MT2527 (Computational Engineering 2-Part 1) ASSIGNMENT 2 - D1

Participants:

- Chetan Bhandare
  941009-3273
- Akhil Goud Mora
  950728-6954
- Raghavendra Machipeddi
  950606-6894
- Pulicherla Yashpal Surendhar Goud
  950419-4995

Abstract
Reynolds Equation is a partial differential equation which can be used to describe the pressure distribution in nearly any type of fluid film bearing, a bearing type in which the bounding bodies are fully separated by a thin layer of liquid or gas. The problem deals with a sector thrust bearing to which non-dimensional pressure distribution, non-dimensional load capacity and non-dimensional relative power loss is to be determined.

Finite Element Method is applied to Reynold's equation to determine the pressure distribution of the sector thrust bearing thereby calculating the load capacity and relative power loss. To simplify the problem, the bearing surface is discretized into number of elements thus simplifying the dynamics of the problem and determining the pressure for each element.

The pressure distribution in the bearing is shown in a three dimensional plot and an isobar plot which are obtained after the execution of FEM on Reynolds equation in MATLAB.

Contents

1. Notations:

ω – Angular velocity
φ – Sector circumference
k_o – constant
h – height function
h_o – non dimensional height function
R_{in} – inner radius of the sector
R_{out} – outer radius of the sector
F – load capacity of the bearing
F_o – non dimensional load capacity of the bearing
h_{min} – minimum height
P – power loss
P_o – non dimensional power loss
p – pressure
p_o – non dimensional pressure
f – relative power loss
f_o – non dimensional relative power loss
η – viscosity of the fluid
r – radius
θ – angle

2. Introduction:

A bearing is a machine which constrains relative motion into required motion, reduce friction between moving parts. There are different types of bearings such as roller bearings, journal bearings, magnetic bearings, flexure bearings, fluid bearing etc.

The sector thrust bearing is a type of fluid bearing which support their loads solely on a thin layer of liquid or gas. Like other bearings they permit rotation between parts, but they are designed to support a predominately axial load. They are used in a high speed and high accurate applications like turbine shafts, power plants in automobile industries etc. where ordinary ball bearings would have short life or cause high noise and vibration.

The sector thrust bearing in this problem has 6 identical pads. The upper part of the bearing has an angular velocity ω and the lower part is stationary. The pressure is considered to be zero at the boundaries to start of the solution for the problem.

A two dimensional Reynolds equation with finite element approach is used to find the pressure distribution along the bearing surface. Solving Reynold's equation analytically is a very time consuming and a complex procedure. The decrease in element size would give us a relatively quicker solution but the accuracy is compromised. As mesh is further refined, the convergence gets better.

The discussion of the problem and the calculation of various parameters in MATLAB can be seen in further chapters.

3. Problem

3.1. Problem Statement:

A Sector Thrust Bearing with six pads is shown to the left in figure 1. The geometry of one pad seen from above is shown to the right in figure 1. All pads are identical and takes up a circumference of $\phi = 0.7$ radians. The rest of the circumference is for lubricant grooves. At the inner and outer radius and in the grooves, the pressure is equal to the ambient pressure (atmospheric). The upper part of the bearing has an angular velocity ω. The lower part is stationary. The viscosity of the lubricant is η.

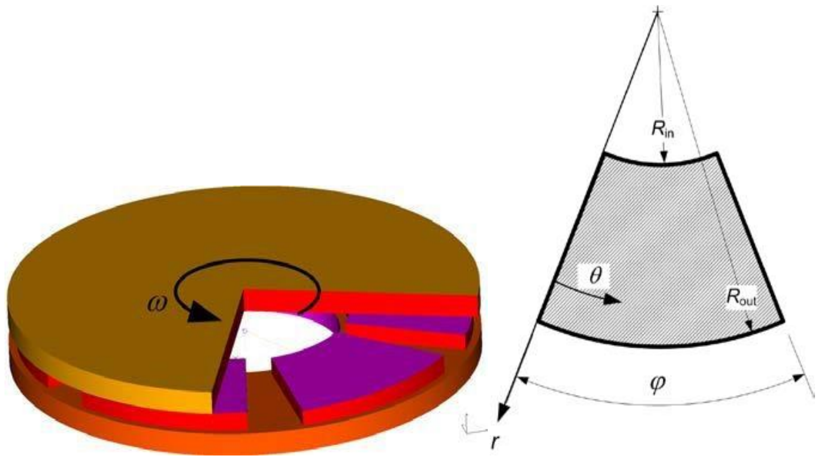

Figure 1 Sector thrust bearing

Each pad has the following height function (in polar coordinates with coordinate system origin at bearing center line)

$$h(\theta) = h_{min}\left(K_o + 1 - \frac{K_o}{\varphi}\theta\right), \qquad \frac{\varphi}{2} \leq \theta \leq \varphi$$

$$h = h_{min}(K_0 + 1), \qquad 0 \leq \theta \leq \frac{\varphi}{2}$$

The non-dimensional groups used in the problem are:

1. $r_0 = \frac{r}{R_{out}};\ R_0 = \frac{R_{in}}{R_{out}};\ \theta_0 = \theta;\ F_0 = \frac{Fh_{min}{}^2}{n\omega R_{out}{}^4};\ h_0 = \frac{h}{h_{min}};\ P_0 = \frac{Ph_{min}}{n\omega^2 R_{out}{}^4};$

$$f_0 = \frac{f}{\omega h_{min}} = \frac{P_0}{F_0}$$

The given data is $k_0=2$ and $R_0=0.4$.

To determine:

- Non-dimensional maximum pressure
- Non-dimensional load capacity, F_0
- Non-dimensional relative power loss, f_0
- To show the non-dimensional pressure distribution in a three-dimensional plot and in an iso-bar plot.

- The position of θ for the switch of height function that gives the highest load capacity of the bearing.

3.2. Method:

Reynolds equation is used to find the pressure distribution in the given sector thrust bearing. The non-dimensional Reynolds equation for the sector thrust bearing is as follows in polar coordinates.

$$\frac{\partial}{\partial r_0}\left(r_0 h_0^3 \frac{\partial p_0}{\partial r_0}\right) + \frac{\partial}{\partial \theta_0}\left(\frac{h_0^3 \partial p_0}{r_0 \partial \theta_0}\right) = 6r_0 \frac{\partial h_0}{\partial \theta_0}$$

FEM is applied to Reynolds equation to determine the pressure distribution of the bearing. Iso parametric mapping is used for controlling the body irrespective of its geometry to obtain iso parametric elements using the coordinated from main domain.
Using gauss integration load matrices and load vectors are obtained.

Shape functions are calculated

$$N_1 = -\frac{1}{4}(\xi - 1)(\eta + 1)$$

$$N_2 = \frac{1}{4}(\xi + 1)(\eta + 1)$$

$$N_3 = \frac{1}{4}(\xi - 1)(\eta - 1)$$

$$N_4 = -\frac{1}{4}(\xi + 1)(\eta - 1)$$

Jacobian matrix is derived from derivatives of shape functions.

Calculation of element matrices and load vectors for a single element in the mesh are as follows:

- Shape functions and their respective derivatives are calculated for one of 4 points of one element.
- Obtain the Cartesian coordinates of a global matrix i.e x_o, y_o by Gauss integration method and calculate the non-dimensional height h_o using the obtained x_o.
- With the derived shape functions derivatives Jacobean matrix transpose, its inverse and determinant of [J] are calculated.
- [B] matrix is obtained using transpose of Jacobean matrix and shape function derivatives.
- $[f^1] = \frac{h_o^3}{12r_o}[B]^T[B]|J|$ is used to obtain the element stiffness matrix for the point.
- $[g^1] = \frac{h_o r_o}{2}\begin{bmatrix} B_{11} \\ B_{12} \\ B_{13} \\ B_{14} \end{bmatrix}|J|$ is used to obtain corresponding element matrix for the point.
- Calculation for all the 4 points are done in same way and summarised for globalised stiffness matrix and load vectors for the element into $[K^1]$ and $\{k_{Uo}\}^1$.
- The above procedure is used for whole bearing domain depending on the mesh.

- Pressure distribution is calculated by giving boundary conditions by the following relation.
  $[K]\{\widetilde{p_0}\} = \{k_{U0}\} + \{q_0\}$

  where $dA=|J|d\,\xi\,d\,\eta$

- $K_{ij}=\iint_{-A}^{A} \frac{h_o^3}{12r_o} \left(\frac{\partial N_i}{\partial x_0}\frac{\partial N_j}{\partial x_0} + \frac{\partial N_i}{\partial y_0}\frac{\partial N_j}{\partial y_0}\right) dA$

- $k_{Uoi}=\iint_{-A}^{A} \frac{h_o r_o}{2}\frac{\partial N_i}{\partial x_0} dA$

- $q_{oi}=\oint N_i(q_{or}n_r + q_{o\theta}n_\theta)dl$

  $q_{o\theta}=\frac{h_o r_o}{2} - \frac{h_o^3}{12r_o}\frac{\partial p_0}{\partial \theta_0}$

  $q_{or}= - \frac{h_o^3}{12r_o}\frac{\partial p_0}{\partial r_0}$

K_{ij} is Coefficient matrix, p is pressure vector, q is Constant vector, k_{Uoi} is element load vector. Using the given height function and solving in matlab we the non-dimensional max pressure $p_{o\ max}$.

Pressure distribution is determined by

$$\frac{\partial}{\partial r_0}\left(r_0 h_0^3 \frac{\partial p_0}{\partial r_0}\right) + \frac{\partial}{\partial \theta_0}\left(\frac{h_0^3 \partial p_0}{r_0 \partial \theta_0}\right) = 6r_0\frac{\partial h_0}{\partial \theta_0}$$

Load capacity of a single pad is given by

$$F_o = \int_{R_{in}}^{R_{out}} \int_0^{2\pi} p_o * r d\theta dr$$

The pressure distribution for all the elements of bearing domain is found in MATLAB and is substituted in load capacity.

The relative power loss is

$$f = \frac{P}{F} \qquad where\ P = power\ loss\ ; F = Load\ capacity$$

$$P = T * \omega \quad where\ T = Rotational\ Torque = -\eta\frac{r * \omega}{h} - \frac{h}{2}\frac{dp}{dx}$$

$$\omega = Angular\ velocity$$

The pressure distribution is plotted in MATLAB in a 3D plot and an isobar plot by using ‘surf’ and ‘countourf’ respectively.

3.3. Solution

The equation used for this task is non dimensional Reynolds equation for a sector thrust bearing is

$$\frac{\partial}{\partial r_0}\left(r_0 h_0^3 \frac{\partial p_0}{\partial r_0}\right) + \frac{\partial}{\partial \theta_0}\left(\frac{h_0^3 \partial p_0}{r_0 \partial \theta_0}\right) = 6 r_0 \frac{\partial h_0}{\partial \theta_0}$$

The height function for each pad is

$$\mathrm{h}(\theta) = \mathrm{h_{min}}\left(\mathrm{K_o} + 1 - \frac{\mathrm{K_o}}{\varphi}\theta\right), \qquad \frac{\varphi}{2} \le \theta \le \varphi$$

$$\mathrm{h} = \mathrm{h_{min}}(\mathrm{K_0} + 1), \qquad 0 \le \theta \le \frac{\varphi}{2}$$

which is converted to non-dimensional results

$$\mathrm{h_0}(\theta) = \mathrm{h_{min}}\left(\mathrm{K_o} + 1 - \frac{\mathrm{K_o}}{\varphi}\theta\right), \ \frac{\varphi}{2} \le \theta \le \varphi$$

$\mathrm{h_o} = \mathrm{h_{min}}(\mathrm{K_0} + 1), \ 0 \le \theta \le \frac{\varphi}{2}$

In FEM the shape functions and their corresponding derivatives are derived respectively for a 4 node iso parametric element.

$$N_1 = -\frac{1}{4}(\xi - 1)(\eta + 1)$$

$$N_2 = \frac{1}{4}(\xi + 1)(\eta + 1)$$

$$N_3 = \frac{1}{4}(\xi - 1)(\eta - 1)$$

$$N_4 = -\frac{1}{4}(\xi + 1)(\eta - 1)$$

Shape function derivatives are used to find transpose of Jacobean matrix. The Jacobian matrix yields way to the [B] matrix from which we calculate the element stiffness matrix and element load vector for the whole operation domain created.

Size of the stiffness matrix depends on the product of the elements along X and Y axes. The local and global numbering is considered and matrices are assembled to form a global stiffness matrix.

The mesh is created in MATLAB according to the bearing and its height function at respective areas. T above mentioned procedure of FEM is implemented in MATLAB and the non-dimensional pressure distribution, non-dimensional load capacity and non-dimensional relative power loss are determined in MATLAB itself by the given formulas.

load capacity of a single pad is given by

$\mathrm{F_0} = \iint P_0 R_0 dr_0 \, d\theta_0$

The pressure distribution for all the elements of bearing domain is found in MATLAB and is substituted in Non-dimensional load capacity.

The Non-dimensional relative power loss is

$$f = \frac{P}{F} \qquad where\ P = power\ loss\ ; F = Load\ capacity$$

$$P = T * \omega \quad where\ T = Rotational\ Torque = -\eta \frac{r * \omega}{h} - \frac{h}{2} \frac{dp}{dx}$$

$$\omega \ = Angular\ velocity$$

The position of θ for the switch of height function is found out using angular sweep.

3.4 Results:

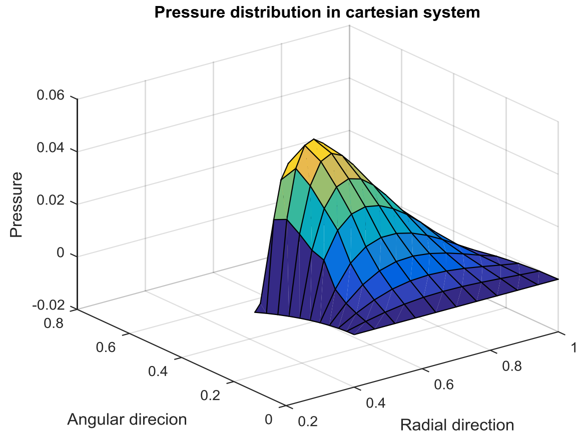

Figure 2 Pressure Distribution in Cartesian coordinate system

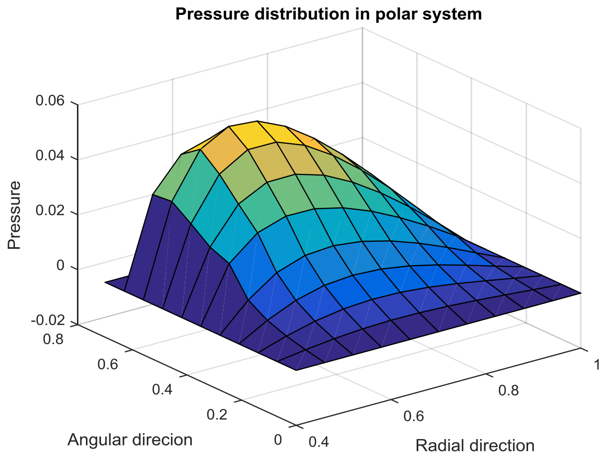

Figure 3 Pressure Distribution in Polar Coordinate system

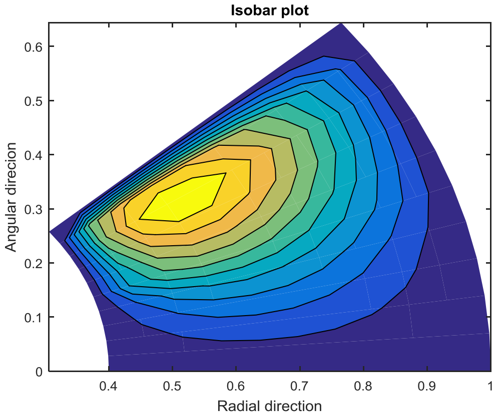

Figure 4 Isobar Plot in Cartesian coordinate system

The Non-dimensional Maximum pressure is 0.0553
The Non-dimensional load carrying capacity is 0.0422 for one pad.

The Non-dimensional load carrying capacity is 0.2532 for whole bearing.
The Non-dimensional Relative power loss is 8.4%

The position of θ for switch of height function to get maximum non-dimensional load carrying capacity is at $\frac{\varphi}{5}$ and the value is 0.0467.

3.5 Discussions and Conclusions:

The experience from the present assignment made us conclude that the computational effort for the given problem was less when solved using FEM compared to that of FDM.
In FDM one should take care of adjacent values and conditions of a node which makes it complicated when implementing in MATLAB.FEM is efficient and more certain when compared to FDM. The approach of convergence is very faster in FEM when compared to FDM. Also the accuracy in FEM is better compared to FDM. Computation time is less in case of FEM than FDM. It would be much less when solved in in any FEA package like COMSOL.

Validation:
The convergence of the result helped us in the validation of the solution.

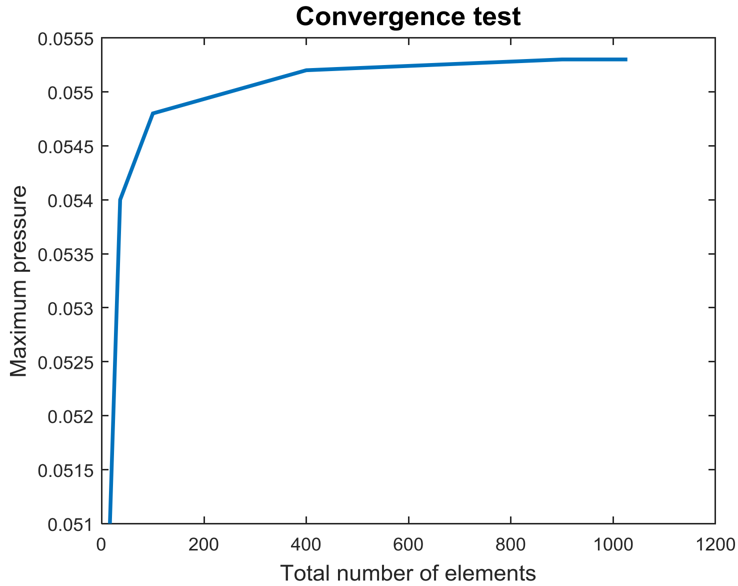

Figure 5 Convergence test

As seen from the above figure the maximum value of non-dimensional pressure converged to 0.0553 for the grid size of 32*32.

4 References

1. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19760019428.pdf
2. Computational engineering by Goran Broman

5 Self evaluation

Name	Much More	More	Equal	Less	Much Less
Chetan			x		
Akhil			x		
Raghavendra			x		
Surendhar			x		

Appendix

```
%% Height functions code

clc

clear all;

close all;

N1=10;

N2=10;

a=N1/2;

i=1:a;

b=zeros(N2,a);

c=zeros(1,a);d=0;

for j=0:N2-1

for k=1:a

c(k)= ((N1-(N1-i(k)))+(j*N2)) ;

end

for l=a:-1:1

d=[c(l) d];

end

d=d(1:a);

b(j+1,:)=d;

end
```

main code

```
clc;
clear all;
close all;
tt
ko=2;
Ro=1;
phi=0.7;
N1=10;
N2=10;
[m,n] = meshgrid(0:0.7/N1:0.7, 0.4:0.6/N2:1);
xnum=(m');
ynum=(n');
[M,N]=meshgrid(0:1:N1-1,0:1:N2-1);
Ntrp=(N');
numel(Ntrp)
for i =1:numel(Ntrp)
nd1(i)=i+Ntrp(i);
nd2(i)=i+Ntrp(i)+1;
nd3(i)=i+Ntrp(i)+N1+1;
nd4(i)=i+Ntrp(i)+N1+2;
end
cc=0;h0=zeros(1,numel(Ntrp));
for ele=1:numel(Ntrp)
theta=[xnum(nd1(ele)) xnum(nd2(ele)) xnum(nd3(ele)) xnum(nd4(ele))]';
rad=(1/Ro)*[ynum(nd1(ele)) ynum(nd2(ele)) ynum(nd3(ele)) ynum(nd4(ele))]';
f=zeros(4,4);
g=zeros(4,1);
for i=-1:2:1
```

```
for j=-1:2:1
eta=i*1/sqrt(3);
pxi=j*1/sqrt(3);
Ns=0.25*[ -(eta-1)*(pxi+1) (eta+1)*(pxi+1) (eta-1)*(pxi-1) -(eta+1)*(pxi-1) ];
dNs=0.25*[-(pxi+1) (pxi+1) (pxi-1) -(pxi-1) ; -(eta-1) (eta+1) (eta-1) -(eta+1) ];
theta0=Ns*theta;
rad0=Ns*rad;
for l=1:((N1*N2)/2)
if (cc==0)
if(ele==b(l))
h0(ele)=3;
cc=cc+1;
else
h0(ele)=3-((2*theta0)/phi);
end
end
end
cc=0;
JT=dNs*[theta rad];
IJ=inv(JT);
detJ=det(JT);
B=IJ*dNs;
f=f+((h0(ele)^3)/(rad0*12))*detJ*B'*B;
g=g+(h0(ele)*rad0/2)*detJ*B(1,:)';
end
end
Kst(:,:,ele)=f;
Ku(:,:,ele)=g;
```

```
end
%%
R=zeros(numel(xnum),numel(xnum));
for i=1:numel(Ntrp)
for r=1
R(nd1(i),nd1(i),i)=Kst(r,r,i);
R(nd1(i),nd2(i),i)=Kst(r,r+1,i);
R(nd1(i),nd3(i),i)=Kst(r,r+2,i);
R(nd1(i),nd4(i),i)=Kst(r,r+3,i);
end
for r=2
R(nd2(i),nd1(i),i)=Kst(r,r-1,i);
R(nd2(i),nd2(i),i)=Kst(r,r,i);
R(nd2(i),nd3(i),i)=Kst(r,r+1,i);
R(nd2(i),nd4(i),i)=Kst(r,r+2,i);
end
for r=3
R(nd3(i),nd1(i),i)=Kst(r,r-2,i);
R(nd3(i),nd2(i),i)=Kst(r,r-1,i);
R(nd3(i),nd3(i),i)=Kst(r,r,i);
R(nd3(i),nd4(i),i)=Kst(r,r+1,i);
end
for r=4
R(nd4(i),nd1(i),i)=Kst(r,r-3,i);
R(nd4(i),nd2(i),i)=Kst(r,r-2,i);
R(nd4(i),nd3(i),i)=Kst(r,r-1,i);
R(nd4(i),nd4(i),i)=Kst(r,r,i);
end
```

```
end
B=zeros(numel(xnum),numel(xnum));
for i=1:numel(Ntrp)
B=B+R(:,:,i);
end
M=zeros(numel(xnum),1);
for i=1:numel(Ntrp)
M(nd1(i),1,i)=Ku(1,1,i);
M(nd2(i),1,i)=Ku(2,1,i);
M(nd3(i),1,i)=Ku(3,1,i);
M(nd4(i),1,i)=Ku(4,1,i);
end
N=zeros(numel(xnum),1);
for i=1:numel(Ntrp)
N=N+M(:,1,i);
end
N1=N1+1;
N2=N2+1;
for i=1:1:N1
B(i,i)=B(i,i)*1e7;
end
for i=1:(N1):N1*(N2-1)+1
B(i,i)=B(i,i)*1e7;
end
for i=N1:N1:N1*N2
B(i,i)=B(i,i)*1e7;
end
for i=N1*(N2-1)+1:N1*N2
```

```
B(i,i)=B(i,i)*1e7;

end

% calculating the value of pressure at each node

finpre=B\N;

r=1;

for i=1:N2

for j=1:N1

pres(i,j)=finpre(r);

r=r+1;

end

end

pres=rot90(pres');

finpre=pres';

surf(ynum,xnum,finpre);

ylabel('Angular direcion');

xlabel('Radial direction');

zlabel('Pressure');

title('Pressure distribution in polar system');

figure()

[x,y]=pol2cart(xnum,ynum);

surf(x,y,finpre);

ylabel('Angular direcion');

xlabel('Radial direction');

zlabel('Pressure');

title('Pressure distribution in cartesian system');

figure()

contourf(x,y,finpre);

ylabel('Angular direcion');
```

```
xlabel('Radial direction');

zlabel('Pressure');

title('Isobar plot');

%% Non-dimensional load carrying capacity and relative power loss

LCC=trapz(trapz(finpre*ynum*(0.7/N1)*(0.6/N2)))

nn1=abs(-1-(1/2*(diff(finpre))));

nn2=abs(-1/2.5-((2.5/2)*(diff(finpre))));

nn=[nn1; nn2];

ffo=(trapz(trapz(nn*ynum*(0.7/N1)*(0.6/N2)))/10);

g=ffo/LCC
```