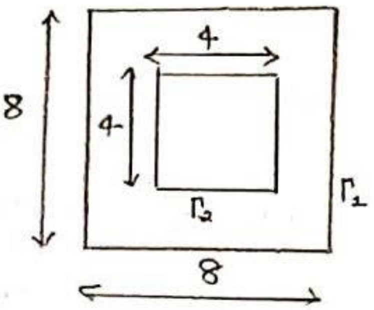
AKSHAY J
21105012

FINITE ELEMENT METHOD

ASSIGNMENT - 3.



1> find $u: \Omega \to \mathbb{R}$ such that

$$\left.\begin{aligned} q_{i,i} &= f && \text{on } \Omega \\ u &= 1 && \text{on } \Gamma_1 \\ u &= 0 && \text{on } \Gamma_2 \end{aligned}\right\} \quad (S)$$

$q \to$ Heat flux

$u \to$ Temperature.

2> Solution space : $\mathcal{S} = \{ u \mid u \in H^1,\ u = 1 \text{ on } \Gamma_1$

weight function space : $\mathcal{V} = \{ \omega \mid \omega \in H^1,\ \omega = 0 \text{ on } \Gamma_1$

$$\left.\begin{aligned} &\text{find } u \in \mathcal{S} \text{, such that for all } \omega \in \mathcal{V} : \\ &-\int_\Omega \omega_{,i}\, q_i \, d\Omega = 0 \end{aligned}\right\} \quad (W)$$

$$a(\omega, u) = \int_\Omega \omega_{,i}\, K_{ij}\, u_{,j}\, d\Omega$$

$$= K \int_\Omega \omega_{,i}\, u_{,j}\, d\Omega.$$

K_{ij} = constant = Thermal conductivity

$$\left.\begin{aligned} &\text{find } u \in \mathcal{S} \text{, such that for all } \omega \in \mathcal{V} : \\ &-a(\omega, u) = 0. \end{aligned}\right\} \quad (W)$$

3> Finite dimensional space ,

$S^h \subset \mathcal{S}$, $\mathcal{V}^h \subset \mathcal{V}$

$$\left.\begin{aligned} &\text{find } u^h = v^h + g^h \in \mathcal{S}^h \text{ such that for all } \omega^h \in \mathcal{V}^h \\ &a(\omega^h, v^h) = -a(\omega^h, g^h) \quad - \text{ (a)} \end{aligned}\right\} (G)$$

$w^h = \sum_{A\in\eta-\eta_g} c_A N_A \qquad g^h = \sum_{B\in\eta_g} d_B N_B$

$\eta \rightarrow$ set of all nodal points.

$\eta_g \rightarrow$ Nodes which are g type.

$N_A, N_B \rightarrow$ Bilinear shape functions.

Substituting in (a):

$$a\Big(\sum_{A\in\eta-\eta_g} c_A N_A ,\ \sum_{B\in\eta-\eta_g} d_B N_B\Big) = -\,a\Big(\sum_{A\in\eta-\eta_g} c_A N_A ,\ \sum_{B\in\eta_g} g_B N_B\Big)$$

$$\sum_A c_A\Big[\sum_B a(N_A, N_B)\, d_B + \sum_B a(N_A, N_B)\, g_B\Big] = 0.$$

$$\Rightarrow \sum_B a(N_A, N_B)\, d_B = -\sum_B a(N_A, N_B)\, g_B .$$

$$\underset{\sim}{K}\,\underset{\sim}{d} = \underset{\sim}{F}$$

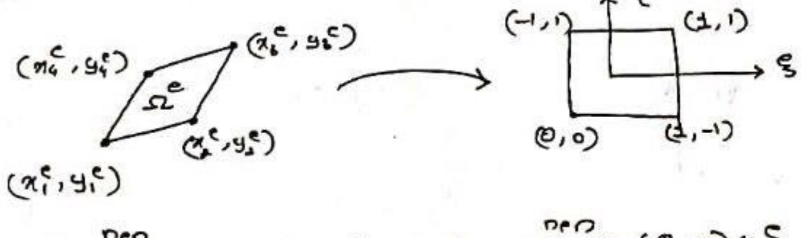
where $\underset{\sim}{K} = a(N_A, N_B) = \int_\Omega (\nabla N_A)^T (\nabla N_B)\, d\Omega .$

$\underset{\sim}{F} = -\sum_B a(N_A, N_B)\, g_B = \int_\Omega (\nabla N_A)^T (\nabla N_B)\, g_B\, d\Omega .$

$\underset{\sim}{d} = K^{-1} F$

$u^h = \sum_{B\in\eta} d_B N_B .$

consider a quadrilateral elements.



$nen = 4.$

$$x = \sum_{a=1}^{nen} N_a(\xi,\eta)\, x_a^e \qquad y = \sum_{a=1}^{nen} N_a(\xi,\eta)\, y_a^e$$

$$N_a(\xi,\eta) = \tfrac{1}{4}(1+\xi_a \xi)(1+\eta_a \eta)$$

$$\underset{\sim}{K}^e = a(N_a^e, N_b^e) = \int_\Omega \langle N_{a,x}\ \ N_{a,y}\rangle^T \langle N_{b,x}\ \ N_{b,y}\rangle\, d\Omega .$$

$$\langle N_{a,x}\ \ N_{a,y}\rangle = \langle N_{a,\xi}\ \ N_{a,\eta}\rangle \begin{bmatrix} \xi_{,x} & \xi_{,y} \\ \eta_{,x} & \eta_{,y} \end{bmatrix}$$

$$\langle N_{a,x}\ \ N_{a,y}\rangle = \langle N_{a,\xi}\ \ N_{a,\eta}\rangle \frac{1}{j} \begin{bmatrix} y_{,\eta} & -x_{,\eta} \\ -y_{,\xi} & x_{,\xi} \end{bmatrix}$$

where $j = x_{,\xi}\, y_{,\eta} - x_{,\eta}\, y_{,\xi}$

$$\underset{\sim}{k}^e = \int\!\!\int \left[\langle N_{a,x} \;\; N_{a,y}\rangle \begin{bmatrix} \xi_{,x} & \xi_{,y} \\ \eta_{,x} & \eta_{,y} \end{bmatrix}\right]^T \langle N_{b,x} \;\; N_{b,y}\rangle \begin{bmatrix} \xi_{,x} & \xi_{,y} \\ \eta_{,x} & \eta_{,y} \end{bmatrix} d\Omega.$$

$$\underset{\sim}{k}^e = \int_{-1}^{1}\!\!\int_{-1}^{1} \left[\frac{\langle N_{a,\xi} \;\; N_{a,\eta}\rangle}{j} \begin{bmatrix} y_{,\eta} & -x_{,\eta} \\ -y_{,\xi} & x_{,\xi} \end{bmatrix}\right]^T \frac{\langle N_{b,\xi} \;\; N_{b,\eta}\rangle}{j} \begin{bmatrix} y_{,\eta} & -x_{,\eta} \\ -y_{,\xi} & x_{,\xi} \end{bmatrix} d\xi\, d\eta$$

$$\underset{\sim}{f}^e = -a(N_a, N_b)\, g_b^e$$

$$\underset{\sim}{f}^e = -\int_{-1}^{1}\!\!\int_{-1}^{1} \left[\frac{\langle N_{a,\xi} \;\; N_{a,\eta}\rangle}{j} \begin{bmatrix} y_{,\eta} & -x_{,\eta} \\ -y_{,\xi} & x_{,\xi} \end{bmatrix}\right]^T \frac{\langle N_{b,\xi} \;\; N_{b,\eta}\rangle}{j} \begin{bmatrix} y_{,\eta} & -x_{,\eta} \\ -y_{,\xi} & x_{,\xi} \end{bmatrix} g_b^e\, d\xi\, d\eta$$

- **CODE FOR THE PROBLEM:-**

```
####AKSHAY J
####21105012
####FEM ASSGMT 3
import numpy as np
from matplotlib import pyplot as plt
import sympy as sp
from sympy import *

######Defining the mesh
nnp=64
n=int(nnp**0.5)
nel=(n-1)**2
n1=int(nel**0.5)
nen=4
x_dim=8   ###length of plate
y_dim=8   ###breadth of plate
h=1/(n-1)
hx=x_dim/(n-1)
hy=y_dim/(n-1)
```

```
x=np.zeros(nnp)
y=np.zeros(nnp)
c=0
for i in range(0,nnp):
    if c==n:
        c=0
    x[i]=c*hx
    c=c+1
c=0
d=1
for i in range(0,nnp):
    if i ==n*d:
        c=c+1
        d=d+1
    y[i]=c*hy

#####SETTING IEN,ID,LM
IEN=np.zeros([nen,nel])
A=np.linspace(1,nnp,nnp)
c=0
d=0
i=1
for e in range(0,nel):
    if e!=0:
        c=c-1
    if e==n1*i:
        i=i+1
        c=c+1
    for a in range(0,nen-2):
        IEN[a,e]=A[c]
        c=c+1
c=1+n
i=1
for e in range(0,nel):
```

```
    if e!=0:
        c=c+3
    if e==n1*i:
        i=i+1
        c=c+1
    for a in range(nen-2,nen):
        IEN[a,e]=A[c]
        c=c-1

####ID ARRAY
ID=np.ones([n,n],dtype=int)
ID[:,0]=0
ID[n-1,:]=0
ID[0,:]=0
ID[:,n-1]=0
ID[int(((n/2)-
1)):int((n/2+1)),int(((n/2)-
1)):int((n/2+1))]=0
ID=np.reshape(ID,nnp)
c=1
for i in range(0,nnp):
    if ID[i]!=0:
        ID[i]=c
        c=c+1
neq=c-1

####LM ARRAY
LM=np.zeros([nen,nel])
for e in range(0,nel):
    for a in range(0,nen):
        index=int(IEN[a,e]-1)
        index=ID[index]
        LM[a,e]=index
```

```
####CREATING THE GIVEN BC
gb=np.ones([n,n])
gb[round(0.4/h):round((0.6/h))+1,round(0.
4/h):round((0.6/h))+1]=0
gb=np.reshape(gb,nnp)
for i in range(0,nnp):
    if ID[i]!=0:
        gb[i]=0

######SHAPE FUNCTION SUBROUTINE
z, nt = sp.symbols('z nt')
def shape_fn(e):
   nta=[-1,-1,1,1]
   za=[-1,1,1,-1]
   A=np.zeros(nen)
   for a in range(0,nen):
       A[a]=int(IEN[a,e]-1)
   N1 = 0.25 * (1 + (z) * za[0]) * (1 +
(nt) * nta[0])
   N2 = 0.25 * (1 + (z) * za[1]) * (1 +
(nt) * nta[1])
   N3 = 0.25 * (1 + (z) * za[2]) * (1 +
(nt) * nta[2])
   N4 = 0.25 * (1 + (z) * za[3]) * (1 +
(nt) * nta[3])
   index1 = int(A[0])
   index2 = int(A[1])
   index3 = int(A[2])
   index4 = int(A[3])

xe=N1*x[index1]+N2*x[index2]+N3*x[index3]
+N4*x[index4]
   ye = N1 * y[index1] + N2 * y[index2] +
N3 * y[index3] + N4 * y[index4]
```

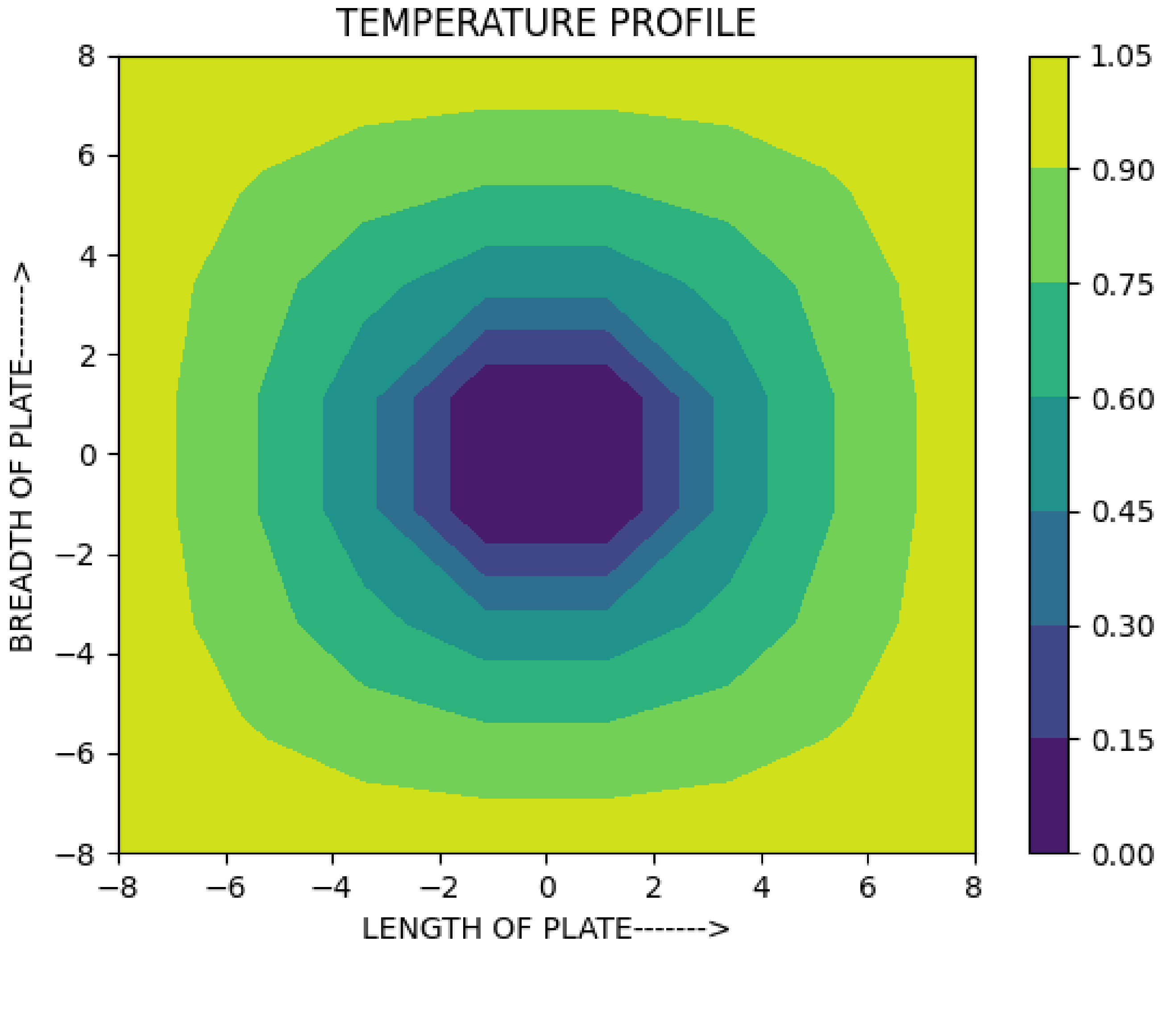
```
    ja=(diff(xe,z)*diff(ye,nt))-
(diff(xe,nt)*diff(ye,z))
    der = [[diff(ye, nt), diff(-xe, nt)],
[diff(-ye, z), diff(xe, z)]]
    n1_der = (1 / ja) *
np.matmul([N1.diff(z), N1.diff(nt)], der)
    n2_der = (1 / ja) *
np.matmul([N2.diff(z), N2.diff(nt)], der)
    n3_der = (1 / ja) *
np.matmul([N3.diff(z), N3.diff(nt)], der)
    n4_der = (1 / ja) *
np.matmul([N4.diff(z), N4.diff(nt)], der)
    N1_der = lambdify([z, nt], n1_der,
'numpy')
    N2_der = lambdify([z, nt], n2_der,
'numpy')
    N3_der = lambdify([z, nt], n3_der,
'numpy')
    N4_der = lambdify([z, nt], n4_der,
'numpy')

shape_sub=[N1_der,N2_der,N3_der,N4_der]
    return(shape_sub)


########OBTAINING ELEMENT LEVEL Ke AND Fe
and ASSEMBLING IT TO GLOBAL K AND F
ke=np.zeros([nen,nen])
fe=np.zeros([nen,1])
K=np.zeros([neq,neq])
F=np.zeros(neq)
o=1
for e in range(0,nel):
    for a in range(0,nen):
```

```
        for b in range(0,nen):
             N=shape_fn(e)

ke[a,b]=o*integrate(np.matmul(N[a](z,nt),
N[b](z,nt)),(nt,-1,1),(z,-1,1))
             P = int(LM[a, e])
             Q = int(LM[b, e])
             if P!=0 and Q!=0:
               K[P-1,Q-1]=K[P-1,Q-
1]+ke[a,b]
        fe[a] = -((ke[a, 0] *
gb[int(IEN[0, e] - 1)]) + (ke[a, 1] *
gb[int(IEN[1, e] - 1)]) + (
                         ke[a, 2] *
gb[int(IEN[2, e] - 1)]) + (ke[a, 3]*
gb[int(IEN[3,e]-1)]))
        if P!=0:
               F[P-1] = F[P-1] + fe[a]

######SOLVING FOR D AND GETTING THE
APPROXIMATE U
d=np.linalg.solve(K,F)
u=np.array(gb,dtype=float)
U=np.array(gb,dtype=float)
for e in range(0,nel):
    for a in range(0,nen):
      z, nt = sp.symbols('z nt')
      nta = [-1, -1, 1, 1]
      za = [-1, 1, 1, -1]
      N1, N2, N3, N4 = sp.symbols('N1 N2
N3 N4', cls=Function)
      N1 = 0.25 * (1 + (z) * za[0]) * (1
+ (nt) * nta[0])
      N2 = 0.25 * (1 + (z) * za[1]) * (1
```

```
+ (nt) * nta[1])
        N3 = 0.25 * (1 + (z) * za[2]) * (1
+ (nt) * nta[2])
        N4 = 0.25 * (1 + (z) * za[3]) * (1
+ (nt) * nta[3])
        P = int(LM[a, e])
        u=N1*d[P-1]+N2*d[P-1]+N3*d[P-
1]+N4*d[P-1]
        u=lambdify([z,nt],u,'numpy')
        if P!=0:
                U[int(IEN[a,e]- 1)] =
u(za[a], nta[a])
for e in range(0,nel):
    for a in range(0,nen):
        if int(LM[a,e])==0:
            U[int(LM[a,e]-1)]=1
U=np.reshape(U,(n,n))
U=np.asarray(U)
print('TEMPERATURE PROFILE= ','\n',U)

######PLOTTING THE FIGURE
lx=np.linspace(-x_dim,x_dim,n)
ly=np.linspace(-y_dim,y_dim,n)
[Lx,Ly]=np.meshgrid(lx,ly)
plt.contourf(Lx,Ly,U)
plt.title('TEMPERATURE PROFILE')
plt.xlabel('LENGTH OF PLATE------->')
plt.ylabel('BREADTH OF PLATE------->')
plt.colorbar()
plt.show()
```

- **OUTPUTS:-**

```
TEMPERATURE PROFILE=
 [[1.         1.         1.         1.         1.         1.
  1.         1.        ]
 [1.         0.91517143 0.83691349 0.78620167 0.78620167 0.83691349
  0.91517143 1.        ]
 [1.         0.83691349 0.64754444 0.50947688 0.50947688 0.64754444
  0.83691349 1.        ]
 [1.         0.78620167 0.50947688 0.         0.         0.50947688
  0.78620167 1.        ]
 [1.         0.78620167 0.50947688 0.         0.         0.50947688
  0.78620167 1.        ]
 [1.         0.83691349 0.64754444 0.50947688 0.50947688 0.64754444
  0.83691349 1.        ]
 [1.         0.91517143 0.83691349 0.78620167 0.78620167 0.83691349
  0.91517143 1.        ]
 [1.         1.         1.         1.         1.         1.
  1.         1.        ]]
```


TEMPERATURE PROFILE
BREADTH OF PLATE------->
LENGTH OF PLATE------->
−8
−6
−4
−2
0
2
4
6
8
1.05
0.90
0.75
0.60
0.45
0.30
0.15
0.00

9) For a total nodal point number of 64 Temperature profile for a rectangular plate with rectangular hole is obtained as given. It can be seen that temperature spread in the plate in concentric (uniform) manner due to the constant thermal conductivity and no heat source. When the number of nodal points are increased the profile variation becomes more and more circular in nature. Even though near the hole temperature variation depends on the geometry of hole, far away from it, the temperature variation is found to be circular (oval shaped if not a square plate).