Alexandria University,

Faculty of Engineering,

Computer and Systems Engineering Dept.

Numerical Analysis

CS213

*Final Project*

System Of Linear Equations

Part2

Names:

1. Ziad Hisham Ali. (19)
2. Mohamed Salah Mahmoud (41)
3. Mohamed Mohamed Abdel-Hakeem (43)
4. Youssef Ahmed Sayed Gabriel. (61)
5. Problem Statement:

Solving system of linear equations.

1. Pseudo Code:

* **Gauss Elimination:**

Parameters : ( Number Of Equations, Equations)

Algorithm : Put the coefficients in Matrix A and result in Matrix B

Make two iterations inside each other each of size n

Calculate factor mij = A(j,i)/A(i,i) & tempA = A\*mij &

tempB = B\*mij & A(j,i) = A(j,i) - tempA & B(j,i) = B(j,i) -

tempB And set arr(m) = B(m)/A(m,m) , then make another

two iterations inside each

others x, intiate sum = 0 in the second iteration

sum = sum + A(i,j)\*arr(j) & Finally the solution is arr

In each iteration “iterate, coefficients, error” stored in

array

Plot : 1.Draw plot between number of iterations & root using the array

2.Draw plot between number of iterations & error using the

Array

3. Draw plot between x & f(x) using the array

* **LU Decomposition:**

Parameters : ( Number Of Equations, Equations)

Algorithm : Put the coefficients in Matrix A and result in Matrix B &

intiate lu = A

Make two iterations inside each other each of size n-1

Calculate factor m = lu(j,i)/lu(i,i) &

lu(j,i to n) = lu(j,i to n)-factor\*lu(i,i to n) & lu(j,i) = m

After this set Matrix y(1) = b(1)

, then make another two iterations inside each other

First one y(i) = b(i) and inside the another

y(i) = y(i) - lu(i,j)\*y(j)

After that set Matrix x(n) = y(n)/lu(n,n) , then make

a while loop from 0 to n-1 , set x(i) = y(i) and make iterate

from i+1 to n to set x(i) = x(i)/lu(i,i) and decrease iterator

by 1 & Finally the solution is x.

In each iteration “iterate, coefficients, error” stored in

array

Plot : 1.Draw plot between number of iterations & root using the array

2.Draw plot between number of iterations & error using the

Array

3. Draw plot between x & f(x) using the array

* **Gauss Jordan:**

Parameters : ( Number Of Equations, Equations)

Algorithm : Put the coefficients in Matrix A and result in Matrix B &

Make three iterations two inside one other each of size n

Calculate factor mij = A(j,i)/A(i,i) & tempA = A\*mij &

tempB = B\*mij & A(j,i) = A(j,i) - tempA & B(j,i) = B(j,i) -

tempB And set arr(m) = B(m)/A(m,m) , then the second

iteration from 1 to iterator of

previous iteration – 1 and also Calculate factor mij =

A(j,i)/A(i,i) & tempA = A\*mij & tempB = B\*mij & A(j,i) =

A(j,i) - tempA & B(j,i) = B(j,i) - tempB & Finally the

solution is arr

In each iteration “iterate, coefficients, error” stored in

array

Plot : 1.Draw plot between number of iterations & root using the array

2.Draw plot between number of iterations & error using the

Array

3. Draw plot between x & f(x) using the array

* **Gauss Seidel:**

Parameters : ( Number Of Equations, Equations, initial guess, Epsilon)

Algorithm : Put the coefficients in Matrix A and result in Matrix B &

set diag\_inverse Matrix , then iterate from 1 to n and set

diag\_inverse(i,i) = 1 / A(i,i) , then set LU\_mat Matrix And

Make two iterations inside each other, first one from 1 to n

And the another from a to n ,set old\_x = x also set

x(j)= diag\_inverse(j,j)\*[B(j) - LU\_mat(j,n)\*x] , then

calculate the error = (x-old\_x)/x and precision = 100-error

and if error < Epsilon , then break And the solution is X

In each iteration “iterate, coefficients, error” stored in

array

Plot : 1.Draw plot between number of iterations & root using the array

2.Draw plot between number of iterations & error using the

Array

1. Analysis Of The Methods:

2\*x1 + x2 + 4\*x3 = 1

x1 + 2\*x2 + 3\*x3 = 1.5

4\*x1 – x2 + 2\*x3 = 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **Executed Time** | **Number Of Iterations** | **Graph** |
| Gauss Elimination | 0.17978 | None |  |
| LU\_Decomposition | 0.17551 | None |  |
| Gauss Jordan | 0.18166 | None |  |
| Gauss Seidel | 0.06632 | 25 |  |

12\*x1 + 3\*x2 - 5\*x3 = 1

x1 + 5\*x2 + 3\*x3 = 28

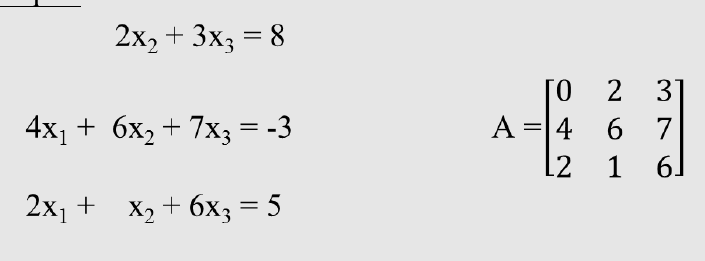
3\*x1 – 7\*x2 + 13 \*x3 = 76

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **Executed Time** | **Number Of Iterations** | **Graph** |
| Gauss Elimination | 0.11245 | None |  |
| LU\_Decomposition | 0.1835 | None |  |
| Jordan Gauss | 0.19651 | None |  |
| Gauss Seidel | 0.29312 | 10 |  |

1. Problematic Functions:

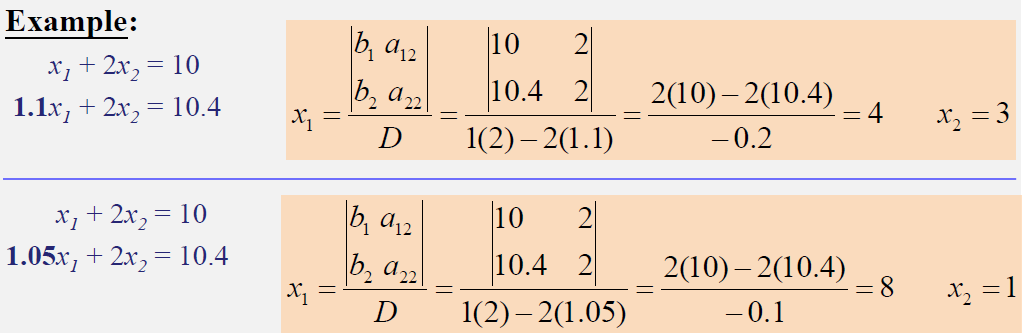
* **Gauss Elimination & Gauss Jordan:**

Division by Zero



Round-Off Error

Ill\_Conditioned System



* **LU Decomposition:**

Could easily be unstable

Does not ﬁnd approximate solutions (least square)

* **Gauss Seidel:**

Not all systems of equations will converge

The coefficient on the diagonal must be at least equal to sum of the other coefficients in the row and at least one row with a diagonal coefficient greater than the sum of the other coefficients in that row

1. Sample Runs:

