~ Linear Algebraic Equations ~

Gauss Elimination

Chapter 9

Credit: Prof. Lale Yurttas, Chemical Eng., Texas A&M University

Solving Systems of Equations

- A linear equation in n variables:

$$a_1x_1 + a_2x_2 + \ \dots + a_nx_n = b$$

- For small ($n \leq 3$), linear algebra provides several tools to solve such systems of linear equations:

  - Graphical method
  - Cramer's rule
  - Method of elimination

- Nowadays, easy access to computers makes the solution of **very large** sets of linear algebraic equations possible

Determinants and Cramer's Rule

$$[\boldsymbol{A}]\{\boldsymbol{x}\} = \{\boldsymbol{b}\}$$

[A] : ***coefficient matrix***

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

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

D : Determinant of A matrix

Computing the Determinant

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$[A]\{x\} = \{B\}$$

$$D_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}\,a_{33} - a_{32}\,a_{23}$$

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$D_{12} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = a_{21}\,a_{33} - a_{31}\,a_{23}$$

$$D_{13} = \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = a_{21}\,a_{32} - a_{31}\,a_{22}$$

$$\text{Determinan t of A} = D = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Gauss Elimination

- Solve $Ax = b$

- Consists of two phases:
  - **–Forward elimination**
  - **–Back substitution**

- *Forward Elimination* reduces $Ax = b$ to an upper triangular system $Tx = b'$

- *Back substitution* can then solve $Tx = b'$ for x

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & 0 & a''_{33} & b''_3 \end{array}\right]$$

Forward Elimination

$$\Downarrow$$

$$x_3 = \frac{b''_3}{a''_{33}} \qquad x_2 = \frac{b'_2 - a'_{23}x_3}{a'_{22}}$$

$$x_1 = \frac{b_1 - a_{13}x_3 - a_{12}x_2}{a_{11}}$$

Back Substitution

Forward Elimination

-(3/1)

-(2/1)

$$\begin{aligned} x_1 - x_2 + x_3 &= 6 \\ 3x_1 + 4x_2 + 2x_3 &= 9 \\ 2x_1 + x_2 + x_3 &= 7 \end{aligned} \Rightarrow \begin{aligned} x_1 - x_2 + x_3 &= 6 \\ 0 + 7x_2 - x_3 &= -9 \\ 0 + 3x_2 - x_3 &= -5 \end{aligned}$$

-(3/7)

$$\Rightarrow \begin{aligned} x_1 - x_2 + x_3 &= 6 \\ 0 \quad 7x_2 - x_3 &= -9 \\ 0 \quad 0 \; -(4/7)x_3 &= -(8/7) \end{aligned}$$

**here Solve using BACK SUBSTITUTION: $x_3 = 2$ $x_2 = -1$ $x_1 = 3$

0								
0	0							
0	0	0						
0	0	0	0					
0	0	0	0	0				
0	0	0	0	0	0			
0	0	0	0	0	0	0		
0	0	0	0	0	0	0	0	

Back Substitution

$$
\begin{array}{lllllll}
1x_0 & +1x_1 & -1x_2 & +4x_3 & = & 8 \\
 & -2x_1 & -3x_2 & +1x_3 & = & 5 \\
 & & 2x_2 & -3x_3 & = & 0 \\
 & & & 2x_3 & = & 4
\end{array}
$$

$\boldsymbol{x_3 = 2}$

Back Substitution

$$1x_0 \quad +1x_1 \quad -1x_2 \quad = \quad 0$$

$$-2x_1 \quad -3x_2 \quad = \quad 3$$

$$2x_2 \quad = \quad 6$$

$\boldsymbol{x_2 = 3}$

Back Substitution

$1x_0$	$+1x_1$	$=$	3
$\boxed{x_1 = -6}$	$-\,2x_1$	$=$	12

Back Substitution

$x_0 = 9$

$$1x_0 = 9$$

Back Substitution
(* Pseudocode *)

for $i \leftarrow n$ down to 1 **do**

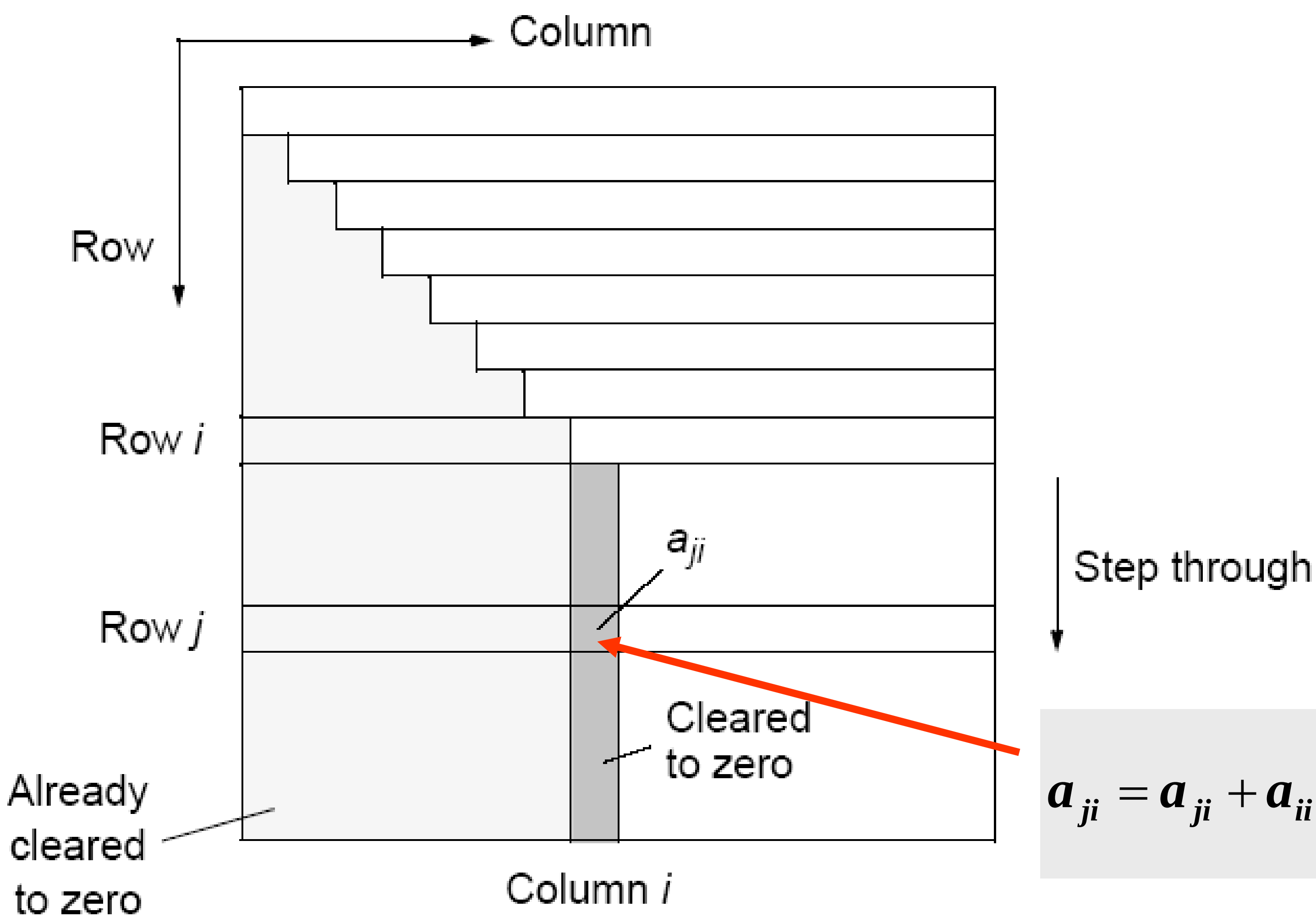
/* calculate x_i */
$x[i] \leftarrow b[i] / a[i, i]$

/* substitute $x[i]$ in the equations above */
for $j \leftarrow 1$ to $i-1$ **do**
$b[j] \leftarrow b[j] - x[i] \times a[j, i]$
endfor

endfor

Time Complexity? ➔➔ ***$O(n^2)$***

Forward Elimination



$$a_{ji} = a_{ji} + a_{ii}\left(\frac{-a_{ji}}{a_{ii}}\right) = 0$$

Forward Elimination

M
U
L
T
I
P
L
I
E
R
S

	$4x_0$	$+6x_1$	$+2x_2$	$-2x_3$	$=$	8
$-(2/4)$	$2x_0$		$+5x_2$	$-2x_3$	$=$	4
$-(-4/4)$	$-4x_0$	$-3x_1$	$-5x_2$	$+4x_3$	$=$	1
$-(8/4)$	$8x_0$	$+18x_1$	$-2x_2$	$+3x_3$	$=$	40

Forward Elimination

MULTIPLIERS

$$4x_0 \quad +6x_1 \quad +2x_2 \quad -2x_3 \quad = \quad 8$$

$$-3x_1 \quad +4x_2 \quad -1x_3 \quad = \quad 0$$

-(3/-3) $$+3x_1 \quad -3x_2 \quad +2x_3 \quad = \quad 9$$

-(6/-3) $$+6x_1 \quad -6x_2 \quad +7x_3 \quad = \quad 24$$

Forward Elimination

$$\begin{array}{cccccc} 4x_0 & +6x_1 & +2x_2 & -2x_3 & = & 8 \\ & -3x_1 & +4x_2 & -1x_3 & = & 0 \\ & & 1x_2 & +1x_3 & = & 9 \\ & & 2x_2 & +5x_3 & = & 24 \end{array}$$

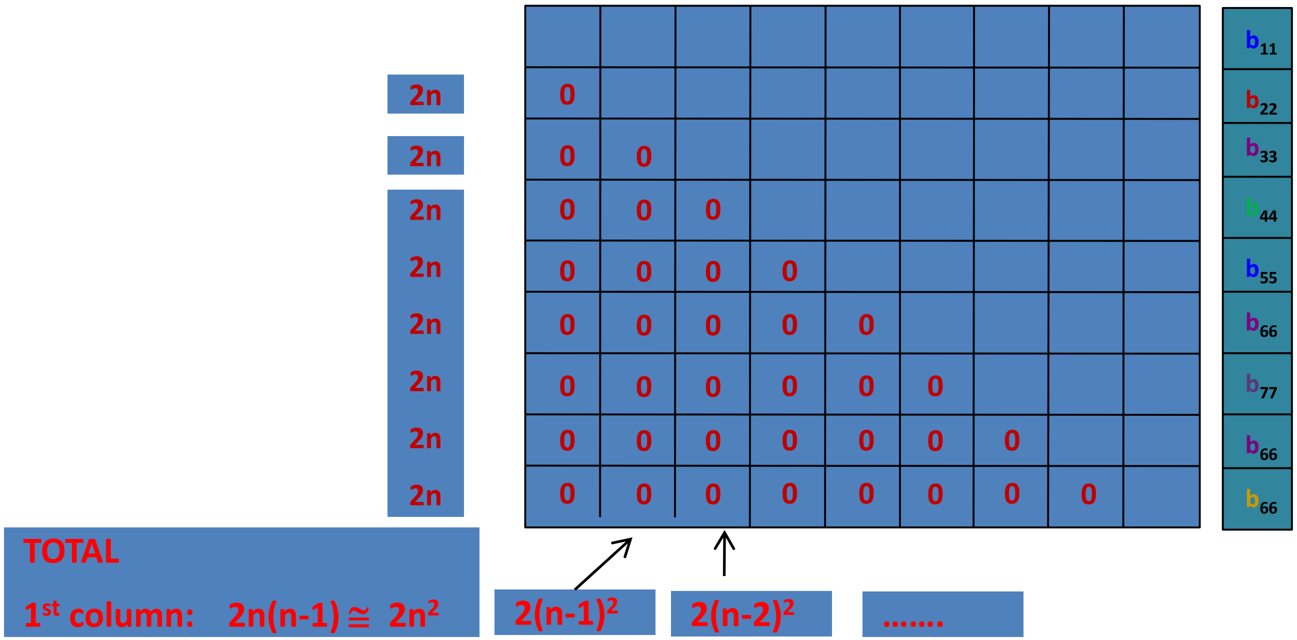
MULTIPLIER

??

Forward Elimination

$$
\begin{array}{cccccc}
4x_0 & +6x_1 & +2x_2 & -2x_3 & = & 8 \\
 & -3x_1 & +4x_2 & -1x_3 & = & 0 \\
 & & 1x_2 & +1x_3 & = & 9 \\
 & & & 3x_3 & = & 6
\end{array}
$$

Operation count in Forward Elimination



TOTAL # of Operations for FORWARDELIMINATION :

$$2n^2 + 2(n-1)^2 + \ldots + 2*(2)^2 + 2*(1)^2 = 2\sum_{i=1}^{n} i^2$$

$$= 2\frac{n(n+1)(2n+1)}{6}$$

$$= O(n^3)$$

Pitfalls of Elimination Methods

Division by zero

It is possible that during both elimination and back-substitution phases a division by zero can occur.

For example:

$$
\begin{aligned}
2x_2 + 3x_3 &= 8 \\
4x_1 + 6x_2 + 7x_3 &= -3 \\
2x_1 + x_2 + 6x_3 &= 5
\end{aligned}
\qquad
A = \begin{matrix} 0 & 2 & 3 \\ 4 & 6 & 7 \\ 2 & 1 & 6 \end{matrix}
$$

Solution: ***pivoting*** (to be discussed later)

Pitfalls (cont.)

Round-off errors

- Because computers carry only a limited number of significant figures, round-off errors will occur and they will *propagate* from one iteration to the next.

- This problem is especially important when **large** numbers of equations (100 or more) are to be solved.

- Always use **double-precision** numbers/arithmetic. It is slow but needed for correctness!

- It is also a good idea to substitute your results back into the original equations and check whether a substantial error has occurred.

Pitfalls (cont.)

ill-conditioned systems **-** small changes in coefficients result in large changes in the solution. Alternatively, a wide range of answers can approximately satisfy the equations.

(*Well-conditioned systems* – small changes in coefficients result in small changes in the solution)

Problem: Since *round off errors* can induce small changes in the coefficients, these changes can lead to large solution errors in *ill-conditioned* systems.

Example:

$$x_1 + 2x_2 = 10$$
$$\mathbf{1.1}x_1 + 2x_2 = 10.4$$

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{D} = \frac{\begin{vmatrix} 10 & 2 \\ 10.4 & 2 \end{vmatrix}}{1(2) - 2(1.1)} = \frac{2(10) - 2(10.4)}{-0.2} = 4 \qquad x_2 = 3$$

$$x_1 + 2x_2 = 10$$
$$\mathbf{1.05}x_1 + 2x_2 = 10.4$$

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{D} = \frac{\begin{vmatrix} 10 & 2 \\ 10.4 & 2 \end{vmatrix}}{1(2) - 2(1.05)} = \frac{2(10) - 2(10.4)}{-0.1} = 8 \qquad x_2 = 1$$

ill-conditioned systems (cont.) –

- Surprisingly, **substitution** of the erroneous values, x_1=8 and x_2=1, into the original equation **will not** reveal their incorrect nature clearly:

$x_1 + 2x_2 = 10$ $\quad$ $8+2(1) = 10$ $\quad$ (the same!)

$1.1x_1 + 2x_2 = 10.4$ $\quad$ $1.1(8)+2(1)=10.8$ $\quad$ (close!)

IMPORTANT OBSERVATION:
An ill-conditioned system is one with a ***determinant*** close to **zero**

- If determinant D=0 then there are infinitely many solutions ➔ ***singular system***

- ***Scaling*** (multiplying the coefficients with the same value) does not change the equations but changes the value of the determinant in a significant way.
  However, it does not change the *ill-conditioned* state of the equations!
  DANGER! It may hide the fact that the system is ill-conditioned!!

How can we find out whether a system is *ill-conditioned* or not?
Not easy! Luckily, most engineering systems yield well-conditioned results!

- One way to find out: change the coefficients slightly and recompute & compare

Techniques for Improving Solutions

- Use of more significant figures – ***double precision*** *arithmetic*

- ***Pivoting***

  If a pivot element is zero, normalization step leads to division by zero. The same problem may arise, when the pivot element is close to zero. Problem can be avoided:

  – ***Partial pivoting***

  Switching the rows below so that the largest element is the pivot element.

  **here* Go over the solution in: CHAP9e-Problem-11.doc

  – ***Complete pivoting***

  - Searching for the largest element in all rows and columns then switching.
  - This is rarely used because switching columns changes the order of x's and adds significant complexity and overhead ➔ costly

- ***Scaling***

  – used to reduce the round-off errors and improve accuracy

Gauss-Jordan Elimination

a_{11}	0	0	0	0	0	0	0	0	b_{11}
0	x_{22}	0	0	0	0	0	0	0	b_{22}
0	0	x_{33}	0	0	0	0	0	0	b_{33}
0	0	0	x_{44}	0	0	0	0	0	b_{44}
0	0	0	0	x_{55}	0	0	0	0	b_{55}
0	0	0	0	0	x_{66}	0	0	0	b_{66}
0	0	0	0	0	0	x_{77}	0	0	b_{77}
0	0	0	0	0	0	0	x_{88}	0	b_{88}
0	0	0	0	0	0	0	0	x_{99}	b_{99}

Gauss-Jordan Elimination: **Example**

$$\begin{bmatrix} 1 & 1 & 2 \\ -1 & -2 & 3 \\ 3 & 7 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 10 \end{bmatrix} \qquad \text{Augmented Matrix}: \begin{bmatrix} 1 & 1 & 2 & | & 8 \\ -1 & -2 & 3 & | & 1 \\ 3 & 7 & 4 & | & 10 \end{bmatrix}$$

R2 ← R2 - (-1)R1

R3 ← R3 - (3)R1

$$\begin{bmatrix} 1 & 1 & 2 & | & 8 \\ 0 & -1 & 5 & | & 9 \\ 0 & 4 & -2 & | & -14 \end{bmatrix}$$

Scaling R2:

R2 ← R2/(**-1**)

$$\begin{bmatrix} 1 & 1 & 2 & | & 8 \\ 0 & 1 & -5 & | & -9 \\ 0 & 4 & -2 & | & -14 \end{bmatrix}$$

R1 ← R1 - (1)R2

R3 ← R3-(4)R2

$$\begin{bmatrix} 1 & 0 & 7 & | & 17 \\ 0 & 1 & -5 & | & -9 \\ 0 & 0 & 18 & | & 22 \end{bmatrix}$$

Scaling R3:

R3 ← R3/(**18**)

$$\begin{bmatrix} 1 & 0 & 7 & | & 17 \\ 0 & 1 & -5 & | & -9 \\ 0 & 0 & 1 & | & 11/9 \end{bmatrix}$$

R1 ← R1 - (7)R3

R2 ← R2-(-5)R3

$$\begin{bmatrix} 1 & 0 & 0 & | & 8.444 \\ 0 & 1 & 0 & | & -2.888 \\ 0 & 0 & 1 & | & 1.222 \end{bmatrix}$$

RESULT:

x_1=8.45, ***x_2=-2.89,*** ***x_3=1.23***

Time Complexity? ➔➔ ***$O(n^3)$***

Systems of Nonlinear Equations

- Locate the roots of a set of simultaneous nonlinear equations:

$$f_1(x_1, x_2, x_3, \ldots, x_n) = 0$$

$$f_2(x_1, x_2, x_3, \ldots, x_n) = 0$$

$$\vdots$$

$$f_n(x_1, x_2, x_3, \ldots, x_n) = 0$$

Example:

$$x_1^{\,2} + x_1 x_2 = 10 \qquad \Rightarrow \qquad \mathrm{f}_1(x_1, x_2) = x_1^{\,2} + x_1 x_2 - 10 = 0$$

$$x_2 + 3x_1 x_2^{\,2} = 57 \qquad \Rightarrow \qquad \mathrm{f}_2(x_1, x_2) = x_2 + 3x_1 x_2^{\,2} - 57 = 0$$

- **First-order** Taylor series expansion of a function with more than one variable:

$$f_{1(i+1)} = f_{1(i)} + \frac{\partial f_{1(i)}}{\partial x_1}(x_{1(i+1)} - x_{1(i)}) + \frac{\partial f_{1(i)}}{\partial x_2}(x_{2(i+1)} - x_{2(i)}) = 0$$

$$f_{2(i+1)} = f_{2(i)} + \frac{\partial f_{2(i)}}{\partial x_1}(x_{1(i+1)} - x_{1(i)}) + \frac{\partial f_{2(i)}}{\partial x_2}(x_{2(i+1)} - x_{2(i)}) = 0$$

- The root of the equation occurs at the value of x_1 and x_2 where $f_{1(i+1)}$ =0 and $f_{2(i+1)}$ =0
  Rearrange to solve for $x_{1(i+1)}$ and $x_{2(i+1)}$

$$\frac{\partial f_{1(i)}}{\partial x_1} x_{1(i+1)} + \frac{\partial f_{1(i)}}{\partial x_2} x_{2(i+1)} = -f_{1(i)} + \frac{\partial f_{1(i)}}{\partial x_1} x_i + \frac{\partial f_{1(i)}}{\partial x_2} x_{2(i)}$$

$$\frac{\partial f_{2(i)}}{\partial x_1} x_{1(i+1)} + \frac{\partial f_{2(i)}}{\partial x_2} x_{2(i+1)} = -f_{2(i)} + \frac{\partial f_{2(i)}}{\partial x_1} x_i + \frac{\partial f_{2(i)}}{\partial x_2} x_{2(i)}$$

$$\begin{bmatrix} \frac{\partial f_{1(i)}}{\partial x_1} & \frac{\partial f_{1(i)}}{\partial x_2} \\ \frac{\partial f_{2(i)}}{\partial x_1} & \frac{\partial f_{2(i)}}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_{1(i+1)} \\ x_{2(i+1)} \end{bmatrix} = -\begin{bmatrix} f_{1(i)} \\ f_{2(i)} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_{1(i)}}{\partial x_1} & \frac{\partial f_{1(i)}}{\partial x_2} \\ \frac{\partial f_{2(i)}}{\partial x_1} & \frac{\partial f_{2(i)}}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_{1(i)} \\ x_{2(i)} \end{bmatrix}$$

- Since $x_{1(i)}$, $x_{2(i)}$, $f_{1(i)}$, and $f_{2(i)}$ are all known at the i^{th} iteration, this represents a set of two linear equations with two unknowns, $x_{1(i+1)}$ and $x_{2(i+1)}$
- You may use several techniques to solve these equations

General Representation for the solution of Nonlinear Systems of Equations

$$f_1(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$f_2(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$\vdots$$
$$f_n(x_1, x_2, x_3, \ldots, x_n) = 0$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

← **Jacobian Matrix**

Solve this set of linear equations at each iteration:

$$[J_i]\{X_{i+1}\} = -\{F_i\} + [J_i]\{X_i\}$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_i \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{Bmatrix}_{i+1} = -\begin{Bmatrix} f_1(x_1, x_2, x_3, \ldots, x_n) \\ f_2(x_1, x_2, x_3, \ldots, x_n) \\ \vdots \\ \vdots \\ f_n(x_1, x_2, x_3, \ldots, x_n) \end{Bmatrix}_i + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_i \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{Bmatrix}_i$$

$\{X\}_{(i+1)}$ values are found by solving this system of linear equations.
These iterations will continue until the convergence is reached;
which will happen when the **approximate relative error** (defined on the vector of values) falls below the desired error tolerance ε_{tol}

Solution of Nonlinear Systems of Equations

Solve this set of linear equations at each iteration: $[J_i]\{X_{i+1}\} = -\{F_i\} + [J_i]\{X_i\}$

Rearrange:

$$[J_i]\{X_{i+1} - X_i\} = -\{F_i\}$$

$$[J_i]\{\Delta X_{i+1}\} = -\{F_i\}$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_i \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \vdots \\ \Delta x_n \end{Bmatrix}_{i+1} = -\begin{Bmatrix} f_1(x_1, x_2, x_3, \ldots, x_n) \\ f_2(x_1, x_2, x_3, \ldots, x_n) \\ \vdots \\ \vdots \\ f_n(x_1, x_2, x_3, \ldots, x_n) \end{Bmatrix}_{i\,i}$$

Notes on the solution of Nonlinear Systems of Equations

***n* non-linear equations**
***n* unknowns to be solved:**

$$f_1(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$f_2(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$\vdots$$
$$f_n(x_1, x_2, x_3, \ldots, x_n) = 0$$

Jacobian Matrix:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

- Need to solve this set of Linear Equations

$$[J_i]\{\Delta X_{i+1}\} = -\{F_i\}$$

in each and every iterative solution of the original Non-Linear system of equations.

- For *n* unknowns, the size of the *Jacobian* matrix is n^2 and the time it takes to solve the ***linear system of equations*** (one iteration of the non-linear system) is proportional to ***O(n³)***
- There is no guarantee for the convergence of the non-linear system. There may also be slow convergence.
- In summary, finding the solution set for the Non-Linear system of equations is an extremely compute-intensive task.