EXPERIMENT – I

Anirban Mukherjee

16BIT0200

STRESS DISTRIBUTION IN A TOWER BRIDGE

Aim:

Aim: Calculating and visualizing the Eigen values of stress matrix for simply supported beam.

Problem Statements:

Find principal stresses for a twodimensional simply supported beam by finding the eigenvalues of the stress matrix with variable components.

Matlab Commands:

Sym – create symbolic objects

Charpoly – characteristic polynomial of matrix

Roots – Polynomial roots

Eig – Eigen values and eigen vectors of a symbolic matrix A

Transpose – Transpose matrix

Description of Physical Experiment:

• An elastic body is subjected to applied loadings, stresses are created inside the body.

• In general these stresses often vary in complicated ways from point to point and from plane to plane within the structure.

• To help characterize this situation, stresses are normally defined with respect to a given coordinate system.

• The illustration below shows that at a typical point within a loaded body, the state of stress can be characterized on a small cube of material defined with respect to a Cartesian coordinate system.

• These nine components are called the stress components, with σ_x σ_y σ_z referred to as normal stresses and τ_{zy} τ_{zx} τ_{xy} called the shearing stresses.

Connection to Mathematics:

$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix}$$

(Here σ refers to normal stress and τ refers to shear stress.)

• The number of components and some other transformation properties, the stress can be expressed as a 3 x 3 matrix

• Since the shearing stresses have the equalities , the stress matrix is symmetric.

• If we changed the orientation of a particular plane the normal stress component will vary.

• There exists a special orientation where the normal stress will be a maximum, and these are called principal planes and the normal stresses acting on them are called the principal stresses.

The general three-dimensional case, the theory to determine principal stresses and the planes on which they act is formulated by the eigenvalue problem

$[\sigma]\{n\} = \lambda\{n\}$

where σ is the stress matrix, $\{n\}$ is the principal direction vector and λ (the eigenvalue) is the principal stress. Thus solving the eigenvalue problem will determine up to three distinct principal stresses and the corresponding three principal directions. It turns out for this application (3x3, symmetric real matrix) the principal directions are mutually orthogonal.

The shear stress components will vanish on these three principal planes and so for a coordinate system that is aligned with the principal directions the stress matrix takes on the simplified diagonal form

$$\sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

where σ_1, σ_2, σ_3 are the problem's eigenvalues (roots of the characteristic equation) and are generally referred to as the principal stresses.

<u>Question:</u>

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

a. Characteristic polynomial of *A*
b. Roots of Characteristic polynomial of *A*
c. Eigen values of *A*
d. Eigen vectors of *A*
e. Eigen values of A^{-1}
f. Eigen values of A^T
g. Eigen values of B = $A^2+3A+2I$

Code:

```
A=[3 0 -1; 0 1 0; 2 0 0]
Eigenvalues_of_A=eig(A)
[X,D]= eig(A)
p=poly(A)
r = roots(p)
sum_of_eigenvalues = sum(r)
trace =trace(A)
product_of_eigenvalues_A=prod(r)
Det_A = det(A) nt
Inv_A =inv(A)
Eigenvalues_of_invA= eig(Inverse_A)
Eigenvalues_Transpose_of_A= eig(A')
Eigenvalues_of_B= eig(A^2+3*A+2*eye(3))
```

Answer:

```
A =

     3     0    -1
     0     1     0
     2     0     0
```

```
Eigenvalues_of_A =

     2
     1
     1


X =

    0.7071    0.4472         0
         0         0    1.0000
    0.7071    0.8944         0


D =

     2     0     0
     0     1     0
     0     0     1


p =

     1    -4     5    -2


r =

   2.0000 + 0.0000i
   1.0000 + 0.0000i
```

```
   1.0000 - 0.0000i


sum_of_eigenvalues =

    4.0000


trace =

     4


product_of_eigenvalues_A =

    2.0000


Det_A =

     2


Inv_A =

         0         0    0.5000
         0    1.0000         0
   -1.0000         0    1.5000


Eigenvalues_of_invA =
```

```
    0.5000
    1.0000
    1.0000


Eigenvalues_Transpose_of_A =

     2
     1
     1


Eigenvalues_of_B =

    12
     6
     6
```

<u>Question:</u>

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 6 & -1 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

a. Characteristic polynomial of *A*
b. Roots of Characteristic polynomial of *A*
c. Eigen values of *A*
d. Eigen vectors of *A*
e. Eigen values of A^{-1}

f. Eigen values of A^T

g. Eigen values of B = $A^2+3A+2I$

Code:

```
A=[1 2 1; 6 -1 0; -1 -2 -1]
Eigenvalues_of_A=eig(A)
[X,D]= eig(A)
p=poly(A)
r = roots(p)
sum_of_eigenvalues = sum(r)
rrace =trace(A)
product_of_eigenvalues_A=prod(r)
Det_A = det(A)
Inv_A =inv(A)
Eigenvalues_of_invA= eig(Inv_A)
Eigenvalues_Transpose_of_A= eig(A')
Eigenvalues_of_B= eig(A^2+3*A+2*eye(3))
```

Output:

```
A =

     1     2     1
     6    -1     0
    -1    -2    -1


Eigenvalues_of_A =

   -4.0000
    3.0000
    0.0000


X =

    0.4082   -0.4851   -0.0697
   -0.8165   -0.7276   -0.4180
   -0.4082    0.4851    0.9058


D =

   -4.0000         0         0
         0    3.0000         0
         0         0    0.0000
```

```
p =

    1.0000    1.0000  -12.0000    0.0000


r =

   -4.0000
    3.0000
    0.0000


sum_of_eigenvalues =

   -1.0000


rrace =

    -1


product_of_eigenvalues_A =

  -3.1168e-15


Det_A =

     0

Warning: Matrix is singular to working precision.
> In demo at 10

Inv_A =

   Inf   Inf   Inf
   Inf   Inf   Inf
   Inf   Inf   Inf

Error using eig
Input to EIG must not contain NaN or Inf.

Error in demo (line 11)
Eigenvalues_of_invA= eig(Inv_A)
```

Question:

$$\sigma_x = -\frac{3P}{4c^3}(L-x)y, \sigma_y = 0, \tau_{xy} = -\frac{3P}{4c^3}(c^2-y^2)$$

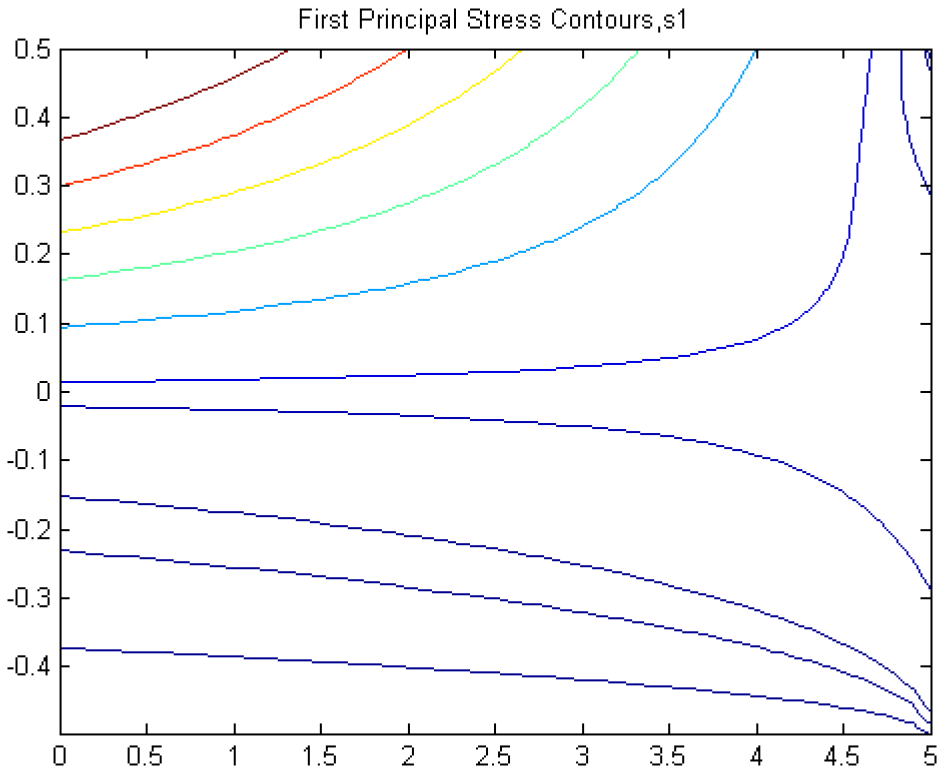
Where P denotes the force 2c denotes the height and 2L denotes the length of the beam. Write the stress matrix and hence calculate the principal stresses.

Draw the stress distribution in the beam using contour plot

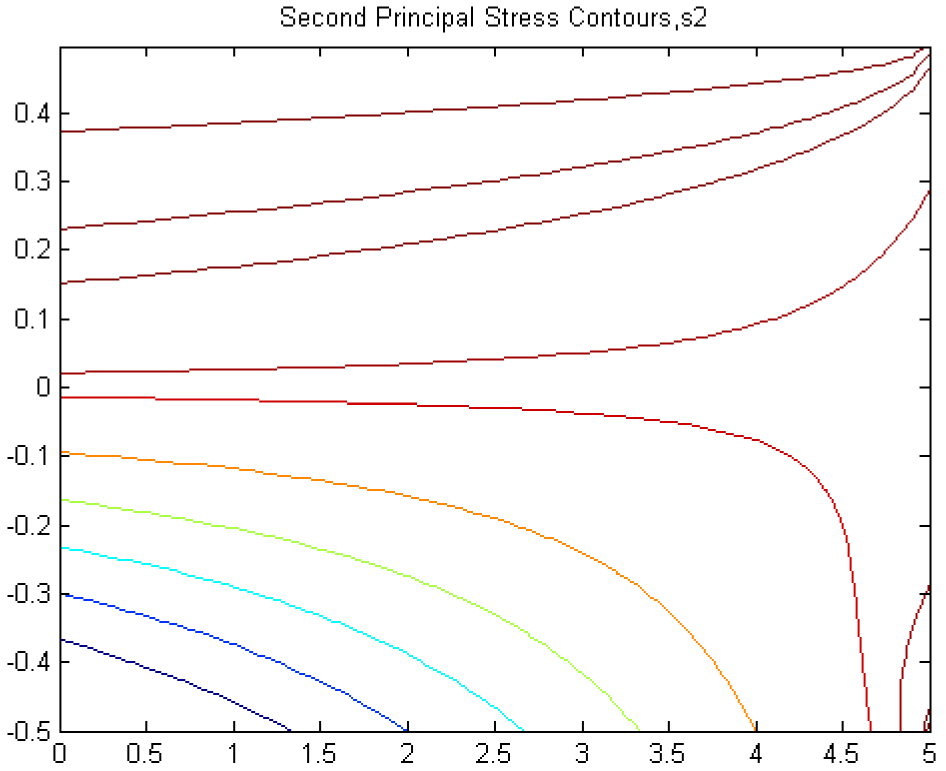
Code:

```
clc;
clear all;
clf;
c=0.5; L=5; P=1;
x=[0:0.1:L];
y=[-c:0.01:c];
[X,Y]=meshgrid(x,y);
sx=(3/(4*c^3))*(L-X).*Y;
sy=zeros(length(y),length(x));
txy=-(3/(8*c^3))*(c^2-Y.^2);
for i=1:length(y)
    for j=1:length(x)
s=[sx(i,j), txy(i,j); txy(i,j), sy(i,j)];
p=eig(s);
s1(i,j)=p(2);
s2(i,j)=p(1);
    end
end
figure (1)
contour(X,Y,s1,[0.01,0.05,0.1,0.5,1,3,5,7,9,11])
title('First Principal Stress Contours,s1')
axis tight
figure (2)
contour(X,Y,s2,[-0.01,-0.05,-0.1,-0.5,-1,-3,-5,-7,-9,-11])
axis tight
title('Second Principal Stress Contours,s2')
```

Output:



First Principal Stress Contours,s1
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5



My Work:

The methodology can be used for making a database which can be further developed as some software for multiple engineering applications.

For example, we can create a database on population or say a player's performance. We input the data as a matrix or equation depending on the equation. We can find the eigenvalues and eigenvectors of the input then.

Then we can interpret the data according to our requirements, for example, we can calculate the rate of growth or population, a player's average on a certain kind of pitch; from this we can infer many possibilities on where to concentrate on an issue and improve upon it.