Numerical Analysis Project 1

- ## Pseudo Code
  - ***Gauss Elimination***

```
%foward elimination
  for i=1:n-1
    if(A(i,i) == 0)
      err = true;
      disp('division by zero occurred');
      return;
    a=A(i+1:n,i)/A(i,i);
    A(i+1:n,:)= A(i+1:n,:)-a*A(i,:)
    B(i+1:n,:)= B(i+1:n,:)-a*B(i,:)
  %back substitution
  if(A(n,n) == 0)
    err = true;
    disp('division by zero occurred');
    return;
  X(n,:)=B(n,:)/A(n,n);
  for i=n-1:-1:1
    if(A(i,i) == 0)
      err = true;
      disp('division by zero occurred');
      return;
    X(i,:)=(B(i,:)-A(i,i+1:n)*X(i+1:n,:))/A(i,i);
```

  - ***Gauss Jordan***

```
while i<=n
    if X(i,i)==0
      fPrintf('Diagonal element zero');
      C = [];
      err = true;
      break;
    X=Eliminate(X,i);
    i=i+1;
```

  - ***Gauss Seidel***

```
I = 1;
Repeat until ea < es or I > maxi
X(1,:) = (B(1,:) - sum(A(1,2:n).*(X(2:n,:)')))/A(1,1);
    for j=2:n
```

```
      if(A(j,j) == 0)
         err = true;
         disp('division by zero occurred');
         return;
      X(j,:) = (B(j,:) - (sum(A(j,:).*(X(:,:)')) - A(j,j) * X(j,:) )) / A(j,j);
I = I+1;
calcEa(X,Xold);
Xold = X;
```

- ***LU Decomposition***

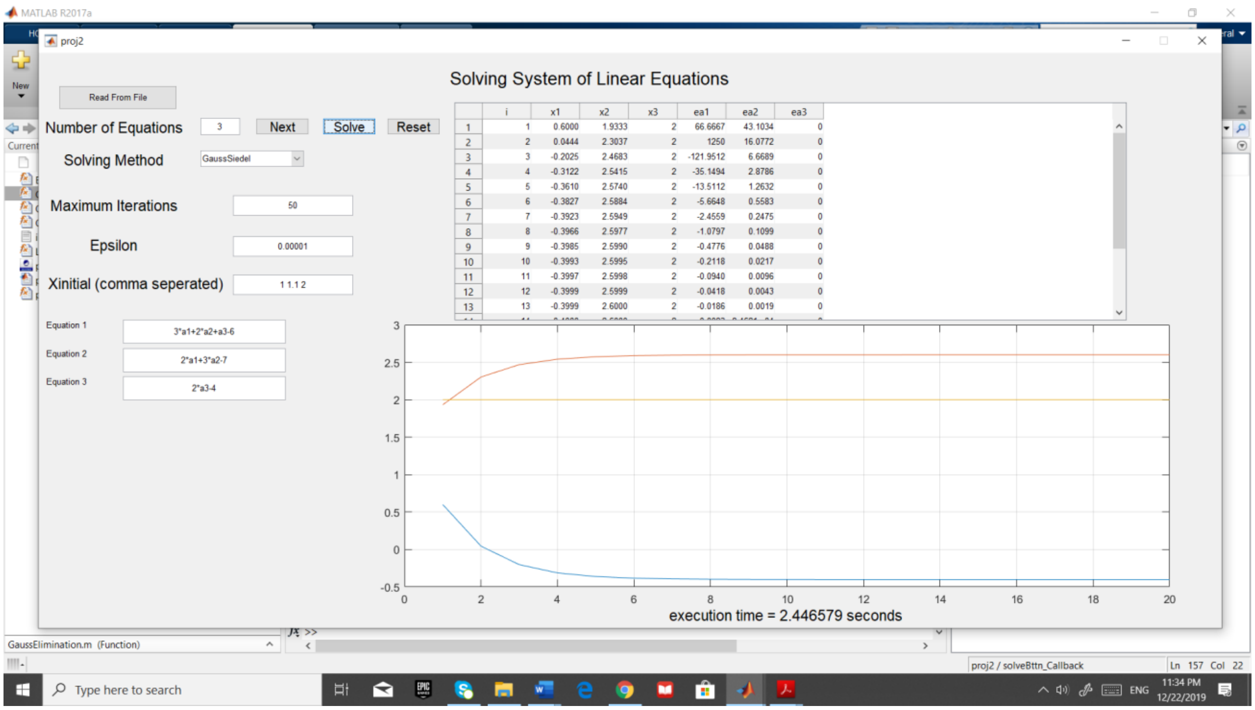
```
   for i=1:n
      for j=1:i
         if(i==j)
            L(i,j) = 1;
         end
      end
   end
   for i=1:n-1
      if(A(i,i) == 0)
         err = true;
         disp('division by zero occurred');
         return;
      end
      L(i+1:n,i) = A(i+1:n,i)/A(i,i);
      A(i+1:n,:)= A(i+1:n,:)-L(i+1:n,i)*A(i,:);
   end
%forward substitution
   Y(1,:)=B(1,:);
   for i=2:n
      Y(i,:)=B(i,:)-L(i,1:i-1)*Y(1:i-1,:);
   end
   %back substitution
   if(A(n,n) == 0)
      err = true;
      disp('division by zero occurred');
      return;
   end
   if(A(n,n) == 0)
      err = true;
      disp('division by zero occurred');
      return;
```

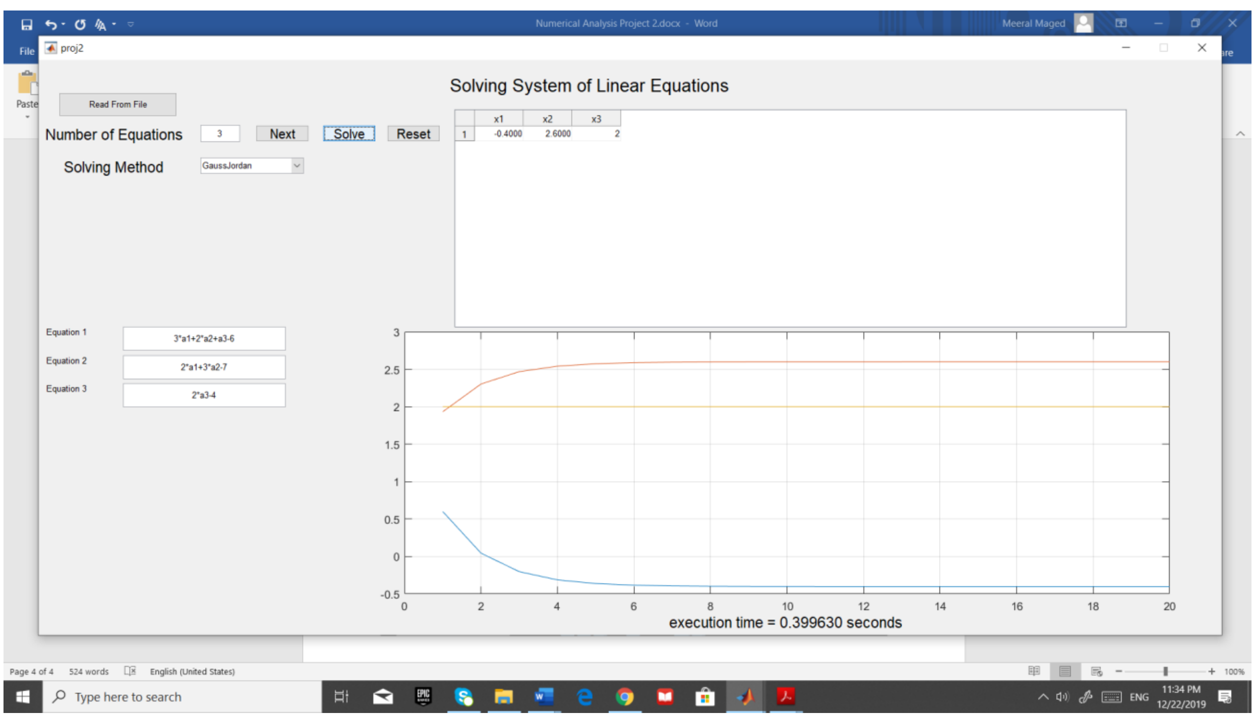
```
end
X(n,:)=Y(n,:)/A(n,n);
for i=n-1:-1:1
   if(A(i,i) == 0)
      err = true;
      disp('division by zero occurred');
      return;
   end
   X(i,:)=(Y(i,:)-A(i,i+1:n)*X(i+1:n,:))/A(i,i);
end
```

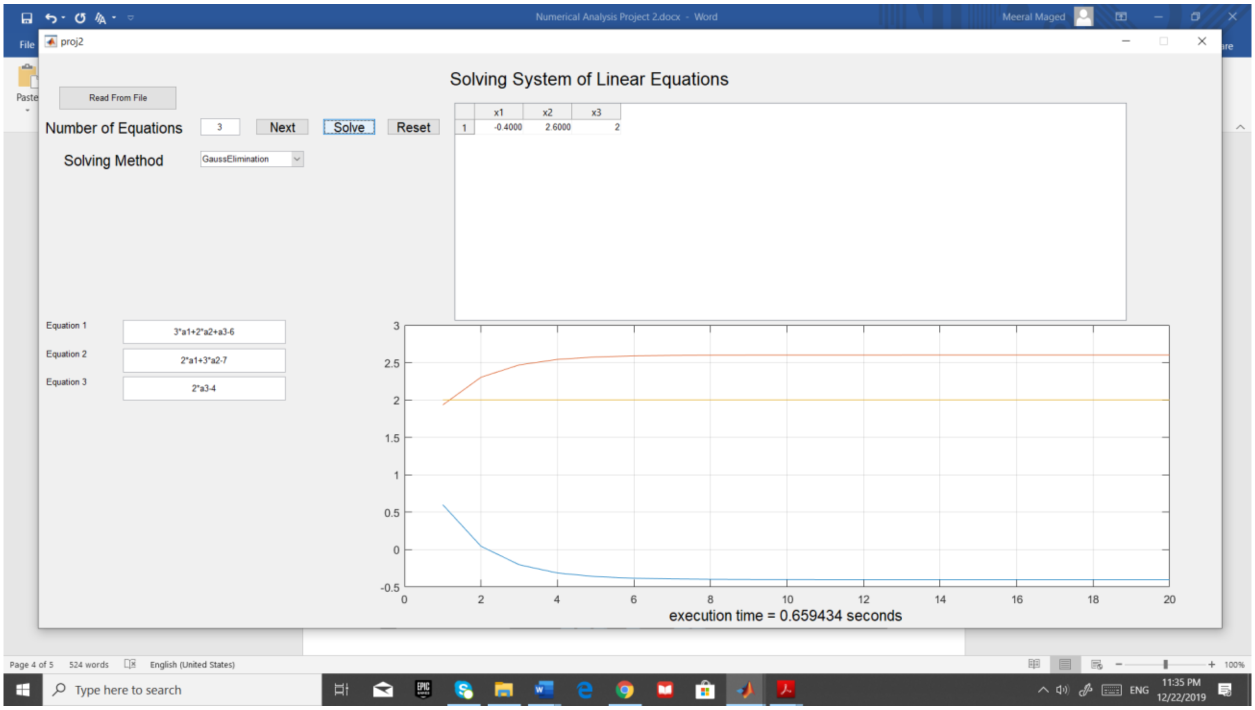
- **<u>DS used</u>**
  - ***Output Matrix from each root finding method***
    Each row represents an iteration in the iterative algorithms. Each column represents a certain variable value. It was very efficient because it could be added in the table object used in our GUI.
  - ***Syms Vector***
    A vector of syms of the size we need to initialize the number of variables based on number of equations.
- **<u>Analysis</u>**
  - ***Example 1:***
    3*a1+2*a2+a3-6
    2*a1+3*a2-7
    2*a3-4
    Gauss-Seidel: finds root after 20 iterations
    The rest of the methods find root equally accurately
  - ***Example 2:***
    10*a1+2*a2-a3-27
    -3*a1-6*a2+2*a3+61.5
    a1+a2+5*a3+21.5
    Gauss-Seidel: finds root after 20 iterations
    The rest of the methods find root equally accurately
- **<u>Problematic functions</u>**
  - ***Functions whose matrices have/end up having zero-valued diagonals***
    In all methods, we divide by an element in the diagonal in each step. Therefore, if a zero diagonal is reached at any point the algorithm is terminated and an error message is diaplayed.
  - ***Reaching xr = 0***
    In calculating ea, when xr = 0 the relative error is infinity even though it doesn’t describe the actual relative error as the root could actually be zero or close to it.

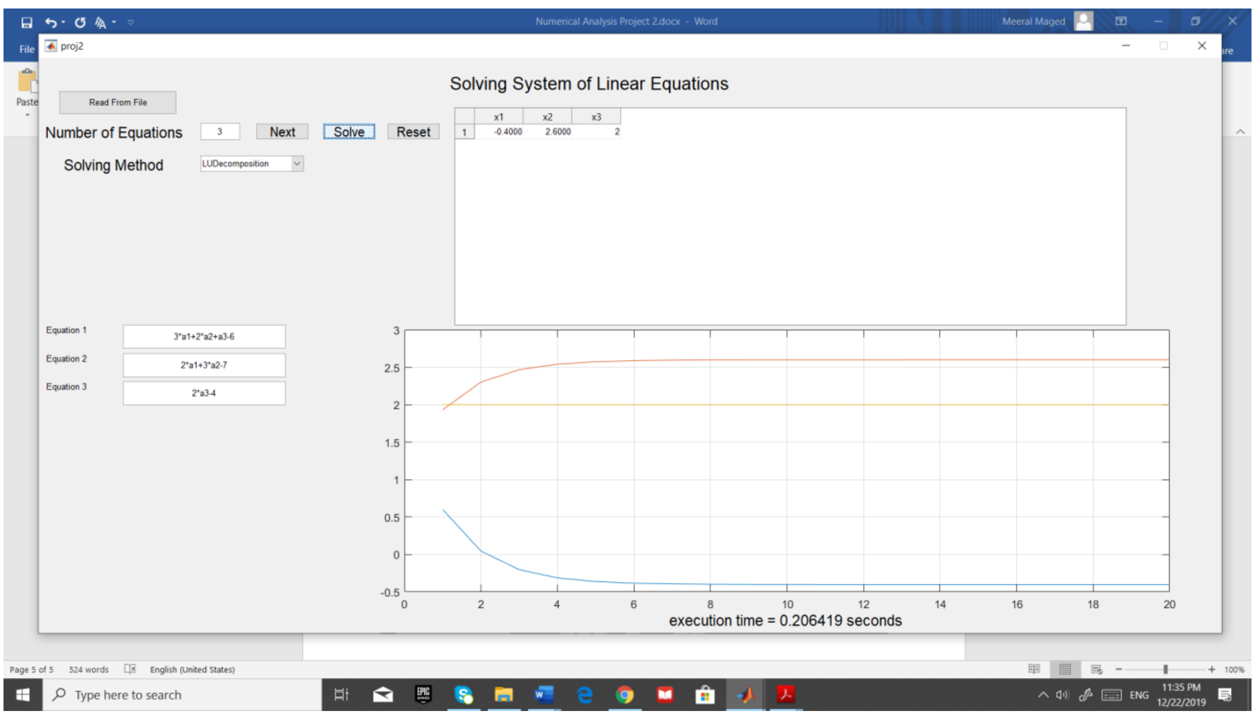
In this program, we don't calculate the ea if xr is zero, it is displayed on the GUI as zero.

- **Sample runs**






Numerical Analysis Project 2.docx - Word
Meeral Maged
proj2
Solving System of Linear Equations
Read From File
Number of Equations
3
Next
Solve
Reset
Solving Method
GaussElimination
x1
x2
x3
1
-0.4000
2.6000
2
Equation 1
3*a1+2*a2+a3-6
Equation 2
2*a1+3*a2-7
Equation 3
2*a3-4
execution time = 0.659434 seconds
Page 4 of 5
524 words
English (United States)
Type here to search
11:35 PM
12/22/2019


Numerical Analysis Project 2.docx - Word
Meeral Maged
proj2
Solving System of Linear Equations
Read From File
Number of Equations
3
Next
Solve
Reset
Solving Method
LUDecomposition
x1
x2
x3
1
-0.4000
2.6000
2
Equation 1
3*a1+2*a2+a3-6
Equation 2
2*a1+3*a2-7
Equation 3
2*a3-4
execution time = 0.206419 seconds
Page 5 of 5
524 words
English (United States)
Type here to search
11:35 PM
12/22/2019