Homework 4

Name: Jackson Lee

Due Date:

Table of Contents


```
clear, clc
```

Problem #1 - Left divide

```
A = [34 15 51
     50 14 36
     38 49 14];          %Material Pit percentages
A = A/100;
B = [6400 5700 3400]';  %Needed Materials
LU = A\B;
fprintf("The Engineer would need to get %.2f m³ from Pit 1, %.2f m³ from Pit 2, and %.2f m³ fro
```

```
The Engineer would need to get 4439.17 m³ from Pit 1, 825.66 m³ from Pit 2, and 9346.73 m³ from Pit 3 to successful
```

Problem #2 - LU factorization

```
Vars = ["AB" "BC" "AD" "BD" "CD" "DE" "CE" "Ax" "Ay" "Ey"];
TRUSS = [0  0  1  0  0  0  0  1  0  0
         1  0  0  0  0  0  0  0  1  0
         0  1  0 3/5 0  0  0  0  0  0
         -1 0  0 -4/5 0 0  0  0  0  0
         0 -1  0  0  0  0 3/5 0  0  0
         0  0  0  0 -1  0 -4/5 0 0  0
         0  0 -1 -3/5 0 1  0  0  0  0
         0  0  0 4/5 1  0  0  0  0  0
         0  0  0  0  0 -1 -3/5 0 0  0
         0  0  0  0  0  0 4/5 0  0  1];

B1 = [0 0 -41 0 0 27 0 0 0 0]';
B2 = [0 0 -58 0 0 92 0 0 0 0]';

A = TRUSS;
A(:,1) = TRUSS(:,8);
A(:,2) = TRUSS(:,9);
A(:,3) = TRUSS(:,4);
A(:,4) = TRUSS(:,1);
A(:,5) = TRUSS(:,2);
```

```
A(:,6) = TRUSS(:,7);
A(:,7) = TRUSS(:,3);
A(:,8) = TRUSS(:,5);
A(:,9) = TRUSS(:,6);
```

a) Solve 1st set of forces

```
L = eye(10);
U = A
```

```
U = 10×10
    1.0000         0         0         0         0         0    1.0000         0 ...
         0    1.0000         0    1.0000         0         0         0         0
         0         0    0.6000         0    1.0000         0         0         0
         0         0   -0.8000   -1.0000         0         0         0         0
         0         0         0         0   -1.0000    0.6000         0         0
         0         0         0         0         0   -0.8000         0   -1.0000
         0         0   -0.6000         0         0         0   -1.0000         0
         0         0    0.8000         0         0         0         0    1.0000
         0         0         0         0         0   -0.6000         0         0
         0         0         0         0         0    0.8000         0         0
```

```
for i = 1:10
    for k = (i+1):10
        if(U(k,i) ~= 0)
            L(k,i) = U(k,i)/(U(i,i));
            U(k,:) = U(k,:) - L(k,i)*U(i,:);
        end
    end
end
d1 = L\B1;
x1 = U\d1;
X(1) = x1(4);
X(2) = x1(5);
X(3) = x1(7);
X(4) = x1(3);
X(5) = x1(8);
X(6) = x1(9);
X(7) = x1(6);
X(8) = x1(1);
X(9) = x1(2);
X(10) = x1(10);
for k = 1:10
    if k == 1
        fprintf("Unknown Force(kN)\n" + ...
            "------- ---------\n" + ...
                "  %s    %8.4f\n", Vars(k),X(k))
    else
        fprintf("  %s    %8.4f\n", Vars(k),X(k))
    end
end
```

```
Unknown Force(kN)
------- ---------
  AB     13.8333
  BC    -30.6250
  AD     41.0000
  BD    -17.2917
  CD     13.8333
  DE     30.6250
  CE    -51.0417
  Ax    -41.0000
  Ay    -13.8333
  Ey     40.8333
```

b) Solve 2nd set of forces

```
d2 = L\B2;
x2 = U\d2;
X(1) = x2(4);
X(2) = x2(5);
X(3) = x2(7);
X(4) = x2(3);
X(5) = x2(8);
X(6) = x2(9);
X(7) = x2(6);
X(8) = x2(1);
X(9) = x2(2);
X(10) = x2(10);
for k = 1:10
    if k == 1
        fprintf("Unknown Force(kN)\n" + ...
            "------- ---------\n" + ...
                "  %s    %8.4f\n", Vars(k),X(k))
    else
        fprintf("  %s    %8.4f\n", Vars(k),X(k))
    end
end
```

```
Unknown Force(kN)
------- ---------
  AB     -7.3333
  BC    -63.5000
  AD     58.0000
  BD      9.1667
  CD     -7.3333
  DE     63.5000
  CE    -105.8333
  Ax    -58.0000
  Ay      7.3333
  Ey     84.6667
```

Problem #3 - Compare methods

```
A = [6.5 1.0 2.1
     1.0 14.4 2.3
     2.1 2.3 3];
```

```
B = [33.54 126.72 36.303]';
```

a) Cramer's rule

```
Ax1 = [B A(:,2:3)];
Ax2 = [A(:,1) B A(:,3)];
Ax3 = [A(:,1:2) B];
Cramer = [det(Ax1)/det(A) det(Ax2)/det(A) det(Ax3)/det(A)]';
```

```
Cramer = 3×1
    2.5800
    7.9500
    4.2000
```

b) Naïve Gauss

```
G = [A B];
factor21 = G(2,1)/G(1,1);
G(2,:) = G(2,:) - factor21*G(1,:);
factor31 = G(3,1)/G(1,1);
G(3,:) = G(3,:) - factor31*G(1,:);
factor32 = G(3,2)/G(2,2);
G(3,:) = G(3,:) - factor32*G(2,:);
Gauss(3) = G(3,4)/G(3,3);
Gauss(2) = (G(2,4)-G(2,3)*Gauss(3))/G(2,2);
Gauss(1) = (G(1,4)-G(1,3)*Gauss(3)-G(1,2)*Gauss(2))/G(1,1);
Gauss = Gauss';
```

```
Gauss = 3×1
    2.5800
    7.9500
    4.2000
```

c) LU factorization

```
[L,U] = lu(A);
d(1) = B(1);
d(2,1) = B(2) - L(2,1)*d(1);
d(3,1) = B(3) - L(3,1)*d(1) - L(3,2)*d(2);
LU(3,1) = d(3)/U(3,3);
LU(2,1) = (d(2) - U(2,3)*LU(3) )/U(2,2);
LU(1,1) = (d(1) - U(1,2)*LU(2) - U(1,3)*LU(3))/U(1,1);
```

```
LU = 3×3
    2.5800    1.0000    2.1000
    7.9500   14.2462    1.9769
    4.2000    0.1388    2.0472
```

d) Cholesky factorization

```
if(eig(A)>0)
    E = "are";
else
    E = "aren't"'
end
```

```
fprintf("All eigenvalues %s positive.\n", E)
```

```
All eigenvalues are positive.
```

```
if(issymmetric(A)==1)
    S = "is";
else
    S = "isn't"'
end
fprintf("The A matrix %s symmetric.\n", S)
```

```
The A matrix is symmetric.
```

```
Uc(1,1) = sqrt(A(1,1));
Uc(1,2) = (A(1,2))/Uc(1,1);
Uc(1,3) = (A(1,3))/Uc(1,1);
Uc(2,2) = sqrt(A(2,2) - Uc(1,2)^2);
Uc(2,3) = (A(2,3) - Uc(1,2)*Uc(1,3))/Uc(2,2);
Uc(3,3) = sqrt(A(3,3) - (Uc(1,3)^2 + Uc(2,3)^2));
dc = Uc'\B;
Cholesky = Uc\dc;
```

```
Cholesky = 3×1
    2.5800
    7.9500
    4.2000
```

e) Gauss-Seidel

```
currErr = 100;
iter = 0;
c = zeros(3,1) ;
target_err = .001;

c1_calc = @(c2,c3) (33.54 - c2 -2.1*c3)/6.5;
c2_calc = @(c1,c3) (126.72 - c1 - 2.3*c3)/14.4;
c3_calc = @(c1,c2) (36.303 - 2.1*c1 - 2.3*c2)/3;

MPRAE = @(new_estimate, old_estimate)...
    abs(new_estimate - old_estimate)./...
    new_estimate * 100;

while max(currErr) > target_err
    c_old = c;
    c(1) = c1_calc(c(2), c(3));
    c(2) = c2_calc(c(1), c(3));
    c(3) = c3_calc(c(1), c(2));
    currErr = MPRAE(c, c_old);
end
GaussSeidel = c;
```

```
GaussSeidel = 3×1
    2.5800
```

```
    7.9500
    4.2000
```

f) Output results

```
fprintf("Method       X1      X2      X3     \n" + ...
        "------------ ------- ------- -------\n" + ...
        "Cramer's     %7.4f %7.4f %7.4f\n" + ...
        "Gauss        %7.4f %7.4f %7.4f\n" + ...
        "LU           %7.4f %7.4f %7.4f\n" + ...
        "Cholesky     %7.4f %7.4f %7.4f\n" + ...
        "Gauss-Seidel %7.4f %7.4f %7.4f\n", Cramer(1), Cramer(2), Cramer(3), Gauss(1), Gauss(2)
```

```
Method       X1      X2      X3
------------ ------- ------- -------
Cramer's      2.5800  7.9500  4.2000
Gauss         2.5800  7.9500  4.2000
LU            2.5800  7.9500  4.2000
Cholesky      2.5800  7.9500  4.2000
Gauss-Seidel  2.5800  7.9500  4.2000
```

g) Extra credit: Is the Gauss-Seidel method guaranteed to converge? Why/Why not?

ANSWER HERE