Assignment Part -1
Computational engineering 2 (MT2527)

KESANA RAM KIRAN (9501273230)

MADDALA PRANAY RAJ REDDY(9407261230)

MORA AKHIL(9507286954)

RUDRARAJU VENKATA SAI KRISHNA VARMA(9409192557)

ABSTRACT:

The criteria of improvisation of load carrying capacity is very important in the design and optimization of a bearing. Maximum pressure, non-dimensional load capacity F_0 and non-dimensional relative power loss f_0 are determined using finite difference method for the given bearing shown in figure 1. The problem is solved for different grid sizes. Non-dimensional pressure distribution and isobar plots are produced.

Contents

1. Notations

ω	angular velocity
η	viscosity
r	radius
R_{out}	outer radius
R_{in}	inner radius
h_{min}	minimum height
p	pressure
F	Load capacity

2. Introduction

The given thrust bearing comprises of six identical pads as shown below. The upper part of the bearing is rotating with an angular velocity of ω and the lower part is stationary. The following are the height functions for the given bearing. There are lubricating grooves in between each pads. The pressure is ambient pressure at lubricating grooves, inner and outer radius.
A thrust bearing is a particular type of rotary rolling bearing which take loads by using balls, rollers or pads in between two bearing rings. The advantage of using these bearings is that they are reliable, durable and have a reasonable load bearing capacity.

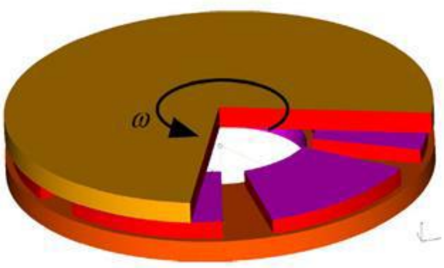
In this assignment, we will be solving a typical thrust bearing using the Finite Difference Method , solve for the pressure and also study other values obtained from pressure such as Load capacity and Power loss.

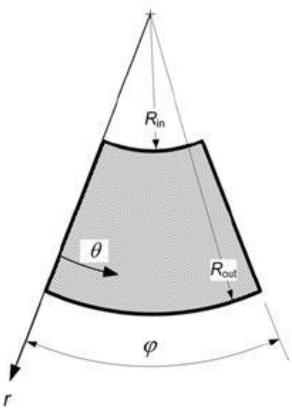
The following non dimensional groups are to be used:

$r_0=\frac{r}{R_{out}}$,$\theta_0=\theta$,$h_0=\frac{h}{h_{min}}$,$p_0=\frac{ph_{min}^2}{\eta\omega R_{out}^2}$, $R_0=\frac{R_{in}}{R_{out}}$, $F_0=\frac{Fh_{min}^2}{\eta\omega R_{out}^4}$,$f_0=\frac{f}{\omega h_{min}}=\frac{P_0}{F_o}$

3.1 Problem statement

We need to determine the maximum pressure ,non-dimensional load capacity and non-dimensional pressure distribution using FDM and the non-dimensional pressure distributions is to be plotted in both 3D and iso-bar plot . The given bearing along with the height function variation with respect to the angular direction is shown below:





$h(r,\theta)=h_{min}+k_0h_{min}\frac{R_{out}(\varphi-\theta)}{r\varphi}\ for\ 0<\theta<\frac{\varphi}{3}$

$h(r,\theta)=h_{min}\frac{R_{out}}{r}\ for\ \frac{\varphi}{3}<\theta<\varphi$

The parameters given for performing analysis:

$k_o=2$,Inner radius (R_{in}) =0.04m, Outer radius (R_{out}) =0.1m, Viscosity (η)= 0.01 $\frac{Ns}{m^2}$ *and*

Minimum height (h_{min}) = 55×10^{-6}m, Angular velocity of upper part of bearing, ω =28 rad/s

3.2 Solution Method

Reynold's equation (in polar co-ordinates) is solved using finite difference method to obtain pressure profile. The Finite difference formulation of equation 1 using (2) and (3) leads to equation 4.

Our approach in MATLAB based on the equation given below was with the idea of forming the coefficient matrix from the Finite difference formulation of the Reynold's equation and finally inverting the coefficient matrix obtained. The main task behind this implementation is to keep track of the surrounding nodes at each and every node.

The non-dimensional Reynolds equation in polar co-ordinates is as follows:

$$\frac{\partial}{\partial r_0} r_o \,(h_0)^3 \frac{\partial}{\partial r_0} p_0 + \frac{\partial}{\partial \theta_0} \left(\frac{h_0^3}{r_0} \frac{\partial}{\partial \theta_0} p_0\right) = 6 r_0 \frac{\partial h_0}{\partial \theta_0} \text{ – equation1}$$

The above equation will be solved using the basic finite difference formulations,

$$\left(\frac{dy}{dx}\right)_i = \frac{y_{i+1/2} - y_{i-1/2}}{\Delta x} - (2)\ ;\ \left(\frac{d^2y}{dx^2}\right)_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} - (3)$$

The partial derivatives in the Reynold's equation are substituted with the finite difference equations. The following is the FDM equation of the sector thrust bearing in polar coordinates.

$$\frac{(r_0 h_0^3)_{i+\frac{1}{2},j}}{\Delta r_0^2} (p_0)_{i+1,j} + \frac{(r_0 h_0^3)_{i-\frac{1}{2},j}}{\Delta r_0^2} (p_0)_{i-1,j} + \frac{\frac{(h_0^3)}{r_o}_{i,j+\frac{1}{2}}}{\Delta \theta_0^2} (p_0)_{i,j+1} + \frac{\frac{(h_0^3)}{r_o}_{i,j-\frac{1}{2}}}{\Delta \theta_0^2} (p_0)_{i,j-1} - \left[\frac{(h_0^3)_{i+\frac{1}{2},j}}{\Delta r_0^2} + \frac{(h_0^3)_{i-\frac{1}{2},j}}{\Delta r_0^2} + \frac{\frac{(h_0^3)}{r_o}_{i,j+\frac{1}{2}}}{\Delta \theta_0^2} + \frac{\frac{(h_0^3)}{r_o}_{i,j-\frac{1}{2}}}{\Delta \theta_0^2}\right] (p_0)_{i,j+1} = 6 r_o \left(\frac{\partial h_0}{\partial \theta_0}\right)_{i,j} \text{ – equation 4}$$

The load capacity for each pad in the bearing is calculated by using the formula,

$$F_o = \int_0^{\theta_o \text{ at end}} \int_{r_o \text{ at start}}^{r_0 \text{ at end}} p_o r_o dr_o \, d\theta_0 = \int_0^{\psi} \int_{\frac{r_{in}}{r_{out}}}^{\frac{r_{out}}{r_{out}}} p_o r_o dr_o \, d\theta_0$$

Here, p is the non-dimensional pressure and from the non-dimensional load capacity we calculate the dimensional load capacity as given below:

$$F_0 = \frac{F h_{min}^2}{\eta \omega R_{out}^4} \Rightarrow F = \frac{F_o \eta \omega R_{out}^4}{h_{min}^2} \quad \text{with units in Newton}$$

Now, we get the total load capacity of the bearing by multiplying the above value by the number of pads we have.

The non-dimensional relative power loss is calculated by taking the ratio of power loss with that of the non-dimensional load capacity. For the non-dimensional power loss, we have that,

$Po{=}To, which\ is\ the\ non{-}dimensional\ frictional\ force$

The above step is written because the Power loss is just the frictional force multiplied by the angular velocity.

The value of the dimensional frictional force is taken as the product of the shear stress times the area of the bearing where magnitude of shear stress is given by,

$$\tau = \eta \frac{U}{h} + \frac{h}{2} \frac{\partial p}{\partial x}$$

In the above equation, we make use of the non-dimensional forms $U{=}r\omega$, $r{=}r_o R_{out}$, $dp{=}\frac{dp_o \eta \omega R_{out}^2}{h_{min}^2}$, $h{=}h_o h_{min}$, and we take out the dimensional terms from the equation and shift them to the LHS, which gives the non-dimensional Frictional force

3.3 Solution

The given sector thrust bearing consists of 6 pads which are identical so, we analyze one pad and then generalize to the whole bearing.
As said before, the number of elements in the y direction were taken as a multiple of 3 since the given height function has a step at an angle of $\varphi/3$ which implies that we take the solution in such a way that we have a mesh line at this particular angle so that we can specify the variation in height function using the topology between the points towards the upper and lower direction of the given node (that is we choose the second height function at the upper point and the first height function at the lower node in MATLAB). Here, we implement the first height function even at this mesh line, to see what happens if we have a sudden step variation. However, we have almost the same pressure distribution if we change the height function a little bit at this particular step

The solution work was done so that we transform the co-ordinate system from Cartesian to polar system which means we have a rectangular grid and then at the end we transform our co-ordinates to the Cartesian system using the function pol2cart in Matlab.

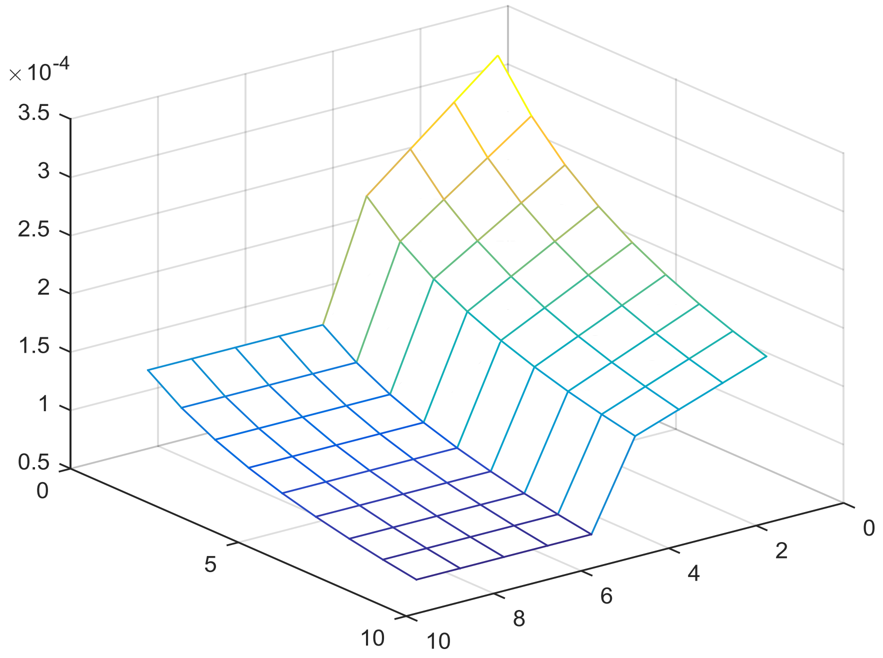


Figure 2 Height variation of the pad

Different conditions are assigned to the nodes depending on the position of the particular set of nodes, four conditions for the nodes at four corners of the pad, four conditions for the nodes lying on the four edges of the pad, a condition for the nodes lying on the step, a condition for the nodes lying over the inner region of the pad from angle 0 to ϕ /3 and another condition for the nodes lying over the inner region of the pad from angle ϕ /3 to ϕ.

Therefore total of 11 conditions are used to assign different height functions to the nodes depending on the position of them.

surf	To get a 3D surface plot
contourf	to get a filled 2-D contour plot
sum	To get sum of vector
inv	To get inverse of a matrix
meshgrid	Used to get a full grid
pol2cart	To get Cartesian coordinatres from polar coordinates

3.4 Result

3.4.1 Non-dimensional Pressure distribution graphs in 3-dimensional plots and in iso-bar plots.

For 9 elements

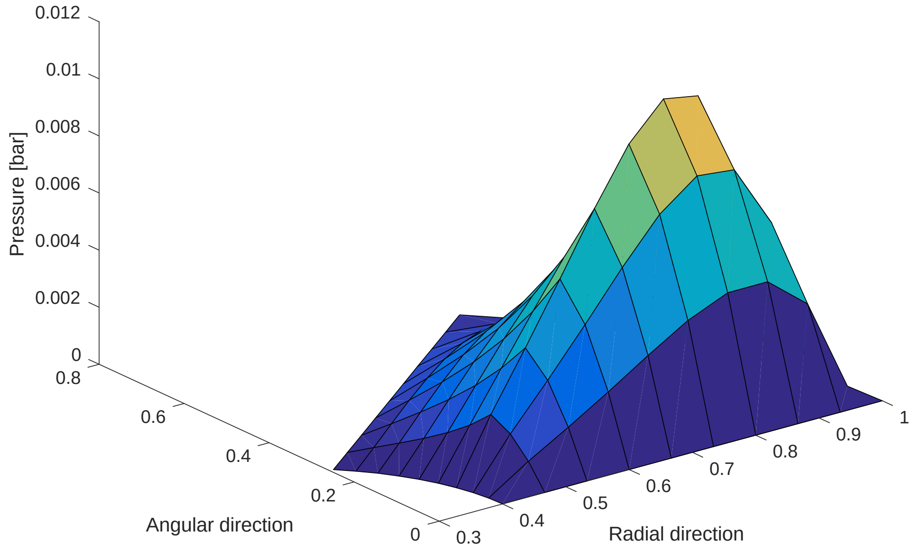


Figure 3 Pressure distribution (9 elements)

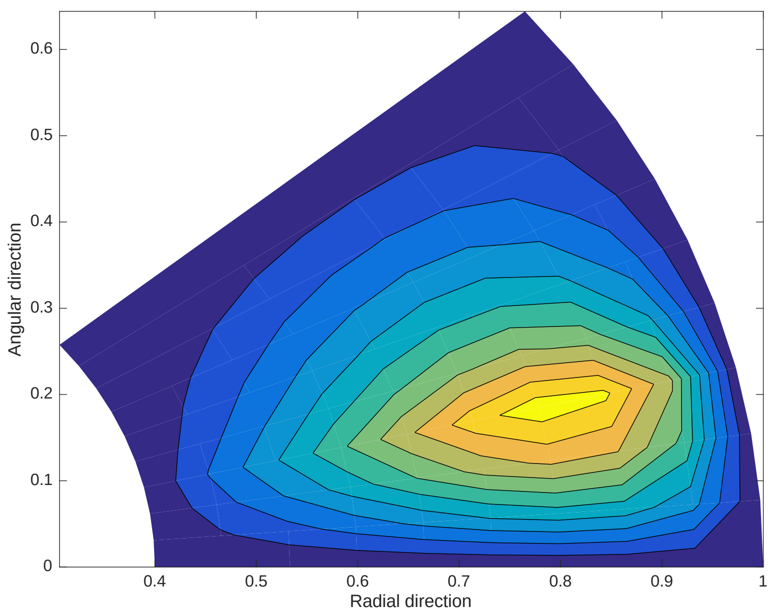


Figure 4 Isobaric Pressure distribution(9 elements)

For 12 elements

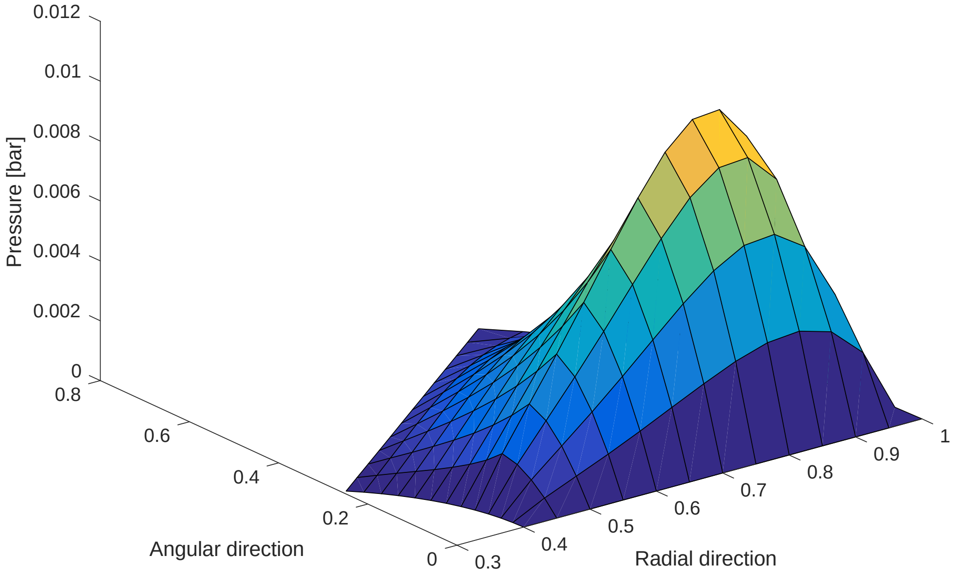


Figure 5 Pressure distribution(12 elements)

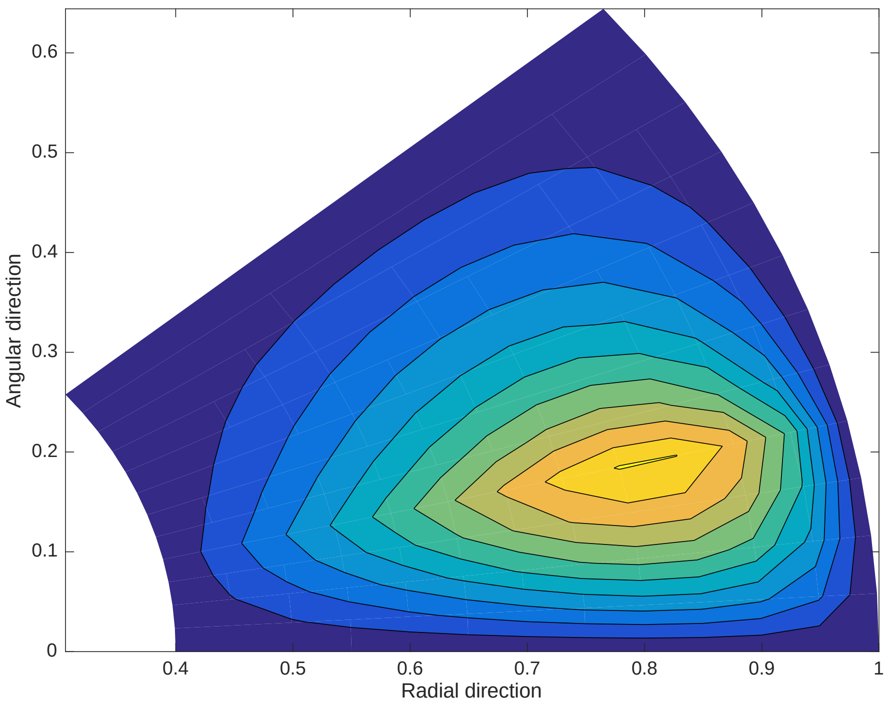


Figure 6 Isobaric pressure distribution(12 elements)

For 15 elements

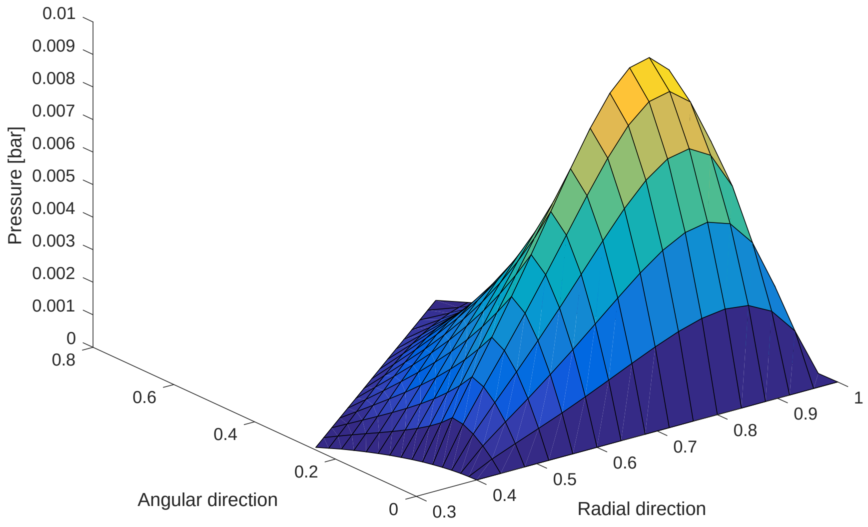


Figure 7 Pressure distribution(15 elements)

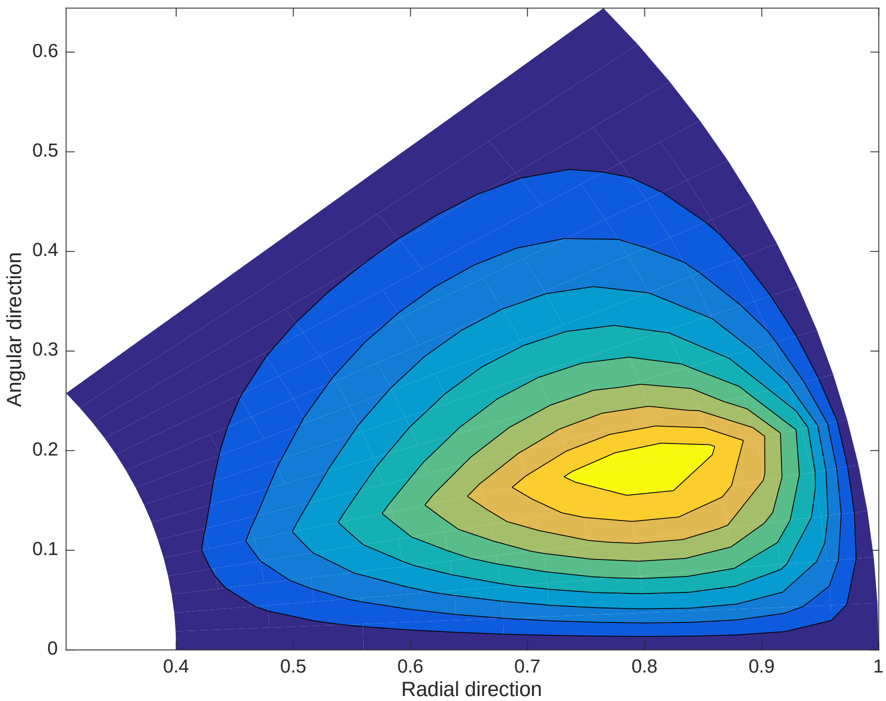


Figure 8 Isobaric pressure distribution(15 elements)

3.4.2 Maximum pressure

Maximum pressure is calculated using the following formula:

$p=\frac{p_o \eta \omega R_{out}^2}{h_{min}^2}$

The maximum pressure obtained is 9.046 kPa for 15×15 elements.
The maximum pressure obtained is 9.3182 kPa for 12×12 elements.

3.4.3 Non-dimensional load capacity (F0)

The dimensional load capacity was calculated using the equation stated before has the following values:
For 15×15 elements Load capacity is 7.5925N
For 12×12 elements Load capacity is 7.8073N

The non-dimensional load capacity is 8.2026e-4 for 15×15 elements.
The non-dimensional load capacity is 8.4346e-4 for 12×12 elements.

The following is for whole bearing:
Dimensional load capacity is 45.555 Newton for 15×15 elements and 46.843 N for 12×12 elements.
Non-dimensional load capacity is 49.2156e-4 for 15×15 elements and 50.6076e-4 for 12×12 elements.

3.4.4 Non-dimensional relative power loss (f0)

The relative power loss obtained is 8.6 percent for 15×15 elements when calculated for single pad, therefore for the whole bearing the power loss is same, as the area increases the work and also the frictional loss increases.
f_0=8.6% -8.8%

3.5 Discussion and conclusion

Due to the formation of step in the height profile of the bearing, there is a sudden increase in pressure which can be seen in the results. The refinement of mesh grid has also impact on the numerical value of the pressure. The results obtained and convergence of the solution from the mesh grid were looked for a maximum of 15×15 elements (dividing the domain into 225 elements) because we think that in calculation of coefficient matrix for the pressure, as we increase the number of elements, becomes lower (without a specific control regarding to topology since we are keen to looking the boundary line of the grid) and decreases as we increase the number of subdomains.

The distribution of pressure obtained from this method was quite uniform, even if we increase the number of elements which gives a particular definite value for the load capacity and Power loss in all the cases.
However, as the number of elements are increased while analysis (for the specific range of elements considered where the requirements also meet a good computational time) the maximum pressure is varying slightly and so is the non-dimensional load capacity.

The relative power loss is calculated for single pad and then it is generalized to the whole bearing, the obtained power loss is 8% which is a sensible range for sector thrust bearing of the given size.
Since the height function varies in both radial as well as angular direction the prediction of the pressure range was somewhat cumbersome and cannot be predicted quantitatively at this point of time, however, the pressure distribution (with a maxima at the step given) seems to be an adequate result.

Finally, the computational efficiency was quite good in the range of elements taken as said above, but, the computational time was seen to increase because of the several number of conditions included in our approach which was based on identifying the node numbers automatically.

4. References

1. Broman. G. In . *Computational Engineering.* Göran ed. Karlskrona: BTH,2003.

2. *Wikipedia*

5. Self-evaluation

Name of group members	MM	M	E	L	ML	A
KESANA RAM KIRAN			X			
MADDALA PRANAY RAJ REDDY			X			
MORA AKHIL			X			
RUDRARAJU VENKATA SAI KRISHNA VARMA			X			

6. Appendix

Matlab code:

```
clc
clear all
% FDM- IMPLEMENTED BY SPECIFYING NUMBER OF ELEMENTS TAKEN AS A MULTIPLE OF
% '3' TO CONSIDER THE VARIATION OF HEIGHT FUNCTION.. better in range of
% lower number of nodes -- 10 to 19 (or 9 to 18 elements for computational
efficiency)
nnodex=13
nnodey=13
ko=2;rout=0.1;rin=0.04;psi=0.7
hmin=55*10^(-6);roi=0.4; %rin/rout=Ro
n=nnodex-4
roe=1; %rout/rout
r1=zeros((nnodey-2)*(nnodex-2),1);m1=1;
c=zeros((nnodey-2)*(nnodex-2),(nnodey-2)*(nnodex-2)) %coefficient matrix
p1=zeros((nnodey)*(nnodex),1);j=1;
%x0=1/(nnodex-1);y0=(1/(2*(nnodey-1)))% since b=L/2
n11=2;k2=2;dt=(psi/(nnodey-1));i=1;k1=2;
n3=(nnodey-1)/3;i=1;t1=1;m=1;dto=2*dt%required step for ll
n33=n3;n32=n3+1;n1=nnodex-3;n22=nnodex-2;dtx=2*dt;
dr=(roe-roi)/(nnodex-1);rox=roi+2*dr;dty=2*dt;%or dx which is rout-rin
nite=(nnodex-4)*(nnodey-4);dtoo=n3*dt

for l=1:(nnodey-2)*(nnodex-2)-nite+(nnodey-4)
    if i==1
        ro=roi+dr;
        c(i,j)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dt/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dt/psi))))^3)/(dr^2))+(((1+((ko/(ro))*(1-
((dt+(dt/2))/psi))))^3)/(ro*dt^2))+(((1+((ko/(ro))*(1-((dt-
(dt/2))/psi))))^3)/(ro*dt^2)))
        c(i,j+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-(dt/psi))))^3)/(dr^2));
        c(i,j+(nnodex-2))=(((1+((ko/(ro))*(1-
((dt+(dt/2))/psi))))^3)/(ro*dt^2));
```

```
        i=i+1;
    elseif i>1 && i<m1*(nnodex-2)
        t=i;ro=roi+i*dr;
        c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dt/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dt/psi))))^3)/(dr^2))+(((1+((ko/(ro))*(1-
((dt+(dt/2))/psi))))^3)/(ro*dt^2))+(((1+((ko/(ro))*(1-((dt-
(dt/2))/psi))))^3)/(ro*dt^2)))
        c(i,t+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-(dt/psi))))^3)/(dr^2));
        c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-(dt/psi))))^3)/(dr^2));
        c(i,t+(nnodex-2))=(((1+((ko/(ro))*(1-
((dt+(dt/2))/psi))))^3)/(ro*dt^2));i=i+1;
    elseif i==(nnodex-2)
        t=i;ro=roi+i*dr;
        c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dt/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dt/psi))))^3)/(dr^2))+(((1+((ko/(ro))*(1-
((dt+(dt/2))/psi))))^3)/(ro*dt^2))+(((1+((ko/(ro))*(1-((dt-
(dt/2))/psi))))^3)/(ro*dt^2)))
        c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-(dt/psi))))^3)/(dr^2));
        c(i,t+(nnodex-2))=(((1+((ko/(ro))*(1-
((dt+(dt/2))/psi))))^3)/(ro*dt^2));i=i+1;
    elseif i==t1+m*(nnodex-2) && i<(n3-1)*(nnodex-2)+1
        t=i;ro=roi+dr;
        c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dto/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dto/psi))))^3)/(dr^2))+(((1+((ko/(ro))*(1-
((dto+(dt/2))/psi))))^3)/(ro*dt^2))+(((1+((ko/(ro))*(1-((dto-
(dt/2))/psi))))^3)/(ro*dt^2)))
        c(i,t+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-(dto/psi))))^3)/(dr^2));
        c(i,t+(nnodex-2))=(((1+((ko/(ro))*(1-
((dto+(dt/2))/psi))))^3)/(ro*dt^2));
        c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dto-
(dt/2))/psi))))^3)/(ro*dt^2));
        dto=dto+dt;m=m+1;i=i+1;
    elseif i==(n3-1)*(nnodex-2)+1 %and i>1 if starting node has angle psi/3
        t=i;ro=roi+dr;%dtoo=dto; these two can or cant be equal
        c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(1/(ro^4*dt^2))+(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2)));
        c(i,t+(nnodex-2))=(1/(ro^4*dt^2)) %here we take the same height
function as before along the line where angle=psi/3
        c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2));
        c(i,t+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2));i=i+1;
   % elseif i>(n3-1)*(nnodex-2)+1 && i<(n3)*(nnodex-2) % not needed step
  %       t=i;ro=roi+2*dr;
   %      c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi)))))/(dr^2))+(1/(ro^4*dt^2))+(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi)))))/(ro*dt^2)));
   %     c(i,t+(nnodex-2))=(1/(ro^4*dt^2))
   %     c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2));
  %      c(i,t+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2));
  %      c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi))))^3)/(dr^2));%here we take the same height function as before
along the line where angle=psi/3
```

```
  %       ro=ro+dr;
    elseif i==k1*(nnodex-2) && i<(n3)*(nnodex-2)
        t=i;ro=roi+(nnodex-2)*dr;
        c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dty/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dty/psi))))^3)/(dr^2))+(((1+((ko/(ro))*(1-
((dty+(dt/2))/psi))))^3)/(ro*dt^2))+(((1+((ko/(ro))*(1-((dty-
(dt/2))/psi))))^3)/(ro*dt^2)))
        c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-(dty/psi))))^3)/(dr^2));
        c(i,t+(nnodex-2))=(((1+((ko/(ro))*(1-
((dty+(dt/2))/psi))))^3)/(ro*dt^2));
        c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dty-
(dt/2))/psi))))^3)/(ro*dt^2));
        k1=k1+1;dty=dty+dt;i=i+1;
    elseif i==(n3)*(nnodex-2)
        t=i;ro=roi+(nnodex-2)*dr; dtx=dtoo+dt %dtoo in this case and dtx
used in future
        c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(1/(ro^4*dt^2))+(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2)));
        c(i,t+(nnodex-2))=(1/(ro^4*dt^2));
        c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2));
        c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi))))^3)/(dr^2));i=i+1;
    elseif i==(n32)*(nnodex-2) && i<(nnodey-2)*(nnodex-2)
        ro=roi+(nnodex-2)*dr;t=i;
        c(i,t)=-((1/(((ro+dr/2)^2)*((dr)^2)))+(1/(((ro-
dr/2)^2)*((dr)^2)))+(1/(ro^4*dt^2))+(1/(ro^4*dt^2)))
        c(i,t+(nnodex-2))=(1/(ro^4*dt^2));
        c(i,t-(nnodex-2))=(1/(ro^4*dt^2));
        c(i,t-1)=(1/(((ro-dr/2)^2)*((dr)^2)));n32=n32+1;i=i+1;
    elseif i==1+(n33)*(nnodex-2) && i<(nnodey-2)*(nnodex-2)-n1
        %from here on we deal with completely new height function till end
        t=i;ro=roi+dr; %here theta should be dtoo+dt
        c(i,t)=-((1/(((ro+dr/2)^2)*((dr)^2)))+(1/(((ro-
dr/2)^2)*((dr)^2)))+(1/(ro^4*dt^2))+(1/(ro^4*dt^2)))
        c(i,t+1)=(1/(((ro+dr/2)^2)*((dr)^2)));
        c(i,t+(nnodex-2))=(1/(ro^4*dt^2));
        c(i,t-(nnodex-2))=(1/(ro^4*dt^2)); i=i+1;n33=n33+1%are included
    elseif i==(nnodey-2)*(nnodex-2)-n1
        t=i;ro=roi+dr;
        c(i,t)=-((1/(((ro+dr/2)^2)*((dr)^2)))+(1/(((ro-
dr/2)^2)*((dr)^2)))+(1/(ro^4*dt^2))+(1/(ro^4*dt^2)))
        c(i,t+1)=(1/(((ro+dr/2)^2)*((dr)^2)));
        c(i,t-(nnodex-2))=(1/(ro^4*dt^2));
        i=i+1;
    elseif i>=(nnodey-2)*(nnodex-2)-n && i<(nnodey-2)*(nnodex-2)
        t=i;
        c(i,t)=-((1/(((rox+dr/2)^2)*((dr)^2)))+(1/(((rox-
dr/2)^2)*((dr)^2)))+(1/(rox^4*dt^2))+(1/(rox^4*dt^2)));
        c(i,t+1)=(1/(((rox+dr/2)^2)*((dr)^2)));
        c(i,t-1)=(1/(((rox-dr/2)^2)*((dr)^2)));
        c(i,t-(nnodex-2))=(1/((rox^4)*dt^2));i=i+1;rox=rox+dr;
    elseif i==(nnodey-2)*(nnodex-2)
        t=i;ro=roi+(nnodex-2)*dr;
        c(i,t)=-((1/(((ro+dr/2)^2)*((dr)^2)))+(1/(((ro-
dr/2)^2)*((dr)^2)))+(1/(ro^4*dt^2))+(1/(ro^4*dt^2)))
        c(i,t-(nnodex-2))=(1/(ro^4*dt^2));
        c(i,t-1)=(1/(((ro-dr/2)^2)*((dr)^2)));i=i+1;
```

```
    else
        for f=1:(nnodex-4)
            ro=roi+(i-n22)*dr;
            if dtx<dtoo
                t=i;
               c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtx/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtx/psi))))^3)/(dr^2))+(((1+((ko/(ro))*(1-
((dtx+(dt/2))/psi))))^3)/(ro*dt^2))+(((1+((ko/(ro))*(1-((dtx-
(dt/2))/psi))))^3)/(ro*dt^2)))
               c(i,t+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtx/psi))))^3)/(dr^2));
               c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtx/psi))))^3)/(dr^2));
               c(i,t+(nnodex-2))=(((1+((ko/(ro))*(1-
((dtx+(dt/2))/psi))))^3)/(ro*dt^2));
               c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dtx-
(dt/2))/psi))))^3)/(ro*dt^2));
               i=i+1;
            elseif dtx==dtoo
                t=i;
               c(i,t)=-((((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi))))^3)/(dr^2))+(1/(ro^4*dt^2))+(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2)));
               c(i,t+(nnodex-2))=(1/(ro^4*dt^2))
               c(i,t-(nnodex-2))=(((1+((ko/(ro))*(1-((dtoo-
(dt/2))/psi))))^3)/(ro*dt^2));
               c(i,t+1)=(((ro+dr/2)*(1+((ko/(ro+dr/2))*(1-
(dtoo/psi))))^3)/(dr^2));
               c(i,t-1)=(((ro-dr/2)*(1+((ko/(ro-dr/2))*(1-
(dtoo/psi))))^3)/(dr^2));
               i=i+1;
            else  % this case for dtx>dtoo
                t=i;
               c(i,t)=-((1/(((ro+dr/2)^2)*((dr)^2)))+(1/(((ro-
dr/2)^2)*((dr)^2)))+(1/(ro^4*dt^2))+(1/(ro^4*dt^2)))
               c(i,t+1)=(1/(((ro+dr/2)^2)*((dr)^2)));
               c(i,t-1)=(1/(((ro-dr/2)^2)*((dr)^2)));
               c(i,t+(nnodex-2))=(1/(ro^4*dt^2));
               c(i,t-(nnodex-2))=(1/(ro^4*dt^2));
               i=i+1;
            end
        end
        n22=n22+(nnodex-2);dtx=dtx+dt;
    end
end
r1(1:(n3)*(nnodex-2),1)=-6*(ko/psi);
%r1((n3-1)*(nnodex-2):(n3)*(nnodex-2),1)=10^10;
r1((n3)*(nnodex-2)+1,1)=0;
p=inv(c)*r1;

p1((nnodex)+2:2*nnodex-1,1)=p(1:(nnodex)-2,1);
k3=(nnodex-1);k4=2*((nnodex)-2);
for r=2:nnodey-2
p1((r*nnodex)+2:(r+1)*nnodex-1,1)=p(k3:k4,1);
k3=k3+(nnodex-2);k4=k4+(nnodex-2);
end
%p1=abs(p1)

g1=1;h1=1;
```

```
for l=1:nnodey
for g=1:nnodex
    pp(l,g)=p1(g1,1);
    g1=g1+1;
end
%l=l+1;
end
%A=25120557.34
%loadcap=sum(p*A,1)
%px=p1*(roe-roi)*psi %this step is equivalent to double integration
 [rxx,tt]=meshgrid(roi:dr:roe,0:dt:psi)
[b,c]=pol2cart(tt,rxx)
loadcap1=trapz(tt(:,1),trapz(rxx(1,:),pp.*rxx,2))%sum(px); %non dimensional
load capacity
pdim=p1*925619.834 %taking pressure with dimensions
lc=loadcap1*((0.01*28*0.1^(4))/((55^2*10^(-12)))) % dimensional load
capacity


%%
%power loss calculations
rot=roi+dr;dtot=dt;vl1=1;vl2=2;vlx=1;
for u=1:nnodey
    for v=1:nnodex
        if dtot<psi/3 % depicting the height variation using conditions
          hot=((1+((ko/(rot)))*(1-((dtot)/psi))))%(((1+((ko/(rot))*(1-
((dtot)/psi)))))%/(rot*dt^2))
        %given conditions for height hot such that you have various heights
        tsr(vlx,1)=(((rot^3/(3*hot))+((rot^2)*hot/(2*2)).*((p1(vl2)-
p1(vl1))/(dt))))
        vlx=vlx+1;
        elseif dtot==psi/3
          hot=((1+((ko/(rot)))*(1-((dtot)/psi))))%(((1+((ko/(rot))*(1-
((dtot)/psi)))))%/(rot*dt^2))
        %given conditions for height hot such that you have various heights
        tsr(vlx,1)=(((rot^3/(3*hot))+((rot^2)*hot/(2*2)).*((p1(vl2)-
p1(vl1))/(dt))))% here we have integration over theta for the first term
only
            vlx=vlx+1;
        elseif u==nnodey && v==nnodey %in loop sense since we
             hot=1/rot;
            tsr(vlx,1)=(((rot^3/(3*hot))+((rot^2)*hot/(2*2)).*(0/(dt))))
            vlx=vlx+1;
        else
            hot=1/rot;
           tsr(vlx,1)=(((rot^3/(3*hot))+((rot^2)*hot/(2*2)).*((p1(vl2)-
p1(vl1))/(dt))))
            vlx=vlx+1;
        end
        rot=rot+dr;vl1=vl1+1;vl2=vl2+1;
    end
    rot=roi+dr;dtot=dtot+dt;
end
%Po=(tsr.*(roe-roi).*psi)*dr; %Po=To from textbook %*((55*10^(-
6))/(0.01*(0.1^(4))*(28^2)))
%P=P.*(roe-roi).*psi
tsr(end+1,1)=0
a1=1;b1=1;b22=1;%b22=2*nnodex;
 for a=1:nnodey%2*ceil(nelementy/2)-1
    for b=1:nnodex
        tsx(a1,b1)=tsr(b22,1);
```

```
        b1=b1+1;b22=b22+1;
%         p1(a1+1,b1)=p(xr2,1);
        %xr1=xr1+2;xr2=xr2+2;b1=b1+1;
     end
     a1=a1+1;b1=1;%xr1=(b22)-1;xr2=(b22); %since end and start are the same
in system matrix
%      b22=b22+(2*nelementx)
 end
Po=trapz(tt(:,1),trapz(rxx(1,:),tsx,2));
Pof=sum(Po);fo=(Pof/loadcap1); % we get for each pad and same for whole
bearing
```