Sol^n of Linear Equations

→ The techniques and method for solving systems of linear algebraic eq^n belongs to two fundamentally different approaches:

① Elimination Approach (Direct Method)

i) Gaussian Elimination

ii) Gaussian elimination with pivoting

iii) Gauss Jordan method

iv) LU decomposition

v) Matrix Inverse method

② Iterative Approach

It involves assumption of some initial value which are then refined repeatedly till they reach some accepted level of accuracy.

i) Gauss Seidal method

ii) Gauss Jackobi Iteration method

○ **Existense of Sol^n:**

There are 4 possibilities

i) System has a unique sol^n

ii) System has a no sol^n

iii) System has a sol^n. but not a unique one (infinite sol^n)

iv) System is ill conditioned

(i) **Unique Sol^n**

Consider the system $x + 2y = 9$, $2x - 3y = 4$

System has a sol^n $x = 5$ & $y = 2$

∵ No other pair of values of x & y would satisfy the sol^n is set to be unique.

(ii) No Solⁿ.

The eqⁿ system $2x - y = 5$, $3x - \frac{3}{2}y = 4$ have no solⁿ.

Thus, two are parallel line, they will never meet.

Such, system is called "inconsistent eqⁿ"

(iii) No unique Solⁿ:

The system of eqⁿ. $-2x + 3y = 6$, $4x - 6y = -12$ has many different solⁿ.

Since, there are two different form of same eqⁿ.

∴ they represent same line.

Such, eqⁿ. are called "dependent eqⁿ"

* Note: The system represented in fig. II & fig III is said to singular system.

(iv) Ill conditioned Solⁿ.

There maybe a situation when the system has a solⁿ but it is very close to singular. The system of eqⁿ. $x - 2y = -2$, $0.45x - 0.91y = -1$ has a solⁿ but it is difficult to identify unknowns values from such a system and those systems are called ill conditioned.

Ill conditioned systems are very sensitive to round off error and may pose ~~prob~~ problems during computation of the solⁿ.

o Basic Gaus Elimination Method:

→ Gaus elimination method proposed a systematic strategy for reducing the system of eqⁿ. to the upper triangular form using the forward elimination approach & then for obtaining values of unknowns using the back substitution process.

→ Forward Elimination Phase:

This phase is concerned with the manupulation of eq^n. in order to eliminate unknowns from system to produce upper triangular system.

→ Back substitution phase:

This phase is concerned with the actual sol^n. of the eq^n. and uses the ~~su~~ backsubstitution process on the reduced upper triangular system.

Q. Solve the system of linear eq^n. using basic gauss elimination.

$$3x_1 + 6x_2 + x_3 = 16$$
$$2x_1 + 4x_2 + 3x_3 = 13$$
$$x_1 + 3x_2 + 2x_3 = 9$$

Ans:
$$\left[\begin{array}{ccc|c} 3 & 6 & 1 & 16 \\ 2 & 4 & 3 & 13 \\ 1 & 3 & 2 & 9 \end{array}\right]$$

$$\xrightarrow[R_3 = R_3 - \frac{1}{3}R_1]{R_2 = R_2 - \frac{2}{3}R_1} \left[\begin{array}{ccc|c} 3 & 6 & 1 & 16 \\ 0 & 0 & 7/3 & 7/3 \\ 0 & 1 & 5/3 & 11/3 \end{array}\right]$$

At this point $a_{22} = 0$ the elimination procedure breaksdown. we need to reorder rows matrix.

$$\xrightarrow{R_2 \rightleftharpoons R_3} \left[\begin{array}{ccc|c} 3 & 6 & 1 & 16 \\ 0 & 1 & 5/3 & 11/3 \\ 0 & 0 & 7/3 & 7/3 \end{array}\right]$$

Using backward substitution:

$\frac{7}{3}x_3 = 7/3 \Rightarrow x_3 = 7/3$

$\therefore x_2 = 11/3 - 5/3 = \frac{6}{3} = 2$

$\therefore 3x_1 = 16 - 12 - 1 = 3$

$\Rightarrow x_1 = 1$

→ **Algorithm:**

Step 1: Arrange eqⁿs such that $a_{11} \neq 0$

Step 2: Eliminate x[1] with all eqⁿ except 1st One.

[This is done as:

a) Normalizaing the first, eqⁿ by dividing a[1][1]

b) Subtract from the second a_{21} times the normalized the first eqⁿ]

Step 3: Eliminate x[2] from 3rd to last (assume $a'_{22} \neq 0$)

[Eliminate from the 3rd eqⁿ a_{32} times to normalize]

[The process will continue till the last eqⁿ contain only one unknown]

Step 4: Obtain solⁿ by backsubstitution
[this can be substituted back $(n-1)^{th}$ eqⁿ for obtaining the solⁿ x_{n-1}
This backsubstitution will be continued till solⁿ for x_1]

Step 5: Stop

◦ Gauss Elimination with Pivoting:

→ In the basic gauss elimination method the element a_{ij} when $i=j$ is known as pivot element. Each row is normalized using by ~~defined~~ dividing the co-efficients element of that row with the pivot element.

i.e. $a_{kj} = \frac{a_{kj}}{a_{kk}}$ $j = 1, 2, \ldots, n$

→ If $a_{kk} = 0$, k^{th} row cannot be normalized (the procedure fails). One way to overcome this problem is to interchange with another row below it which doesn't have a zero element in that position.

→ In general, the reordering of eqⁿ is done to ~~help~~ improve the accuracy. Even if the pivot element is not zero.

→ Procedure of reordering:

(i) Search & locate the largest abs. value among the co-efficients in the first column.

(ii) Exchange the first row with the row containing that element.

(iii) Then eliminate the first variable with second eqⁿ

(iv) When the 2nd row becomes, search for the co-eff. for the 2nd col. from 2nd row to n^{th} row & locate the largest element. Exchange the second row ~~&~~ with the row containing the large co-eff.

(v) Continue the procedure till $(n-1)$ variables eliminate.

The above process is known as "partial pivoting".
There is an alternative scheme known as complete pivoting at each stage the largest element in the remaining row is pivot element.

compute pivotting takes a lot of overhead. thats why it is not generally used.

Q. solve using partial pivotting technique.

$2x_1 + 2x_2 + x_3 = 6$

$4x_1 + 2x_2 + 3x_3 = 4$

$x_1 - x_2 + x_3 = 0$

Ans: The original Augmented matrix:

$$\left[\begin{array}{ccc|c} 2 & 2 & 1 & 6 \\ 4 & 2 & 3 & 4 \\ 1 & -1 & 1 & 0 \end{array}\right]$$

modified original system;

$$\rightarrow \left[\begin{array}{ccc|c} \boxed{4 \quad 2 \quad 3} & & & \boxed{4} \\ 2 & 2 & 1 & 6 \\ 1 & -1 & 1 & 0 \end{array}\right] \text{(pivot)}$$

1st derived system;

$R_2 = R_2 - \frac{1}{2}R_1$
$R_3 = R_3 - \frac{1}{4}R_1$

$$\xrightarrow{} \left[\begin{array}{ccc|c} 4 & 2 & 3 & 4 \\ 0 & 1 & -1/2 & 4 \\ 0 & -3/2 & 1/4 & -1 \end{array}\right]$$

modified 1st derived system:

$$\rightarrow \left[\begin{array}{ccc|c} 4 & 2 & 3 & 4 \\ \boxed{0} & \boxed{-3/2} & \boxed{1/4} & \boxed{-1} \\ 0 & 1 & -1/2 & 4 \end{array}\right] \text{(pivot}$$

Final derived;

$$\xrightarrow{R_3 = R_3 + \frac{2}{3}R_2} \left[\begin{array}{ccc|c} 4 & 2 & 3 & 4 \\ 0 & -3/2 & 1/4 & -1 \\ 0 & 0 & -1/3 & 10/3 \end{array}\right]$$

solving using backsubstitution

$x_3 = 10$, $x_2 = -1$, $x_1 = 9$

→ Algorithm:

(S1) Input n, a_{ij}, B_i

(S2) Beginning from the first eq^n:

a) Check for the pivot element

b) If it is the largest among the elements below it, obtain the derive system.

c) Else, identify the largest element and make it the pivot element.

d) Interchange the original pivot eq^n with the one containing the largest element and so that the latter becomes the new pivot element.

e) obtain the derive system.

(S3) Repeat step 2 till the system is reduced to triangular form.

(S4) Complex x_j values by backsubstitution

(S5) Print result

(S6) stop

○ Gauss Jordan Method:

→ In gauss elimination method, a value is eliminated from the rows below the pivot element. but in gauss jordan method, it is eliminated from all other rows both below & above.

→ This process thus eliminated the all offdiagonal elements producing a diagonal matrix rather than upper triangular.

→ All rows are normalized by dividing them by their pivot elements.

→ The values of unknown can be obtains directly from the const & co-eff. matrix without employing backsubstitution.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\longrightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1' \\ b_2' \\ b_3' \end{bmatrix}$$ (result of gauss jordan)

Q $2x_1 + 4x_2 - 6x_3 = -8$

$x_1 + 3x_2 + x_3 = 10$

$2x_1 - 4x_2 - 2x_3 = -12$

Ans: Normalize the first eq^n by dividing 2

∴ The result is $x_1 + 2x_2 - 3x_3 = -4$

∴ The augmented matrix:

$$\left[\begin{array}{ccc|c} 1 & 2 & -3 & -4 \\ 1 & 3 & 1 & 10 \\ 2 & -4 & -2 & -12 \end{array}\right]$$

$$\xrightarrow[R_3 = R_3 - 2R_1]{R_2 = R_2 - R_1} \left[\begin{array}{ccc|c} 1 & 2 & -3 & -4 \\ 0 & 1 & 4 & 14 \\ 0 & -8 & 4 & -4 \end{array}\right]$$

Normalize the second eq^n

(It is already in normalized form)

$$\xrightarrow[R_2 = R_3 + 8R_2]{R_1 = R_1 - 2R_2} \left[\begin{array}{ccc|c} 1 & 0 & -11 & -32 \\ 0 & 1 & 4 & 14 \\ 0 & 0 & 36 & 108 \end{array}\right]$$

Normalize the 3rd eq^n by dividing with 36.

∴ The result: $x_3 = 3$

∴ Augmented mat:

$$\left[\begin{array}{ccc|c} 1 & 0 & -11 & -32 \\ 0 & 1 & 4 & 14 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

$$\xrightarrow[R_1 = R_1 + 11R_3]{R_2 = R_2 - 4R_3} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

$\therefore$ The values of unknown:

$x_1 = 1$ $\quad$ $x_2 = 2$ $\quad$ $x_3 = 3$

→ Algorithm:

(S1) Normalizing the first eq^n by dividing it with its pivot.

(S2) Eliminate x_1 term ~~for~~ from all the other eq^n

(S3) Normalize the second eq^n by dividing it with its pivot.

(S4) Eliminat x_2 ~~the~~ term from all the eq^n above & below of the normalized pivotal eq^n.

(S5) Repeat (S2), (S3), (S4) until x_n is eliminated from all but the last eq^n.

(S6) The resultant vector is the sol^n. vector.

(S7) stop.

* Note:

1. The gauss jordan requires only the elimination process.
2. It shows that gauss method required only $\frac{2}{3}$rd of the no. of multiplication or subtraction than gauss jordan method require.

i.e. Gauss Jordan requires 33% more multiplication & subtraction.

* Application:

- ## Jacobi Iteration Method:

It is one of the simple iterative method. For a system of eq^n. It is a direct substitution method where the values of unknown improve by substituting directly the previous value.

Let, us consider a system of n eq^n. in n unknown.

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

We rewrite the system of linear eq^n:

$$x_1 = \left[b_1 - (a_{12}x_2 + \dots + a_{1n}x_n)\right]\frac{1}{a_{11}}$$
$$x_2 = \left[b_2 - (a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n)\right]\frac{1}{a_{22}}$$
$$\vdots$$
$$x_n = \left[b_n - (a_{n1}x_1 + \dots + a_{n\,n-1}x_{n-1})\right]\frac{1}{a_{nn}}$$

Now we can compute $x_1, x_2, \dots, x_n$ by using initial guesses by the values, these new values will again use to compute the next set of x value.

The process will continue till we obtain a desired level of accuracy in the x value.

Q. Use Jacobi Iteration Method to find the values:

$$2x_1 + x_2 + x_3 = 5$$
$$3x_1 + 5x_2 + 2x_3 = 15$$
$$2x_1 + x_2 + 4x_3 = 8$$

Ans: ~~At initial, $x_1 = x_2 = x_3 = 0$~~

~~$\therefore$ we get, $(x_1^1$~~

$$x_1 = (5 - x_2 - x_3)\frac{1}{2}$$
$$x_2 = (15 - 3x_1 - 2x_3)\frac{1}{5}$$
$$x_3 = (8 - 2x_1 - x_2)\,1/4$$

At initial stage, $x_1 = x_2 = x_3 = 0$

$\therefore (x_1)^1 = (5-0-0)\frac{1}{2} = 2.5$

$(x_2)^1 = (15 - 3\times 0 - 2\times 0)\frac{1}{5} = 3$

$(x_3)^1 = (8 - 2\times 0 - 0)\frac{1}{4} = 2$

Iteration 2:

$(x_1)^2 = (5-3-2)\frac{1}{2} = 0$

$(x_2)^2 = (15 - 3\times 2.5 - 2\times 2)\frac{1}{5} = 0.7$

$(x_3)^2 = (8 - 2\times 2.5 - 3)\frac{1}{4} = 0$

Iteration 3:

$(x_1)^3 = (5-0.7-0)\frac{1}{2} = 2.15$

$(x_2)^3 = \frac{1}{5}(15 - 3\times 0 - 2\times 0) = 3$

$(x_3)^3 = (8 - 2\times 0 - 0.7)\frac{1}{4} = 1.825$

Iteration 4:

$(x_1)^4 = (5-3-1.825)\frac{1}{2} = 0.0875$

$(x_2)^4 = (15 - 3\times 2.15 - 2\times 1.825)\frac{1}{5} = 0.98$

$(x_3)^4 = (8 - 2\times \overset{2.15}{\cancel{0.0875}} - 3)\frac{1}{4} = 0.175$

Iteration 5:

$(x_1)^5 = (5 - 0.98 - 0.175)\frac{1}{2} = 1.9225$

$(x_2)^5 = (15 - 3\times 0.0875 - 2\times 0.175)\frac{1}{5} = 2.8775$

$(x_3)^5 = (8 - 2\times 0.0875 - 0.98)\frac{1}{4} = 1.71125$

Iteration 6:

$(x_1)^6 = (5 - 2.8775 - 1.71125)\frac{1}{2} = 0.205625$

$(x_2)^6 = (15 - 3\times 1.9225 - 2\times 1.71125)\frac{1}{5} = 1.162$

$(x_3)^6 = (8 - 2\times 1.9225 - 2.8775)\frac{1}{4} = 0.319375$

→ Algorithm of Jacobi Iteration method:

(S1) Obtain n, a_{ij}, b_i values

(S2) set, $x_{0,i} = b_i / a_{ii}$ (for $i = 1$ to n)

(S3) set key = 0

(S4) for $i = 1, 2, \dots, n$

a) set sum = b_i

b) for $j = 1, 2, \dots, n$ $(j \neq i)$

set sum $-= a_{ij}\, x_j$

repeat j

c) set $x_i = \text{sum}/a_{ii}$

d) if key = 0 then

if $\left| \dfrac{x_i - x_{0,i}}{x_i} \right| >$ error!

key = 1

(S5) If key = 0 then set $x_{0,i} = x_i$

go to (S3)

(S6) Write Result

(S7) Stop

A Note on Iterative Methods:-

convergence of iteration methods:

A sufficient condition for convergence is that for each row the abs. values of the diagonal element should be greater than the sum of abs. values of the other eq^n of the system

↳ It is called diagonal dominant system.

◦ Gauss Seidal Method:

→ Gauss seidal method is a improved of jacobi iteration method.

→ In jacobi method we begin with the initial values $x_1 = x_2 = \dots = x_n = 0$ & obtain next appx. $x_1', x_2', \dots, x_n'$

→ In computing x_2' we used x_1 & not x_1^0 which just not been computed. Since both x_1' & x_1^0 are available we can use x_1' which is better appx. for computation for x_2'. — This idea can be extended to all subsequent computation & the approach is called Gauss Seidal Method.

→ The gauss seidal method uses the most current value of x_i as soon as available at any point of iteration process.

Q. $2x_1 + x_2 + x_3 = 5$
$3x_1 + 5x_2 + 2x_3 = 15$
$2x_1 + x_2 + 4x_3 = 8$
solve using Gauss Seidal Method.

Ans: $x_1 = [5 - x_2 - x_3]\,1/2$
$x_2 = [15 - 3x_1 - 2x_3]\,1/5$
$x_3 = [8 - 2x_1 - x_2]\,1/4$

Assuming $x_1 = 0,\ x_2 = 0,\ x_3 = 0$

[For all values rounded to 1 decimal point]

Iteration 1:

$x_1' = 5/2 = 2.5$

$x_2' = [15 - 3\times 2.5 - 2\times 0]\frac{1}{5} = 1.5$

$x_3' = [8 - 2\times 2.5 - 1.5]\frac{1}{4} = 0.4$

Iteration 2:

$x_1^2 = [5 - 1.5 - 0.4]\frac{1}{2} = 1.6$

$x_2^2 = 1.9$

$x_3' = [8 - 2\times 1.6 - 1.9]\frac{1}{4} = 0.7$

Iteration 3:

$x_1^3 = [5 - 1.9 - 0.7]\frac{1}{2} = 1.4$

$x_2^3 = [15 - 3\times 1.4 - 2\times 0.7]\frac{1}{5} = 1.9$

$x_3^3 = [8 - 2\times 1.4 - 1.9]\frac{1}{4} = 0.8$

Iteration 4:

$x_1^4 = [5 - 1.9 - 0.8] \frac{1}{2} = 1.1$

$x_2^4 = [15 - 3 \times 1.1 - 2 \times 0.8] \frac{1}{5} = 2.0$

$x_3^4 = [8 - 2 \times 1.1 - 2.0] \frac{1}{4} = 0.9$

Iteration 5:

$x_1^5 = [5 - 2.0 - 0.9] \frac{1}{2} = 1.0$

$x_2^5 = [15 - 3 \times 1.0 - 0.9 \times 2] \frac{1}{5} = 2.0$

$x_3^5 = [8 - 2 \times 1.0 - 2.0] \frac{1}{4} = 1.0$

Iteration 6:

$x_1^6 = 1.0$

$x_2^6 = 2.0$

$x_3^6 = 1.0$

$\therefore x_1 = 1.0 \quad x_2 = 2.0 \quad x_3 = 1.0$

Q. Solve using gauss seidal:

$3x_1 + x_2 = 5$

$x_1 - 3x_2 = 5$

Ans: $x_1 = [5 - x_2] \frac{1}{3} \qquad x_2 = [x_1 - 5] \frac{1}{3}$

$x_1^0 = x_2^0 = 0.0000$

All values will be taken upto 4 decimal points.

$x_1^1 = 5/3 = 1.6667$

$x_2^1 = -1.1111$

$x_1^2 = 2.0370$

$x_2^2 = -0.9877$

$x_1^3 = 1.9959$

$x_2^3 = -1.0014$

$x_1^4 = 2.0005$

$x_2^4 = -0.9998$

$x_1^5 = 1.9999$

$x_2^5 = -1.0000$

$x_1^6 = 2.0000$

$x_2^6 = -1.0000$

$x_1^7 = 2.0000$

$x_2^7 = -1.0000$

$\boxed{\therefore x_1 = 2 \quad x_2 = -1}$

The process convergence to the solⁿ ($x_1 = 2$ $x_2 = -1$) in 6th iteration. The given system is diagonally dominant. Convergence of such system are garunteed.

Q.

$$x_1 - 3x_2 = 5$$
$$3x_1 + x_2 = 5$$

solve using gauss-seidal method.

Ans: $x_1 = [5 + 3x_2]$ $\quad$ $x_2 = [5 - 3x_1]$

$x_1^0 = 0 = x_2^0$

$x_1^1 = 5$

$x_2^1 = 5 - 15 = -10$

$x_1^2 = 5 - 30 = -25$

$x_2^2 = 5 + 75 = 80$

$x_1^3 = 5 + 240 = 245$

$x_2^3 = 5 - 735 = -730$

※ Note:

① The gauss seidel method, the prev. value is no longer needed, and therefore the prev. value can be replaced by new one — we need to store the current value of x

② The system of eqn. that satisfies the diagonally dominant system convergence of such system as gaunteed.

→ Algorithm:

(S1) Obtain n, a_{ij}, b_i values.

(S2) Set $a_{ii} = b_i / a_{ii}$ ($i = 1$ to n)

(S3) set key=0

(S4) for $i = 1, 2, \dots, n$

a) set sum = b_i

b) for $j = 1, 2, \dots, n$ ($j \neq i$)

set sum -= $a_{ij} x_j$

repeat j

c) dummy = sum / a_{ii}

d) if key = 0 then

if $\left| \frac{\text{dummy} - x_i}{\text{dummy}} \right| >$ error

set key=1

e) set x_i = dummy

repeat i loop

(S5) if key = 1

go to (S3)

(S6) write result

(S7) stop