ME 310

Numerical Methods

Solving Systems of Linear Algebraic Equations

These presentations are prepared by

Dr. Cuneyt Sert

Mechanical Engineering Department

Middle East Technical University

Ankara, Turkey

csert@metu.edu.tr

They can not be used without the permission of the author

Math Introduction

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\ldots \\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

A general set of equations.
n equations, n unknowns.

Linear Algebraic Equation: An equation of the form f(x)=0 where f is a polynomial with linear terms.

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
\ldots \quad \ldots \quad \ldots \quad \ldots & \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n
\end{aligned}
$$

A general set of linear algebraic equations.
n equations, n unknowns.

Matrix Form: [A] {x} = {b}

[A] nxn Coefficient matrix
{x} nx1 Unknown vector
{b} nx1 Right-Hand-Side (RHS) vector

Review of Matrices

$$[A]=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}_{n\times m}$$

$\leftarrow$ 2nd row

m^{th} column

Elements are indicated by a_{ij}

row column

Row vector: $[R]=\begin{bmatrix} r_1 & r_2 & \cdots & r_n \end{bmatrix}_{1\times n}$

Column vector: $[C]=\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}_{m\times 1}$

Square matrix: $[A]_{nxm}$ is a square matrix if n=m. A system of n equations with n unknonws has a square coefficient matrix.

Main (principle) diagonal: of $[A]_{nxn}$ consists of elements a_{ii} ; i=1,...,n

Symmetric matrix: If $a_{ij} = a_{ji} \rightarrow [A]_{nxn}$ is a symmetric matrix

Diagonal matrix: $[A]_{nxn}$ is diagonal if $a_{ij} = 0$ for all i=1,...,n ; j=1,...,n and $i \neq j$

Identity matrix: $[A]_{nxn}$ is an identity matrix if it is diagonal with $a_{ii}=1$ i=1,...,n . Shown as [I]

Review of Matrices (cont'd)

Upper triangular matrix: $[A]_{nxn}$ is upper triangular if $a_{ij}=0$ i=1,...,n ; j=1,...,n and i>j

Lower triangular matrix: $[A]_{nxn}$ is lower triangular if $a_{ij}=0$ i=1,...,n ; j=1,...,n and i<j

Banded matrix: $[A]_{nxn}$ is a banded matrix if $a_{ij}=0$ for all $j - i > HBW$ or $i - j > HBW$, where HBW is the half bandwidth.

Tridiagonal matrix: is a banded matrix with HBW=1.

Inverse of a matrix: $[A]^{-1}$ is the inverse of $[A]_{nxn}$ if $[A]^{-1}[A]=[I]$

Transpose of a matrix: [B] is the transpose of $[A]_{nxn}$ if $b_{ij}=a_{ji}$ Shown as [A]′ or $[A]^T$

Matrix multiplication: $[C]_{pxs} = [A]_{pxr}\,[B]_{rxs} \rightarrow c_{ij} = \sum_{k=1}^{r} a_{ik}\, b_{kj}$ i=1,...,p ; j=1,...,s

Note: $[A][B] \neq [B][A]$

Augmented matrix: is a special way of showing two matrices together.

For example $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ augmented with the column vector $\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ is $\left[\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{array}\right]$

Determinant of a matrix: is a single number. Determinant of [A] is shown as |A|.

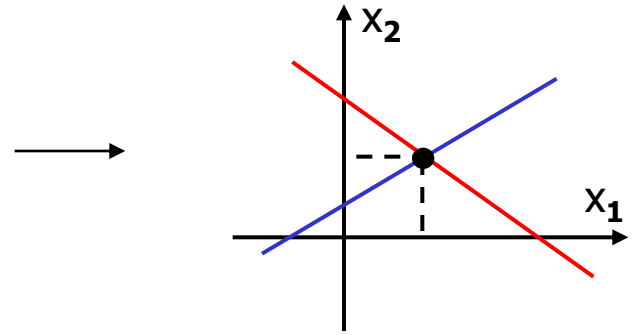
Graphical Method for Solving a Small (n≤3) Set of Equations

- Consider a set of 2 equations

$$a_{11}\, x_1 + a_{12}\, x_2 = b_1$$
$$a_{21}\, x_1 + a_{22}\, x_2 = b_2$$

- Plot these on the Cartesian coordinate system with axes x_1 and x_2.

$$x_2 = -\frac{a_{11}}{a_{12}} x_1 + \frac{b_1}{a_{12}}$$
$$x_2 = -\frac{a_{21}}{a_{22}} x_1 + \frac{b_2}{a_{22}}$$



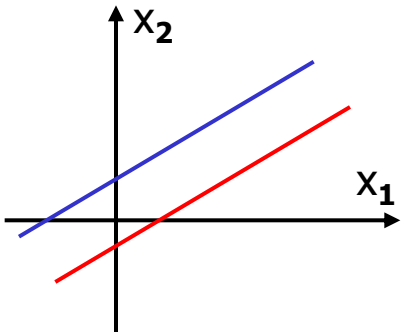
- For n=3, each equation will be a plane on a 3D coordinate system. Solution is the point where these planes intersect.

- For n>3, graphical solution is not practical.

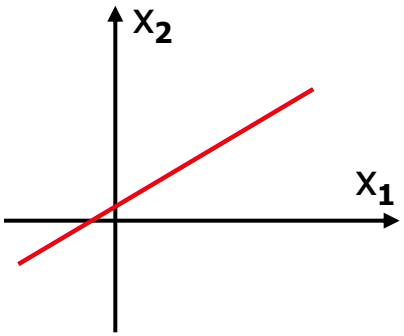
Graphical Method (cont'd)

Graphical Method is useful to illustrate:

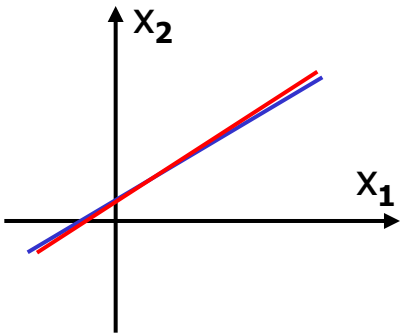
(1) no solution



(2) infinitely many solutions



(3) ill-conditioned system



- In system (1) and (2), equations are linearly dependent.
- In system (3), the slopes of the lines are very close to each other.

Mathematically

- Coefficient matrices of (1) & (2) are singular. Their determinants are zero and their inverse do not exist.
- Coefficient matrix of (3) is almost singular. Its inverse is difficult to take. This system has a unique solution, which is not easy to determine numerically because of its extreme sensitivity to round-off errors.

Cramer's Rule for Solving a Set of Equations

Determinant of a system

- Determinant of a 2x2 matrix is $D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$

- Determinant of a 3x3 matrix is $D = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$

- Determinant of a general nxn matrix can be calculated recursively using the above pattern.

- Determinant of a singular system is zero.

$$\begin{aligned} x_1 - 3x_2 &= 5 \\ 2x_1 - 6x_2 &= 7 \end{aligned}$$

- Determinant of an illl-conditioned system is close to zero.

$$\begin{aligned} 2x_1 - 3x_2 &= 5 \\ 3.98x_1 - 6x_2 &= 7 \end{aligned}$$

Warning: Multiply both equations of the above system with 100

$$\begin{aligned} 200x_1 - 300x_2 &= 500 \\ 398x_1 - 600x_2 &= 700 \end{aligned}$$

This system is as ill-conditioned as the previous one but it has a determinant 10000 times larger. Therefore we need to scale a system when we talk about the magnitude of its determinant. Details will come later.

Cramer's Rule for Solving a Set of Equations (Cont'd)

Cramer's Rule: Each unknown is calculated as a fraction of two determinants. The denominator is the determinant of the system, D. The numerator is the determinant of a modified system obtained by replacing the column of coefficients of the unknown being calculated by the RHS vector.

For a 3x3 system

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

- For a singular system D=0 $\rightarrow$ Solution can not be obtained.

- For large systems Cramer's rule is not practical because calculating determinants is costly.

- To solve an nxn system of equations, Cramer's rule needs n+1 determinant evaluations. Using a recursive algorithm, determinant of an nxn matrix requires 2n!+2n-1 arithmetic operations (+,-,x,÷). Therefore solving an nxn system with the Cramer's Rule needs more than (n+1)(2n!+2n-1). For a 20x20 system this means about $1x10^{20}$ operations (Check these numbers).

C and Matrices:

- A matrix is nothing but an array of arrays.
- In C array indices start from 0.
- C does not do check array bounds. Compilers might have switches for this.
- A 2D array (a matrix) is declared as `double A[4][5]` and its elements are reached as `A[i][j]`
- If you do not know the size of the matrices before starting the program, you need to use dynamic memory allocation with pointers. This can be quite confusing.

Exercise 11:

(i) Write two functions to add and multiply two matrices. These functions should take two matrices and their sizes as input arguments and calculate a third matrix. They should check if the input matrices are suitable for addition or muliplication.

(ii) Write a main program. Generate two small matrices. Add and multiply them by calling proper functions.

Exercise 12:

(i) Write a function that calculates the determinant of a matrix. Call this function recursively to solve systems of equations using the Cramer's Rule.

(ii) Use this program to solve a 10x10 set of equations. Generate the coefficient matrix of the system inside a loop. Make sure that your system is not singular. How long it takes to solve this system? Try solving larger systems.

Elimination of Unknowns Method

Example 11: Given a 2x2 set of equations:

$$2.5\ x_1 + 6.2\ x_2 = 3.0$$
$$4.8\ x_1 - 8.6\ x_2 = 5.5$$

- Multiply the 1st eqn by 8.6 and the 2nd eqn by 6.2:

$$21.50\ x_1 + 53.32\ x_2 = 25.8$$
$$29.76\ x_1 - 53.32\ x_2 = 34.1$$

- Add these equations: $51.26\ x_1 + 0\ x_2 = 59.9$

- Solve for x_1: $x_1 = 59.9/51.26 = 1.168552478$

- Using the 1st eqn solve for x_2: $x_2 = (3.0-2.5*1.168552478)/6.2 = 0.01268045242$

- Check if these satisfy the 2nd eqn: $4.8*1.168552478-8.6*0.01268045242 = 5.500000004$

(Difference is due to the round-off errors).

Naive Gauss Elimination Method

- It is a formalized way of the previous elimination technique.
- Consider the following system of n equations.

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \quad (1)$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \quad (2)$$
$$\ldots$$
$$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \quad (n)$$

Step 0 (optional): Form the augmented matrix of [A|B].

Step 1 Forward Elimination: Reduce the system to an upper triangular system.

(1.1) First eliminate x_1 from 2^{nd} to n^{th} equations.

- Multiply the 1^{st} eqn. by a_{21}/a_{11} & subtract it from the 2^{nd} equation. This is the new 2^{nd} eqn.
- Multiply the 1^{st} eqn. by a_{31}/a_{11} & subtract it from the 3^{rd} equation. This is the new 3^{rd} eqn.

...

- Multiply the 1^{st} eqn. by a_{n1}/a_{11} & subtract it from the n^{th} equation. This is the new n^{th} eqn.

Important: In these steps the 1^{st} eqn is the pivot equation and a_{11} is the pivot element. Note that a division by zero may occur if the pivot element is zero. Naive-Gauss Elimination does not check for this.

Naive Gauss Elimination Method (cont'd)

The modified system is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ 0 & a'_{32} & a'_{33} & \cdots & a'_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a'_{n2} & a'_{n3} & \cdots & a'_{nn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b'_2 \\ b'_3 \\ \vdots \\ b'_n \end{Bmatrix}$$

′ indicates that the system is modified once.

(1.2) Now eliminate x_2 from 3rd to nth equations.

The modified system is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ 0 & 0 & a''_{33} & \cdots & a''_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a''_{n3} & \cdots & a''_{nn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b'_2 \\ b''_3 \\ \vdots \\ b''_n \end{Bmatrix}$$

″ indicates that the system is modified once.

Repeat (1.1) and (1.2) upto (1.n-1).

At the end of Step 1, we will get this upper triangular system

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

Primes are removed for clarity.

Naive Gauss Elimination Method (cont'd)

Step 2 Back substitution: Find the unknowns starting from the last equation.

(2.1) Last equation involves only x_n. Solve for it.
(2.1) Use this x_n in the $(n-1)^{th}$ equation and solve for x_{n-1}.
...
(2.n) Use all previously calculated x values in the 1^{st} eqn and solve for x_1.

Example 12: Solve the following system using Naive Gauss Elimination.

$$\begin{aligned} 6x_1 - 2x_2 + 2x_3 + 4x_4 &= 16 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 &= 26 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 &= -19 \\ -6x_1 + 4x_2 + x_3 - 18x_4 &= -34 \end{aligned}$$

Step 0: Form the augmented matrix

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 12 & -8 & 6 & 10 & 26 \\ 3 & -13 & 9 & 3 & -19 \\ -6 & 4 & 1 & -18 & -34 \end{array}\right]$$

Naive Gauss Elimination Method (cont'd)

Step 1: Forward elimination

(1.1) Eliminate x_1

$$\begin{bmatrix} \boxed{6} & -2 & 2 & 4 & | & 16 \\ 0 & -4 & 2 & 2 & | & -6 \\ 0 & -12 & 8 & 1 & | & -27 \\ 0 & 2 & 3 & -14 & | & -18 \end{bmatrix}$$

(Does not change. Pivot is 6)

(1.2) Eliminate x_2

$$\begin{bmatrix} 6 & -2 & 2 & 4 & | & 16 \\ 0 & \boxed{-4} & 2 & 2 & | & -6 \\ 0 & 0 & 2 & -5 & | & -9 \\ 0 & 0 & 4 & -13 & | & -21 \end{bmatrix}$$

(Does not change.)
(Does not change. Pivot is -4)

(1.3) Eliminate x_3

$$\begin{bmatrix} 6 & -2 & 2 & 4 & | & 16 \\ 0 & -4 & 2 & 2 & | & -6 \\ 0 & 0 & \boxed{2} & -5 & | & -9 \\ 0 & 0 & 0 & -3 & | & -3 \end{bmatrix}$$

(Does not change.)
(Does not change.)
(Does not change. Pivot is 2)

Step 2: Back substitution

(2.1) Find x_4 $\quad x_4 = (-3)/(-3) = 1$

(2.2) Find x_3 $\quad x_3 = (-9+5*1)/2 = -2$

(2.3) Find x_2 $\quad x_2 = (-6-2*(-2)-2*1)/(-4) = 1$

(2.4) Find x_1 $\quad x_1 = (16+2*1-2*(-2)-4*1)/6 = 3$

Pseudocode for the Naive Gauss Elimination Method

For a general nxn system [A] {x} = {B}

Forward Elimination

```
LOOP k from 1 to n-1
   LOOP i from k+1 to n
      FACTOR = A ik / A kk
      LOOP j from k+1 to n
         A ij = A ij – FACTOR * A kj
      END LOOP
      B i = B i – FACTOR * B k
   ENDLOOP
ENDLOOP
```

Back Substitution

```
X n = B n / A nn
LOOP i from n-1 to 1
   SUM = 0.0
   LOOP j from i+1 to n
      SUM = SUM + A ij * X j
   END LOOP
   X i = (B i – SUM) / A ii
ENDLOOP
```

Exercise 13: Implement this in C. Write a main program and two functions. You need to know how to pass matrices to the functions.

Operation Count for the Naive Gauss Elimination Method

Let's count the FLOPS (floating point operations). Consider only multiplications and divisions, since they are more expensive to perform.

Forward Elimination

FACTOR = A_{ik} / A_{kk} is evaluated $\sum_{k=1}^{n-1} (n-k)$ times.

$A_{ij} = A_{ij} - FACTOR * A_{kj}$ is evaluated $\sum_{k=1}^{n-1} (n-k)(n-k)$ times.

$B_i = B_i - FACTOR * B_k$ is evaluated $\sum_{k=1}^{n-1} (n-k)$ times.

Total * and / $\sum_{k=1}^{n-1} (n-k)(n-k+2) = \boxed{\frac{n^3}{3}} + O(n^2)$

Back Substitution

$SUM = SUM + A_{ij} * X_j$ is evaluated $\sum_{i=1}^{n-1} (n-i)$ times.

$X_i = (B_i - SUM) / A_{ii}$ is evaluated $\sum_{i=1}^{n-1} 1$ times.

Total * and / $\sum_{i=1}^{n-1} (n-i+1) = \boxed{\frac{n^2}{2}} + O(n)$

Total $n^3/3 + n^2/2 + O(n^2) + O(n)$ as n increases $\rightarrow$ $\boxed{n^3/3}$

Example 13:

n	* & / in FE	* & / in BS	Total * & /	$n^3/3$	% due to FE
10	375	55	430	333	87.2 %
100	338250	5050	343300	333333	98.5 %
1000	3.34 E8	500500	3.34 E8	3.33 E8	99.8 %

Pivoting

• In Naive Gauss Elimination, a division by zero occurs if the pivot element is zero. Note that zero pivot elements may be created during the elimination step even if they are not present in the original matrix.

• Pivoting is used to avoid this problem. We interchange rows and columns at each step to put the coefficient with the largest magnitude on the diagonal.

• In addition to avoiding the division by zero problem, pivoting reduces the round-off errors. It makes the solution of ill-conditioned systems easier.

• **Complete pivoting** uses both row and column interchanges. It is not used frequently.

• **Partial pivoting** uses only row interchanges. We will use this.

• When there are large differences in magnitude of coefficients in one equation compared to the other equations we may also need **scaling**. Details will come later.

Pivoting Example

Example 14: Solve the following system using Gauss Elimination with pivoting.

$$
\begin{aligned}
2x_2 + \quad + x_4 &= 0 \\
2x_1 + 2x_2 + 3x_3 + 2x_4 &= -2 \\
4x_1 - 3x_2 \quad + x_4 &= -7 \\
6x_1 + x_2 - 6x_3 - 5x_4 &= 6
\end{aligned}
$$

Step 0: Form the augmented matrix

0	2	0	1	|	0
2	2	3	2	|	-2
4	-3	0	1	|	-7
6	1	-6	-5	|	6

Step 1: Forward Elimination

(1.1) Eliminate x_1 . But the pivot element is 0. We have to interchange the 1st row with one of the rows below it. Interchange it with the 4th row because 6 is the largest possible pivot.

6	1	-6	-5	|	6
2	2	3	2	|	-2
4	-3	0	1	|	-7
0	2	0	1	|	0

Now eliminate x_1

6	1	-6	-5	|	6
0	1.6667	5	3.6667	|	-4
0	-3.6667	4	4.3333	|	-11
0	2	0	1	|	0

Pivoting Example (cont'd)

(1.2) Eliminate x_2 .from the 3rd and 4th eqns. Pivot element is 1.6667. There is no division by zero problem. Still we will perform pivoting to reduce round-off errors. Interchange the 2nd and 3rd rows. Note that complete pivoting would interchange 2nd and 3rd columns.

```
6   1        -6  -5        |    6
0  -3.6667    4    4.3333  |  -11
0   1.6667    5    3.6667  |   -4
0   2         0    1       |    0
```

Eliminate x_2

```
6   1        -6        -5        |    6
0  -3.6667    4         4.3333   |  -11
0   0         6.8182    5.6364   |  -9.0001
0   0         2.1818    3.3636   |  -5.9999
```

(1.3) Eliminate x_3 . 6.8182 > 2.1818, therefore no pivoting is necessary.

```
6   1        -6         -5        |    6
0  -3.6667    4          4.3333   |  -11
0   0         6.8182     5.6364   |  -9.0001
0   0         0          1.5600   |  -3.1199
```

Step 2: Back substitution

x_4 = -3.1199 / 1.5600 = **-1.9999**

x_3 = [-9.0001 – 5.6364*(-1.9999)] / 6.8182 = **0.33325**

x_2 = [-11 – 4.3333*(-1.9999) – 4*0.33325] / -3.6667 = **1.0000**

x_1 = [6 – (-5)*(-1.9999) – (-6)*0.33325 – 1*1.0000] / 6 = **-0.50000**

Exact solution is x = [-2 1/3 1 -0.5]^T . Use more than 5 sig. figs. to reduce round-off errors.

Scaling

• Normalize the equations so that the maximum coefficient in every row is equal to 1.0. That is, divide each row by the coefficient in that row with the maximum magnitude.

• It is advised to scale a system before calculating its determinant. This is especially important if we are calculating the determinant to see if the system is ill-conditioned or not.

• Consider the following systems

$$\begin{aligned} 2x_1 - 3x_2 &= 5 \\ 3.98x_1 - 6x_2 &= 7 \end{aligned} \qquad\qquad \begin{aligned} 20x_1 - 30x_2 &= 50 \\ 39.8x_1 - 60x_2 &= 70 \end{aligned}$$

• They are actually the same system. In the second one the equations are multiplied by 10.
• Determinant of the 1st system is 2(-6) – (-3)(3.98) = -0.06 , which is close to zero.
• Determinant of the 2nd system is 20(-60) – (-30)(39.8) = -6 , which is not that close to zero.
• So is this system ill-conditioned or not?

• Scale this system. They are the same, use the 1st one.

$$\begin{aligned} 0.6667\,x_1 + x_2 &= -1.6667 \\ -0.6633\,x_1 + x_2 &= -1.1667 \end{aligned}$$

• Now calculate the determinant as 0.6667*1 – 1*(-0.6633) = 1.33. Use this value in judging the systems condition. Note that "How close to zero?" is still an open question. There are other ways to determine a systems condition (See page 277).

Scaled Partial Pivoting

- Scaling is also useful when some rows have coefficients that are large compared to those in other rows.

- Consider the following system

$$2x_1 + 100000x_2 = 100000$$
$$x_1 + x_2 = 2$$

Exact solution is (1.00002, 0.99998)

(a) If we solve this system with Gauss Elimination, no pivoting is necessary (2 is larger than 1). Use only 3 sig. figs to emphasize the round-off errors.

$$2.00*10^0\, x_1 + 1.00*10^5\, x_2 = 1.00*10^5$$
$$-5.00*10^4\, x_2 = -5.00*10^4$$

$\longrightarrow$

$x_1 = 0.00*10^1$ WRONG
$x_2 = 1.00*10^1$ OK

(b) First scale the system and than solve with Gauss Elimination.

$$2.00*10^{-5}\, x_1 + x_2 = 1$$
$$x_1 + x_2 = 2$$

This system now needs pivoting. Interchange the rows and solve.

$$2.00*10^{-5}\, x_1 + x_2 = 1.00*10^0$$
$$x_1 + x_2 = 2.00*10^0$$

$\longrightarrow$

$$x_1 + x_2 = 2$$
$$x_2 = 1$$

$\longrightarrow$

$x1 = 1.00$ OK
$x2 = 1.00$ OK

Conclusion: Scaling showed that pivoting is necessary. But scaling itself is not necessary (pivot the original system and solve). Scaling also introduces additional round-off errors. Therefore use scaling to decide whether pivoting is necessary or not but than use the original coefficients.

Example for Gauss Elimination with Scaled Partial Pivoting

Example 15: Solve the following system using Gauss Elimination with scaled partial pivoting. Keep numbers as fractions of integers to eliminate round-off errors.

```
 3   -13     9     3   | -19
-6     4     1   -18   | -34
 6    -2     2     4   |  16
12    -8     6    10   |  26
```

Start by forming the scale vector. It has the largest coefficient (in magnitude) of each row.

```
SV = {13   18   6   12}
```

This scale vector will be updated if we interchange rows during pivoting.

Step 1: Forward Elimination

Example for Gauss Elimination with Scaled Pivoting (cont'd)

(1.1) Compare scaled coefficients 3/13, 6/18, 6/6, 12/12. Third one is the largest (actually fourth one is the same but we use the first occurance). Interchange rows 1 and 3.

```
 6   -2    2    4  |  16
-6    4    1  -18  | -34
 3  -13    9    3  | -19
12   -8    6   10  |  26
```

Update the scale vector `sv = {6 18 13 12}`

Eliminate x_1.

Subtract (-6/6) times row 1 from row 2.

Subtract (3/6) times row 1 from row 3.

Subtract (12/6) times row 1 from row 4.

Resulting system is

```
 6   -2    2    4  |  16
 0    2    3  -14  | -18
 0  -12    8    1  | -27
 0   -4    2    2  |  -6
```

Example for Gauss Elimination with Scaled Pivoting (cont'd)

(1.2) Compare scaled coefficients 2/18, 12/13, 4/12. Second one is the largest. Interchange rows 2 and 3.

```
6   -2    2    4  |  16
0  -12    8    1  | -27
0    2    3  -14  | -18
0   -4    2    2  |  -6
```

Update the scale vector `sv = {6 13 18 12}`

Eliminate x_2.

Subtract (2/(-12)) times row 2 from row 3.

Subtract ((-4)/(-12)) times row 2 from row 4.

Resulting system is

```
6   -2    2      4     |  16
0  -12    8      1     | -27
0    0  13/3   -83/6   | -45/2
0    0  -2/3    5/3    |   3
```

Example for Gauss Elimination with Scaled Pivoting (cont'd)

(1.3) Compare scaled coefficients (13/3)/18, (2/3)/12. First one is larger. No need for pivoting.

Scale vector remains the same **`sv = {6 13 18 12}`**

Eliminate x_3.

Subtract ((-2/3)/(13/3)) times row 3 from row 4.

Resulting system is

```
6    -2     2      4     |  16
0   -12     8      1     | -27
0     0  13/3   -83/6     | -45/2
0     0     0   -6/13     | -6/13
```

Step 2: Back substitution

Equation 4 $\rightarrow$ $x_4 = 1$

Equation 3 $\rightarrow$ $x_3 = -2$

Equation 2 $\rightarrow$ $x_2 = 1$

Equation 1 $\rightarrow$ $x_1 = 3$

These are exact results because round-off errors are eliminated by not using floating point numbers.

Ill-conditioned Systems

- They have almost singular coefficient matrices.
- They have a unique solution.
- Consider the following 3x3 system

```
  3.02  -1.05   2.53 | -1.61
  4.33   0.56  -1.87 |  7.23
 -0.83  -0.54   1.47 | -3.38
```

The solution of this system is $[1 \quad 2 \quad -1]^T$

- Change b_1 from –1.61 to –1.60. The solutions changes to $[1.0566 \quad 4.0051 \quad -0.2315]^T$
- Or change a_{11} from 3.02 to 3.00. The solutions changes to $[1.1277 \quad 6.5221 \quad 0.7333]^T$
- This is a typical ill-conditioned system. Its solution is very sensitive to the changes in the coefficient matrix or the right-hand-side vector.
- This means it is very sensitive to round-off errors. Double precision must be used to reduce round-off errors as much as possible.
- Scaled pivoting should also be used to decrease round-off errors.
- Fortunately not many engineering problems will result in an ill-conditioned system.

Other Uses of Gauss Elimination

• Gives the LU decomposition of A such that [L][U]=[A]. LU decomposition is useful if we are solving many systems with the same coefficient matrix A but different right-hand-side vectors (See page 266).

• It can be used to calculate the determinant of a matrix. At the end of the Forward Elimination step we get an upper triangular matrix. For this matrix the determinant is just the multiplication of diagonal elements. If we interchanged rows m times, than the deteminant can be calculated as

$$|A| = (-1)^m * a_{11} * a_{22} * \dots a_{nn}$$

Example 16: Remember the example we used to describe pivoting. The A matrix was

```
0    2    0    1
2    2    3    2
4   -3    0    1
6    1   -6   -5
```

After the Forward Elimination step we found the following upper triangular matrix.

```
6   1        -6         -5
0  -3.6667    4          4.3333
0   0         6.8182     5.6364
0   0         0          1.5600
```

To get this, we used pivoting and interchanged rows twice. Therefore the determinant of A is

$|A| = (-1)^2 * 6 * (-3.6667) * 6.8182 * 1.56 = -234.0028$

Gauss-Jordan Method

• This is another elimination technique. It is a variation of Gauss Elimination.

• The difference is, when an unknown is eliminated, it is eliminated from all other equations, not just the subsequent ones. At the same time all rows are normalized by dividing them to their pivot element.

• At the end of the forward elimination step Gauss-Jordan method yields an identity matrix, not an upper triangular one.

• There is no need for back substitution. Right-hand-side vector has the results.

Example 17: Solve the following 4x4 system using G-J method with pivoting. Note that this is the same system that we used to demonstrate Gauss Elimination with pivoting.

```
0    2    0    1 |   0
2    2    3    2 |  -2
4   -3    0    1 |  -7
6    1   -6   -5 |   6
```

• Interchange rows 1 and 4, divide the new first row by 6 and eliminate x_1 from the 2nd, 3rd and 4th equations.

```
1   0.1667 -1   -0.8333 |   1
0   1.6667  5    3.6667 |  -4
0  -3.6667  4    4.3333 | -11
0   2       0    1      |   0
```

Gauss-Jordan Method (cont'd)

• Interchange rows 2 and 3, divide the new second row by -3.6667 and eliminate x_2 from the 1st, 3rd and 4th equations.

```
1   0   -0.8182   -0.6364 |  0.5
0   1   -1.0909   -1.1818 |  3
0   0    6.8182    5.6364 | -9
0   0    2.1818    3.3636 | -6
```

• No pivoting is required. Divide the third row by 6.8182 and eliminate x_3 from the 1st, 2nd and 4th equations.

```
1   0   0    0.04    |  0.58
0   1   0   -0.280   |  1.56
0   0   1    0       | -1.32
0   0   0    1.5599  | -3.12
```

• Divide the last row by 1.5599 and eliminate x_4 from the 1st, 2nd and 3rd equations.

```
1   0   0   0 | -0.5
0   1   0   0 |  1.0001
0   0   1   0 |  0.3333
0   0   0   1 | -2
```

• Right-hand-side vector is the solution. No back substitution is required.

• Gauss-Jordan requires almost 50% more operations than Gauss Elimination ($n^3/2$ instead of $n^3/3$).

Calculating the Inverse of a Matrix with Gauss-Jordan

• Calculating the inverse of a matrix is important if we want to solve many systems with the same coefficient matrix, but different right-hand-side vectors.

• To take the inverse of A, augment it with an identity matrix and apply the Gauss-Jordan method.

Example 18: Find the inverse of the following matrix.

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix}$$

• Augment A with a 3x3 identity matrix.

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

• Apply the Gauss-Jordan method to this system to get

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0.4 & -0.2 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -0.2 & 0.6 \end{array}\right]$$

• 3x3 matrix at the right is the inverse of A. This inverse can now be used to solve a system with the coefficient matrix A and different right-hand-side vectors as $\{x\} = [A]^{-1} \{B\}$

• LU decomposition can be used for the same purpose and it is more efficient (See page 273).

Iterative Methods - Gauss-Seidel Method

- Gauss Elimination and its variants are called direct methods. They are not preferred for large systems.
- Iterative methods start with an initial solution vector and iteratively converge to the true solution.
- They stop when the pre-specified tolerance is reached and round-off errors are not an issue.
- Given the system [A]{x}={B} and starting values $\{x\}^0$, Gauss-Seidel uses the first equation to solve for x_1, second for x_2, etc.

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3 - \ \ldots\ldots - a_{1n}x_n) / a_{11}$$
$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3 - \ \ldots\ldots - a_{2n}x_n) / a_{22}$$
$$\ldots\ldots$$
$$x_n = (b_n - a_{n1}x_1 - a_{n2}x_2 - \ \ldots\ldots - a_{(n-1)(n-1)}x_{n-1}) / a_{nn}$$

After the first iteration you get $\{x\}^1$. Use these values to start a new iteration. Repeat until the tolerance is satisfied as

$$|\varepsilon_{a,i}| = \left|\frac{x_i^k - x_i^{k-1}}{x_i^k}\right| 100\% < \varepsilon_s$$

for all the unknowns (i=1,…,n), where k and k-1 represent the present and previous iterations.

Gauss-Seidel Example

Solve the following system using the Gauss-Seidel Method

$$\begin{aligned} 6x_1 - 2x_2 + x_3 &= 11 \\ -2x_1 + 7x_2 + 2x_3 &= 5 \\ x_1 + 2x_2 - 5x_3 &= -1 \end{aligned}$$

starting with $x_1^0 = x_2^0 = x_3^0 = 0.0$

- Rearrange the equations

$$\begin{aligned} x_1 &= (11 + 2x_2 - x_3) / 6 \\ x_2 &= (5 + 2x_1 - 2x_3) / 7 \\ x_3 &= (1 + x_1 + 2x_2) / 5 \end{aligned}$$

- First iteration

$$\begin{aligned} x_1^1 &= (11 + 2x_2^0 - x_3^0) / 6 &&= (11 + 0 - 0)/6 = 1.833 \\ x_2^1 &= (5 + 2x_1^1 - 2x_3^0) / 7 &&= (5 + 2*1.8333 - 0)/7 = 1.238 \\ x_3^1 &= (1 + x_1^1 + 2x_2^1) / 5 &&= (1 + 1.8333 + 2*1.2381)/5 = 1.062 \end{aligned}$$

- Second iteration

$$\begin{aligned} x_1^2 &= (11 + 2x_2^1 - x_3^1) / 6 &&= (11 + 2*1.238 - 1.062)/6 = 2.069 \\ x_2^2 &= (5 + 2x_1^2 - 2x_3^1) / 7 &&= (5 + 2*2.069 - 2*1.062)/7 = 1.002 \\ x_3^2 &= (1 + x_1^2 + 2x_2^2) / 5 &&= (1 + 2.069 + 2*1.002)/5 = 1.015 \end{aligned}$$

- Continue like this to get (Do not forget to check convergence using the specified tolerance)

	Zeroth	First	Second	Third	Fourth	Fifth
x_1	0.000	1.833	2.069	1.998	1.999	2.000
x_2	0.000	1.238	1.002	0.995	1.000	1.000
x_3	0.000	1.062	1.015	0.998	1.000	1.000

Jacobi Method

• Gauss- Seidel always uses the newest available x values. Jacobi Method uses x values from the previous iteration.

Example 19: Repeat the previous example using the Jacobi method.

• Rearrange the equations

$$x_1 = (11 + 2x_2 - x_3) / 6$$
$$x_2 = (5 + 2x_1 - 2x_3) / 7$$
$$x_3 = (1 + x_1 + 2x_2) / 5$$

• First iteration (use x^0 values)

$$x_1^1 = (11 + 2x_2^0 - x_3^0) / 6 = (11 + 0 - 0)/6 = 1.833$$
$$x_2^1 = (5 + 2x_1^0 - 2x_3^0) / 7 = (5 + 0 - 0)/7 = 0.714$$
$$x_3^1 = (1 + x_1^0 + 2x_2^0) / 5 = (1 + 0 + 0)/5 = 0.200$$

• Second iteration (use x^1 values)

$$x_1^2 = (11 + 2x_2^1 - x_3^1) / 6 = (11 + 2*0.714 - 0.200)/6 = 2.038$$
$$x_2^2 = (5 + 2x_1^1 - 2x_3^1) / 7 = (5 + 2*1.833 - 2*0.200)/7 = 1.181$$
$$x_3^2 = (1 + x_1^1 + 2x_2^1) / 5 = (1 + 1.833 + 2*0.714)/5 = 0.852$$

• Continue to get

	Zeroth	First	Second	Third	Fourth	Fifth	. . .	Eighth
x_1	0.000	1.833	2.038	2.085	2.004	1.994	. . .	2.000
x_2	0.000	0.714	1.181	1.053	1.001	0.990	. . .	1.000
x_3	0.000	0.200	0.852	1.080	1.038	1.001	. . .	1.000

Convergence of the Iterative Methods

• Note that these methods can be seen as the multiple application of Simple One-Point Iteration.

• They may converge or diverge. A convergence criteria can be derived starting from the convergence criteria of the Simple One-Point Iteration method.

• When the system of equations can be ordered so that each diagonal entry of the coefficient matrix is larger in magnitude than the sum of the magnitudes of the other coefficients in that row (**diagonally dominant system**) the iterations converge for any starting values. This is a sufficient but not necessary criteria.

$$\textbf{For all } i = 1,\ldots,n \qquad |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

• Many engineering problems satisfy this requirement.

• The example we used has a diagonally dominant coefficient matrix.

• Gauss-Seidel converges usually faster than the Jacobi method.

Exercise 14: Interchange the 2nd and 3rd equations of the previous example. Is the system diagonally dominant now? Solve this system using both iterative methods. Compare their convergence rates with that of the solved example. Do you observe any divergence?

Improving Gauss-Seidel with Relaxation

• After each new value of x is computed, that value is modified by a weighted average of the results of the previous and present iterations,

$$x_i^{new} = \lambda\, x_i^{new} + (1 - \lambda)\, x_i^{old} \qquad 0 < \lambda < 2 \text{ : weighting factor}$$

• If $\lambda = 1$ $\rightarrow$ no relaxation (Original Gauss-Seidel)

• If $0<\lambda<1$ $\rightarrow$ under relaxation (Used to make a diverging system converge)

• If $1<\lambda<2$ $\rightarrow$ over relaxation (Used to speed up the convergence of an already converging system. Called SOR, Successive (or Simultaneous) Over Relaxation)

• Choice of λ is problem specific.

Exercise 15: Solve the system we used in the Gauss-Seidel example with different weighting factors. Do you notice any difference in the convergence rate?

Pseudocode for the Gauss-Seidel Method

```
LOOP k from 1 to maxIter
    LOOP i from 1 to n
        x_i = B_i
        LOOP j from 1 to n
            IF (i ≠ j)   x_i = x_i - A_ij x_j
        ENDLOOP
        x_i = x_i / A_ii
    ENDLOOP
    CONVERGED = TRUE
    LOOP i from 1 to n
        OUTPUT x_i
        ε_a = | (x_i – x_i^old) / x_i | * 100
        IF (ε_a > tolerance)   CONVERGED = FALSE
    ENDLOOP
    IF (CONVERGED = TRUE)  STOP
ENDLOOP
```

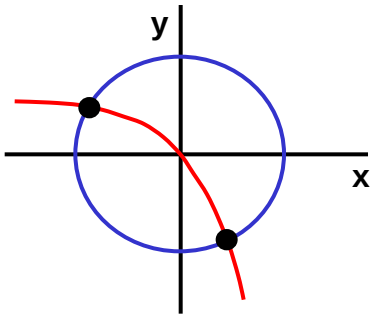
Exercise 16: Improve this pseudocode with relaxation.

Exercise 17: Modify this pseudocode for the Jacobi method.

Solving System of Nonlinear Equations

Example 20: Solving a 2x2 System Using Simple One-Point Iteration

Solve the following system of equations $x^2 + y^2 = 4$, $e^x + y = 1$



Put the functions into the form $x=g_1(x,y)$, $y=g_2(x,y)$

$$x = g_1(y) = \ln(1 - y)$$

$$y = g_2(x) = -\sqrt{4 - x^2}$$

Select a starting values for x and y, such as $x_0=0.0$ and $y_0=0.0$. They don't need to satisfy the equations. Use these values in g functions to calculate new values.

$x_1 = g_1(y_0) = 0$ $\quad$ $y_1 = g_2(x_1) = -2$

$x_2 = g_1(y_1) = 1.098612289$ $\quad$ $y_2 = g_2(x_2) = -1.67124236$

$x_3 = g_1(y_2) = 0.982543669$ $\quad$ $y_3 = g_2(x_3) = -1.74201261$

$x_4 = g_1(y_3) = 1.00869218$ $\quad$ $y_4 = g_2(x_4) = -1.72700321$

The solution is converging to the exact solution of x=1.004169 , y=-1.729637

Exercise 18: Solve the same system but rearrange the equations as $x=\ln(1-y)$ $\quad$ $y = (4-x^2)/y$ and start from $x_0=1$ $y_0=-1.7$. Remember that this method may diverge.

Solving System of Nonlinear Equations (cont'd)

Solving a 2x2 System Using Newton-Raphson Method

Consider the following general form of a two equation system

$$u(x,y) = 0$$
$$v(x,y) = 0$$

Write 1st order TSE for these equations

$$u_{i+1} = u_i + \frac{\partial u_i}{\partial x}(x_{i+1} - x_i) + \frac{\partial u_i}{\partial y}(y_{i+1} - y_i)$$

$$v_{i+1} = v_i + \frac{\partial v_i}{\partial x}(x_{i+1} - x_i) + \frac{\partial v_i}{\partial y}(y_{i+1} - y_i)$$

To find the solution set $u_{i+1} = 0$ and $v_{i+1} = 0$. Rearrange

$$\frac{\partial u_i}{\partial x}x_{i+1} + \frac{\partial u_i}{\partial y}y_{i+1} = -u_i + \frac{\partial u_i}{\partial x}x_i + \frac{\partial u_i}{\partial y}y_i$$

$$\frac{\partial v_i}{\partial x}x_{i+1} + \frac{\partial v_i}{\partial y}y_{i+1} = -v_i + \frac{\partial v_i}{\partial x}x_i + \frac{\partial v_i}{\partial y}y_i$$

$$\begin{bmatrix} \frac{\partial u_i}{\partial x} & \frac{\partial u_i}{\partial y} \\ \frac{\partial v_i}{\partial x} & \frac{\partial v_i}{\partial y} \end{bmatrix} \begin{Bmatrix} x_{i+1} \\ y_{i+1} \end{Bmatrix} = \begin{Bmatrix} -u_i + \frac{\partial u_i}{\partial x}x_i + \frac{\partial u_i}{\partial y}y_i \\ -v_i + \frac{\partial v_i}{\partial x}x_i + \frac{\partial v_i}{\partial y}y_i \end{Bmatrix}$$

The first matrix is called the Jacobian matrix.

Solve for x_{i+1}, y_{i+1} and iterate.

Can be generalized for n simultaneous equations

Solving System of Eqns Using NR Method (cont'd)

Solve for x_{i+1} and y_{i+1} using Cramer's rule (ME 210)

$$x_{i+1} = x_i - \frac{u_i \frac{\partial v_i}{\partial y} - v_i \frac{\partial u_i}{\partial y}}{\frac{\partial u_i}{\partial x}\frac{\partial v_i}{\partial y} - \frac{\partial u_i}{\partial y}\frac{\partial v_i}{\partial x}} \quad , \quad y_{i+1} = y_i - \frac{v_i \frac{\partial u_i}{\partial x} - u_i \frac{\partial v_i}{\partial x}}{\frac{\partial u_i}{\partial x}\frac{\partial v_i}{\partial y} - \frac{\partial u_i}{\partial y}\frac{\partial v_i}{\partial x}}$$

The denominator is the determinant of the Jacobian matrix |J|.

Example 21: Solve the same system of equations $u = x^2 + y^2 - 4 = 0$, $v = e^x + y - 1 = 0$ stating with $x_0=1$, $y_0=1$

$$\frac{\partial u}{\partial x} = 2x \;,\; \frac{\partial u}{\partial y} = 2y \;,\; \frac{\partial v}{\partial x} = e^x \;,\; \frac{\partial v}{\partial y} = 1 \quad \rightarrow \quad |J| = 2x - 2ye^x$$

$$x_{i+1} = x_i - \frac{u_i - 2y_i v_i}{|J|} \;,\; y_{i+1} = y_i - \frac{2xv_i - e^{x_i} u_i}{|J|}$$

i=0,	x_0=1	y_0=1	u_0=-2	v_0=2.718282	$\|J_0\|$=-3.436564
i=1,	x_1=-1.163953	y_1=3.335387	u_1=8.479595	v_1=2.647636	$\|J_1\|$=-4.410851
i=2,	x_2=-3.245681	y_2=1.337769	u_2=8.324069	v_2=0.376711	$\|J_2\|$=-6.595552
i=3,	x_3=-2.136423	y_3=0.959110	u_3=1.484197	v_3=0.077187	$\|J_3\|$=-4.499343

Looks like it is converging to the root in the 2nd quadrant $x \approx -1.8$, $y \approx 0.8$.

Exercise 19: Can you start with $x_0=0$ and $y_0=0$?

Exercise 20: Try to find starting points that will converge to the solution in the 4th quadrant.

Solving System of Nonlinear Equations (cont'd)

• Previous N-R solution of a system of 2 nonlinear equations can be generalized to the solution of a system of n nonlinear equations (See page 257 for details).

• Given the following n nonlinear equations with n unknowns

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

• Form the Jacobian matrix.

$$[Z]^k = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}^k$$

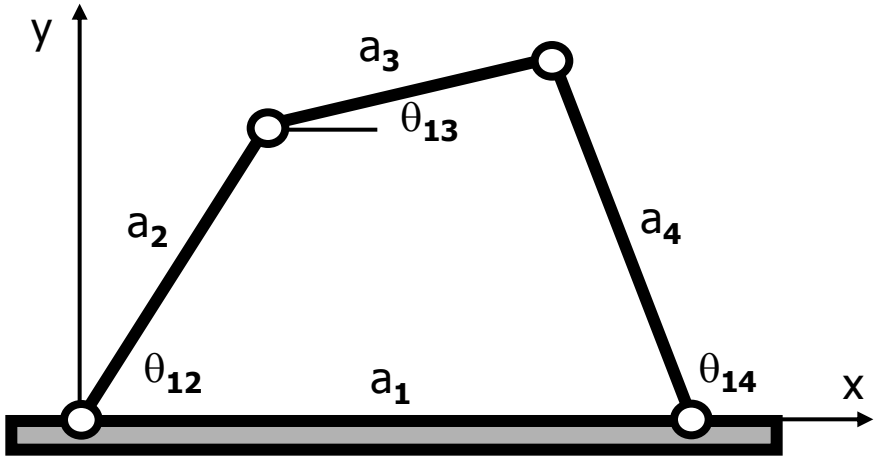
k is the iteration counter.

Calculate a new [Z] at each iteration.

• Solve the following system using any of the techniques that we learned in this chapter.

$$[Z]^k\{x\}^{k+1} = -\{F\}^k + [Z]^k\{x\}^k \qquad \text{where} \qquad F_i^k = f_i(x_1^k, x_2^k, \ldots, x_n^k)$$

Example 22 (Fall 2002 Exam Question):



Link lengths:

$a_1 = 4$, $a_2 = 2$, $a_3 = 4$, $a_4 = 5$

Crank angle: $\theta_{12} = 45° = \pi/4$

- Use the Newton-Raphson method to solve the following set of equations for θ_{13} and θ_{14}.
- Start with the following intial guesses: $\theta^0_{13} = 55°$, $\theta^0_{14} = 80°$
- Perform two iterations.

$$f_1(\theta_{13}, \theta_{14}) = a_2 \cos(\theta_{12}) + a_3 \cos(\theta_{13}) - a_4 \cos(\theta_{14}) - a_1 = 0$$

$$f_2(\theta_{13}, \theta_{14}) = a_2 \sin(\theta_{12}) + a_3 \sin(\theta_{13}) - a_4 \sin(\theta_{14}) = 0$$

Example 22 (cont'd)

- N-R can be applied to nonlinear equations as follows

$$[Z]^{k}\{X\}^{k+1} = -\{F\}^{k} + [Z]^{k}\{X\}^{k}$$

where $[Z]^{k}$ is the Jacobian matrix of the k^{th} iteration

$$[Z]^{k} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}^{k}$$

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial f_1}{\partial \theta_{13}} = -a_3 \sin(\theta_{13}) \qquad \frac{\partial f_1}{\partial x_2} = \frac{\partial f_1}{\partial \theta_{14}} = a_4 \sin(\theta_{14})$$

$$\frac{\partial f_2}{\partial x_1} = \frac{\partial f_2}{\partial \theta_{13}} = a_3 \cos(\theta_{13}) \qquad \frac{\partial f_2}{\partial x_2} = \frac{\partial f_2}{\partial \theta_{14}} = -a_4 \cos(\theta_{14})$$

- Iteration 0: $\theta^{0}_{13} = 55°$, $\theta^{0}_{14} = 80°$

Example 22 (cont'd)

- Iteration 1: $\left.\frac{\partial f_1}{\partial \theta_{13}}\right|^0 = -3.2766$ $\left.\frac{\partial f_1}{\partial \theta_{14}}\right|^0 = 4.9240$ $\left.\frac{\partial f_2}{\partial \theta_{13}}\right|^0 = 2.2943$ $\left.\frac{\partial f_2}{\partial \theta_{14}}\right|^0 = -0.8682$

$$[Z]^0 = \begin{bmatrix} -3.2766 & 4.9240 \\ 2.2943 & -0.8682 \end{bmatrix} \qquad \{F\}^0 = \begin{Bmatrix} -1.1597 \\ -0.2332 \end{Bmatrix}$$

Therefore the linear system is

$$\begin{bmatrix} -3.2766 & 4.9240 \\ 2.2943 & -0.8682 \end{bmatrix} \begin{Bmatrix} \theta_{13}^1 \\ \theta_{14}^1 \end{Bmatrix} = -\begin{Bmatrix} -1.1597 \\ -0.2332 \end{Bmatrix} + \begin{bmatrix} -3.2766 & 4.9240 \\ 2.2943 & -0.8682 \end{bmatrix} \begin{Bmatrix} 55 \\ 80 \end{Bmatrix} = \begin{Bmatrix} 214.8667 \\ 56.9637 \end{Bmatrix}$$

Solve this system using Gauss Elimination (or any other technique that we learned) to get

$\theta^1_{13} = 55.2550°$ $\theta^1_{14} = 80.4052°$

Example 22 (cont'd)

- Iteration 2: $\frac{\partial f_1}{\partial \theta_{13}}^{1} = -3.2868$ $\frac{\partial f_1}{\partial \theta_{14}}^{1} = 4.9301$ $\frac{\partial f_2}{\partial \theta_{13}}^{1} = 2.2797$ $\frac{\partial f_2}{\partial \theta_{14}}^{1} = -0.8334$

$$[Z]^1 = \begin{bmatrix} -3.2868 & 4.9301 \\ 2.2797 & -0.8334 \end{bmatrix} \qquad \{F\}^1 = \begin{Bmatrix} -1.1395 \\ -0.2291 \end{Bmatrix}$$

Therefore the linear system is

$$\begin{bmatrix} -3.2868 & 4.9301 \\ 2.2797 & -0.8334 \end{bmatrix} \begin{Bmatrix} \theta_{13}^1 \\ \theta_{14}^1 \end{Bmatrix} = -\begin{Bmatrix} -1.1395 \\ -0.2291 \end{Bmatrix} + \begin{bmatrix} -3.2868 & 4.9301 \\ 2.2797 & -0.8334 \end{bmatrix} \begin{Bmatrix} 55.2550 \\ 80.4052 \end{Bmatrix} = \begin{Bmatrix} 215.9330 \\ 59.1842 \end{Bmatrix}$$

Solve this system using Gauss Elimination (or any other technique that we learned) to get

$\theta^{1}_{13} = 55.4996°$ $\theta^{1}_{14} = 80.7994°$