Computational Materials Engineering Lab - Experiment 1

Sesha Sai Behara
MM14B008

Vishal S
MM14B048

1.

$$-3y + 9z = 36$$

$$2x + 17z = 35$$

$$-x - y + 34z = 34$$

In matrix representation,

$$\begin{bmatrix} 0 & -3 & 9 \\ 2 & 0 & 17 \\ -1 & -1 & 34 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 36 \\ 35 \\ 34 \end{bmatrix}$$

On solving the above equation using the following MATLAB code the result is,

```
Ainv = inv(A);
%Solving for the answer
constants = Ainv*B;
% Printing the solution
constants
```

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9 \\ -9 \\ 1 \end{bmatrix}$$

2. Using Kirchoff's voltage law for the given circuit, we get the following set of linear equations

$$-72I_1 + 20I_2 + 50I_3 = -10$$

$$20I_1 - 46I_2 + I_3 + 25I_4 = -5$$

$$50I_1 + I_2 - 116I_3 + 50I_5 = 5$$

$$25I_2 - 55I_4 = 3$$

$$50I_3 - 65I_5 = 0$$

Using matrix representation we have,

$$\begin{bmatrix} -72 & 20 & 50 & 0 & 0 \\ 20 & -46 & 1 & 25 & 0 \\ 50 & 1 & -116 & 0 & 50 \\ 0 & 25 & 0 & -55 & 0 \\ 0 & 0 & 50 & 0 & -65 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix} = \begin{bmatrix} -10 \\ -5 \\ 5 \\ 3 \\ 0 \end{bmatrix}$$

On solving the above equation using the MATLAB code shown below we get,

```
Ainv = inv(A);
%Solving for the answer
constants = Ainv*B;
% Printing the solution
constants
```

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix} = \begin{bmatrix} 0.328 \\ 0.2988 \\ 0.1529 \\ 0.0813 \\ 0.1176 \end{bmatrix}$$

3. In matrix representation the given set of linear equations boil down to,

$$\begin{bmatrix} 9 & 6 & 12 \\ 6 & 13 & 11 \\ 12 & 11 & 26 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 17.4 \\ 23.6 \\ 30.8 \end{bmatrix}$$

Using LU decomposition the above equation becomes,

$$LA = \begin{bmatrix} 17.4 \\ 23.6 \\ 30.8 \end{bmatrix} \tag{1}$$

where A is given by,

$$A = U \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}$$

The LU decomposition is done using the following code ,

```
N = 3;
ans = zeros(N,1);
b = [17.4,23.6,30.8];
array1 = zeros(N);
L = zeros(N);
```

```
U = zeros(N);

array1 = [9,6,12;6,13,11;12,11,26];

% LU decomposition of the array1
for i = 1:N
    L(i,i) = 1;
    U(1,i) = array1(1,i);
end

for i = 2:N
    L(i,1) = array1(i,1)/ U(1,1);
end

U(2,2) = array1(2,2) - L(2,1)*U(1,2);
L(3,2) = (array1(3,2) - L(3,1)*U(1,2) )/(U(2,2));
U(2,3) = array1(2,3) - L(2,1)*U(1,3);
U(3,3) = array1(3,3) - L(3,1)*U(1,3) - L(3,2)*U(2,3);
```

The L and U matrix as obtained from the code is as follows,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 4/3 & 1/3 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 9 & 6 & 9 \\ 0 & 9 & 3 \\ 0 & 0 & 9 \end{bmatrix}$$

Using the L matrix obtained first we solve equation 1 which gives A. Using the obtained A and U matrices we solve equation 2 which gives the final result.

```
Linv = inv(L);
Uinv = inv(U);
ans1= Linv*transpose(b);
ans = Uinv*ans1
```

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.6 \\ 1.2 \\ 0.4 \end{bmatrix}$$

The obtained result is compared with the answer obtained from inverse built in function in MATLAB.

```
ans2 = inv(array1) * transpose(b)
```

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.6 \\ 1.2 \\ 0.4 \end{bmatrix}$$

4. Considering the path of the missile to be parabolic we have,

$$z = ax^2 + bx + c \tag{3}$$

Given three points coordinates through which the missile passes, (15,50), (60,135) and (105,90)

Substituting the above points in equation 3 yields the following set of linear equations with variables a, b and c.

$$225a + 15b + c = 50$$
$$3600a + 60b + c = 135$$
$$11025a + 105b + c = 90$$

In matrix representation we have,

$$\begin{bmatrix} 225 & 15 & 1 \\ 3600 & 60 & 1 \\ 11025 & 105 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 50 \\ 135 \\ 90 \end{bmatrix}$$

Upon solving the equation using the MATLAB code we get,

```
constants = zeros(3,1);

A = zeros(3,3);
B = zeros(3,1);
Z = [50,135,90];
X = [15,60,105];
% Populating the coefficent matrix
for i = 1:3
    A(i,1) = X(i)*X(i);
    A(i,2) = X(i);
    A(i,3) = 1.0;
    B(i) = Z(i);
end
Ainv = inv(A);
%Solving for the answers
constants = Ainv*B;
% Printing the solution
constants
```

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -0.0321 \\ 4.2963 \\ -7.2222 \end{bmatrix}$$

Substituting the values of a, b and c in equation 3 we get

$$z = -0.0321x^2 + 4.2963x - 7.2222$$

To get where the missile hits the ground we need to substitute z as zero in equation 3,

$$0 = -0.0321x^2 + 4.2963x - 7.2222$$

This is a quadratic equation and can be solved

```
target  = -constants(2)
- sqrt((constants(2)*constants(2))
- (4*constants(1)*constants(3)));
target = target/(2*constants(1))
```

Upon solving for roots of the above equation we get,

$$x = 132.138m$$

Horizontal velocity of the missile,

```
vx = (X(2) - X(1) )/( 10);
```

$$u_x = (105 - 60)/10 = 4.5ms^{-1}$$

After the third measurement the time taken by the missile to hit the ground is,

```
time = (target - X(3))/vx
```

$$t = (132.138 - 105)/4.5 = 6.0307s$$

5. In this Question we are trying to solve the temperature profile across the rod. Dirichlet boundary condition is applied at both ends of the rod. There is also heat loss to the surrounding through the sides of the rod.

   Thus, the final equation is

$$\frac{d^2T}{dx^2} - h(T - T_a) = 0 \ , \tag{4}$$

   The above equation 4 is then descritised according to finite difference formulation and then solved using TDMA algorithm.

   The temperature profile is then plotted across the length.

   The code is again run by changing the number of nodes and the temperature profile is compared with the previous run.

```
% Initialsing variables
delx = 2.0;
N = 4;
T0 = 40.0;
T5 = 200.0;
h = -0.02;
Ta = 10.0;
temp = zeros(N,1);
X = zeros(N,1);
ae = zeros(N,1);
aw = zeros(N,1);
ap = zeros(N,1);
su = zeros(N,1);

% Populating the coefficient matrix
for i = 1:N
    ap(i) = -2/(delx*delx) + h;
    ae(i) = 1.0/(delx*delx);
    aw(i) = 1.0/(delx*delx);
    su(i) = h*Ta;
    if ( i == 1)
        X(i) = delx;
    else
        X(i) = X(i-1) + delx;
    end
end

% TDMA
su(1) = su(1) - aw(1)*T0;
su(N) = su(N) - ae(N)*T5;
ae(N) = 0.0;
aw(1) = 0.0;

for i = 1:N
    if(ap(i) == 0)
        disp ("Error ap becomes zero");
    end
end

ae(1) = ae(1)/ap(1);
su(1) = su(1)/ap(1);
ap(1) = 1.0;
for i = 2:N
  ap(i) = ap(i) - ae(i-1)*aw(i)/ap(i-1) ;
  su(i) = su(i) - su(i-1)*aw(i)/ap(i-1) ;
  aw(i) = 0;
```

```
  ae(i) = ae(i) / ap(i);
  aw(i) = aw(i) / ap(i);
  su(i) = su(i) / ap(i);
  ap(i) = 1.0;
end
% Back-Substitution
temp(N) = su(N)/ap(N);
for i = 2 : N
    j = N-i + 1;
    temp(j) = (su(j) - ae(j)*temp(j+1))/(ap(i)) ;
end
% For plotting properly
Temp_1 = zeros(N+2,1);
X_1 = zeros(N+2,1);
X_1(1) = 0 ;
Temp_1(1) = T0;
Temp_1(N+2) = T5;
for i = 1:N
    Temp_1(i+1) = temp(i);
    X_1(i+1) = X(i) ;
end
X_1(N+2) = X_1(N+1) + delx ;
plot (X_1,Temp_1,'LineWidth',4);
hold on ;
title ("Temperature Profile across length")   ;
ylabel("Temperature ");
xlabel("Length");
legend("h = 0.02 , N = 9",'location','northwest');
box  on ;
grid on ;
```

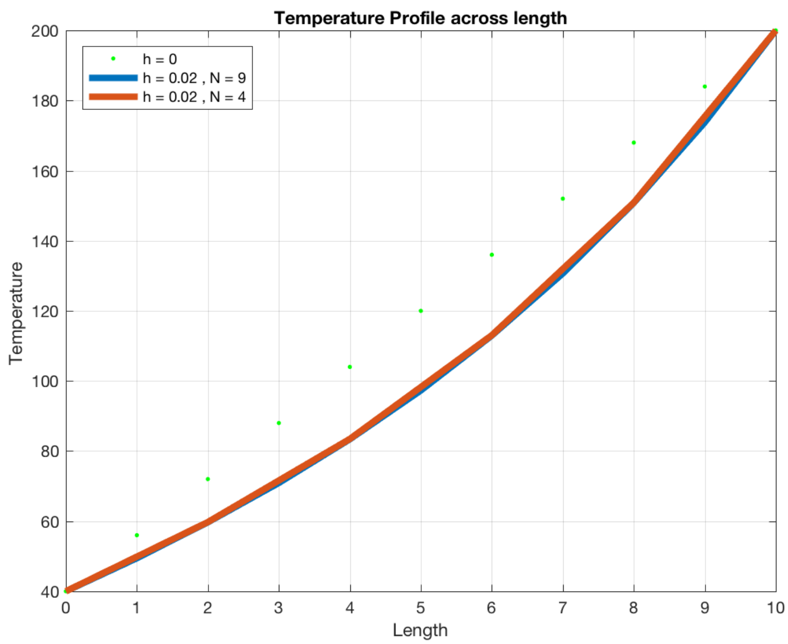

Figure 1: Temperature Profile across the rod.