-conditioned.

Insight into this approach can be gained by recalling that a way to check whether an approximate solution $\{X\}$ is acceptable is to substitute it into the original equations and see whether the original right-hand-side constants result. This is equivalent to

$$\{R\} = \{B\} - [A]\{\tilde{X}\} \tag{B10.1.1}$$

where $\{R\}$ is the residual between the right-hand-side constants and the values computed with the solution $\{\tilde{X}\}$. If $\{R\}$ is small, we might conclude that the $\{\tilde{X}\}$ values are adequate. However, suppose that $\{X\}$ is the exact solution that yields a zero residual, as in

$$\{0\} = \{B\} - [A]\{X\} \tag{B10.1.2}$$

Subtracting Eq. (B10.1.2) from (B10.1.1) yields

$$\{R\} = [A]\Big\{\{X\} - \{\tilde{X}\}\Big\}$$

Multiplying both sides of this equation by $[A]^{-1}$ gives

$$\{X\} - \{\tilde{X}\} = [A]^{-1}\{R\}$$

This result indicates why checking a solution by substitution can be misleading. For cases where elements of $[A]^{-1}$ are large, a small discrepancy in the right-hand-side residual $\{R\}$ could correspond to a large error $\{X\} - \{\tilde{X}\}$ in the calculated value of the unknowns. In other words, a small residual does not guarantee an accurate solution. However, we can conclude that if the largest element of $[A]^{-1}$ is on the order of magnitude of unity, the system can be considered to be well-conditioned. Conversely, if $[A]^{-1}$ includes elements much larger than unity, we conclude that the system is ill-conditioned.

Homework #3 (Cont')

1. Solve the following set of equations [Due: 10/13]

$$\boldsymbol{A}_i \boldsymbol{x}_i = \boldsymbol{b}_i, \qquad i = 1, 2, 3.$$

using

1) Gauss-Jordan Elimination(gaussj())
2) LU Decomposition(ludcmp())
3) Singular Value Decomposition(svdcmp()).

You may use the routines in the Chap.2 of Numerical Recipes in C.

The $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$'s are available from the course homepage. They are given in such a way that lineq{i}.dat contain $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$. Each file starts with orders (n n) and matrix elements($a_{11}\ a_{12} \cdots a_{21}\ a_{22} \cdots a_{nn}$) followed by vector elements($b_1\ b_2 \cdots b_n$).

Discuss empirically the advantage/disadvantage of each method.

(Cont') Homework #3

2. Apply the method of iterative improvement(mprove()) to the above problem and discuss the results.

3. Find the inverse and the determinant of the matrix A_i in the above problem.