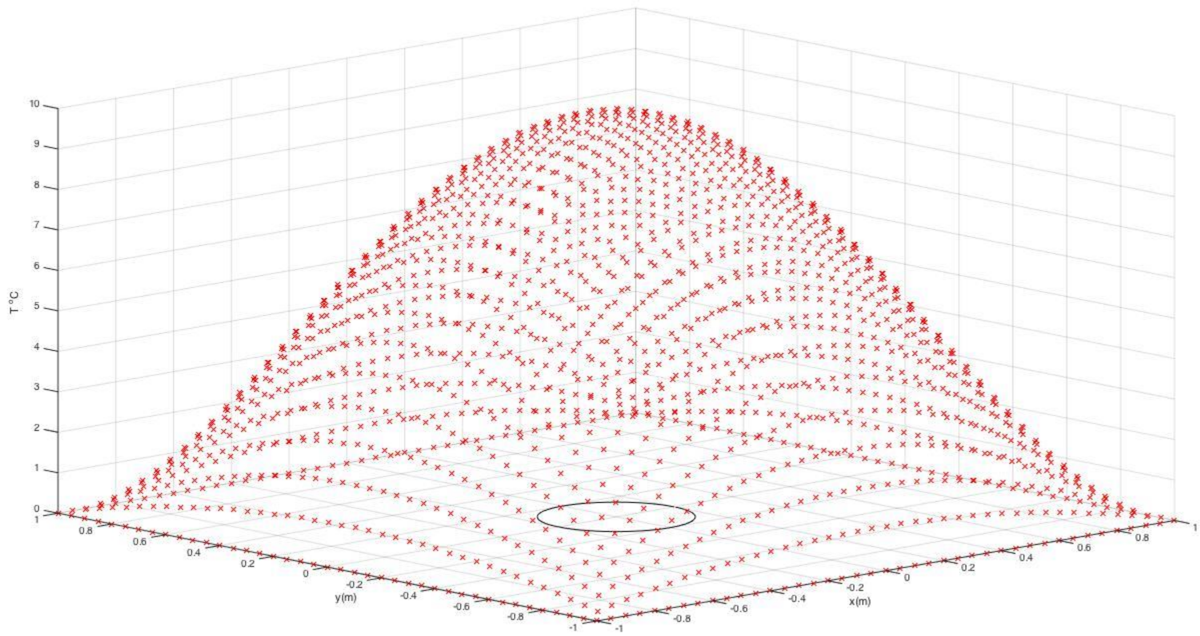
MAE 598

FINITE ELEMENTS IN ENGINEERING

PROJECT 2



KARTHIK RAJAN VENKATESAN

1207607553

GAURAV BORGAONKAR

1207884193

RISHIKA VENKA

1207757937

CONTENTS

PROBLEM STATEMENT

Analyze the following heat transfer problem in a thin plate with a center hole, shown in the Fig (1). The outer boundary of the plate is fixed at (T = 0 ºC), and a non-uniform heat flux is applied on the surface of the hole. A heat source term is applied over the entire body. The strong form for the heat equation is

$$\nabla \cdot q - s = 0 \tag{1}$$

Where the heat flux, q, is evaluated as

$$q = -k \nabla T \tag{2}$$

Using the method of manufactured solutions, we first choose an arbitrary solution for the temperature distribution,

$$T(x, y) = 10\left(x - \frac{L}{2}\right)\left(x + \frac{L}{2}\right)\left(y - \frac{L}{2}\right)\left(y + \frac{L}{2}\right)$$

Determine the heat source, by substituting (2) and (3) into (1) and solving for s. To compute the heat flux on Γq, evaluate (2) given (3), and integrate

$$\mathbf{f}_q^e = \int_{\Gamma_q} \mathbf{N}^e \bar{q}\, d\Gamma \quad \text{where } \bar{q} = \mathbf{q} \cdot \mathbf{n}$$

And n is the outward pointing normal direction.

Implement a finite element program in MATLAB to approximate the temperature distribution. Use exact solution to determine the rate of convergence.

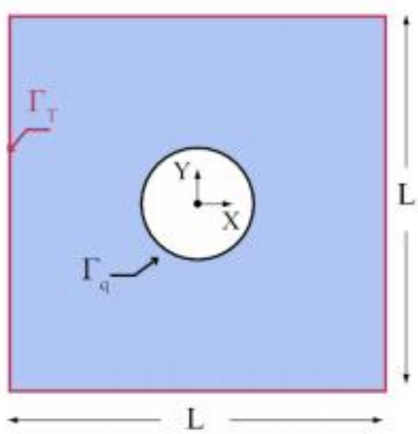


Fig.1. 2D plate with a center hole

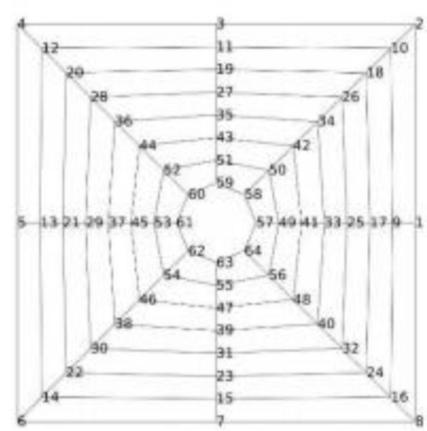


Fig. 2. Mesh with node numbering

Parameter		Value
Plate edge length	L	2 m
Hole radius	r	20 cm
Heat capacity	κ	54 W/(mK)

ANALYTICAL SOLUTION

1. ANALYTICAL SOLUTION FOR THE HEAT SOURCE FIELD

The strong form of the heat conduction equation in 1D is given by,

$$\frac{d}{dx}\left(Ak\frac{dT}{dx}\right) + s = 0, \qquad on\ \Omega$$

$$qn = -kn\frac{dT}{dx} = \bar{q} \qquad on\ \Gamma_q$$

$$T = \bar{T} \qquad on\ \Gamma_T$$

The strong form for the heat conduction is given by,

$$\nabla \cdot q - s = 0 \quad \dots\dots\dots\dots\dots (i)$$

Where the heat flux q is calculated as,

$$q = -\kappa\nabla T \quad \dots\dots\dots\dots\dots (ii)$$

In two dimensions, we have two flux components and two temperature gradient components. For isotropic materials in two dimensions, Fourier's law is given by,

$$\kappa\nabla^2 T + s = 0 \quad \dots\dots\dots\dots\dots (iii)$$

Where

$$\nabla^2 = \vec{\nabla} \cdot \vec{\nabla} = \nabla^T\nabla = \frac{d^2}{dx^2} + \frac{d^2}{dy^2}$$

The arbitrary solution for temperature is given by,

$$T(x,y) = 10\left(x - \frac{L}{2}\right)\left(x + \frac{L}{2}\right)\left(y - \frac{L}{2}\right)\left(y + \frac{L}{2}\right)$$

Differentiating with respect to x, we get,

$$\frac{\partial T}{\partial x} = 20x\left(y^2 - \frac{L^2}{4}\right)$$

$$\frac{d^2T}{dx^2} = 20\left(y^2 - \frac{L^2}{4}\right)$$

Differentiating with respect to y, we get,

$$\frac{\partial T}{\partial y} = 20y\left(x^2 - \frac{L^2}{4}\right)$$

$$\frac{d^2T}{dy^2} = 20\left(x^2 - \frac{L^2}{4}\right)$$

From equation (ii),

$$q = -\kappa\frac{\mathrm{dT}}{\mathrm{dx}} \dots\dots\dots (Equation\ for\ Heat\ Flux)$$

The flux vector $\bar{q}$ can be expressed in terms of two components: the component tangential to the boundary q_t and the component normal to the boundary q_n. The tangential component q_t does not contribute to the heat entering or exiting the control volume.

$$\vec{q} = q_x\vec{\imath} + q_y\vec{\jmath}$$

Where $$\vec{n} = n_x\vec{\imath} + n_y\vec{\jmath},$$

$$n_x{}^2 + n_y{}^2 = 1$$

The normal component q_n is given by the scalar product of the heat flux with the normal to the body:

$$q_n = \vec{q}.\vec{n} = q_x n_x + q_y n_y$$

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} k_{xx} & k_{xy} \\ k_{xy} & k_{yy} \end{bmatrix} \begin{bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{bmatrix}$$

$$q = - k \left\{20\, x^2 \left(y^2 - \frac{L^2}{4}\right) + 20\, y^2 \left(x^2 - \frac{L^2}{4}\right)\right\}/r$$

$$\boxed{\mathbf{q = 20\ k\{\ (x^2+y^2\,)\frac{L^2}{4} - 2x^2y^2\}/r}}$$

This is the equation for the heat flux distribution over the body.

2. ANALYTICAL SOLUTION FOR THE HEAT FIELD

From (i), we have,

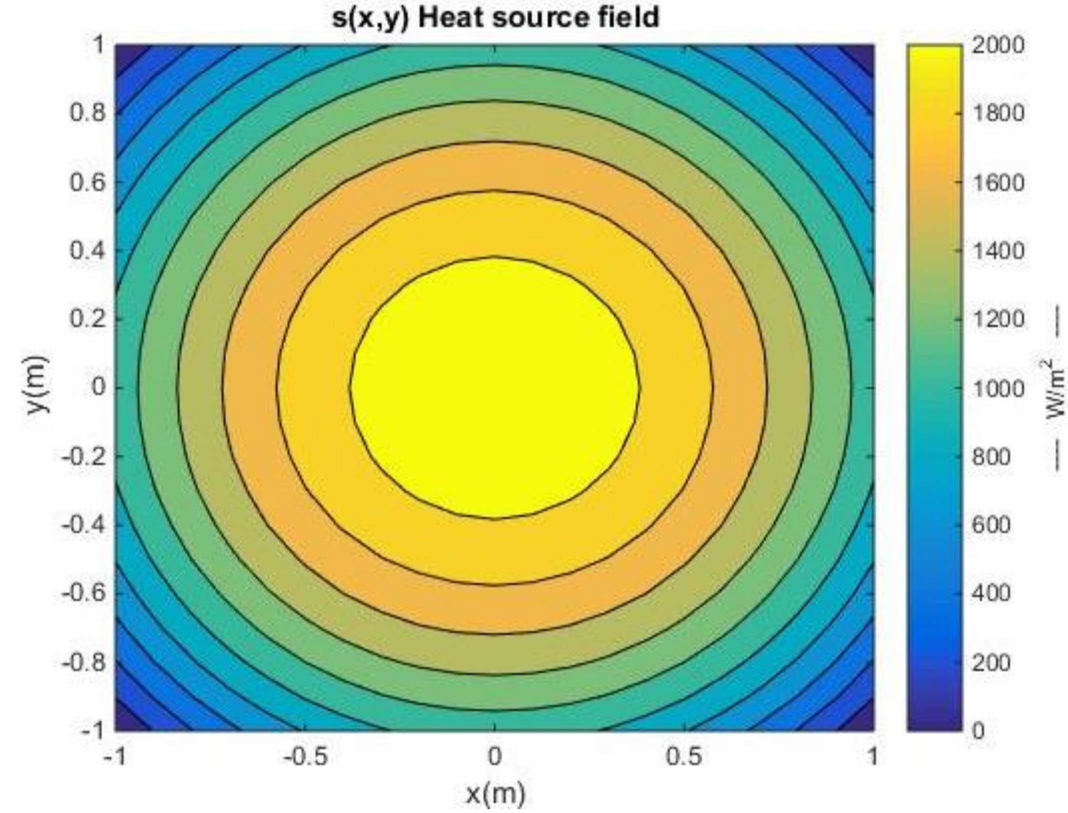
$$k\left(\frac{d^2T}{dx^2} + \frac{d^2T}{dy^2}\right) + s = 0$$

$$54 \times 20\left\{\left(x^2 - \frac{L^2}{4}\right) + \left(y^2 - \frac{L^2}{4}\right)\right\} + s = 0$$

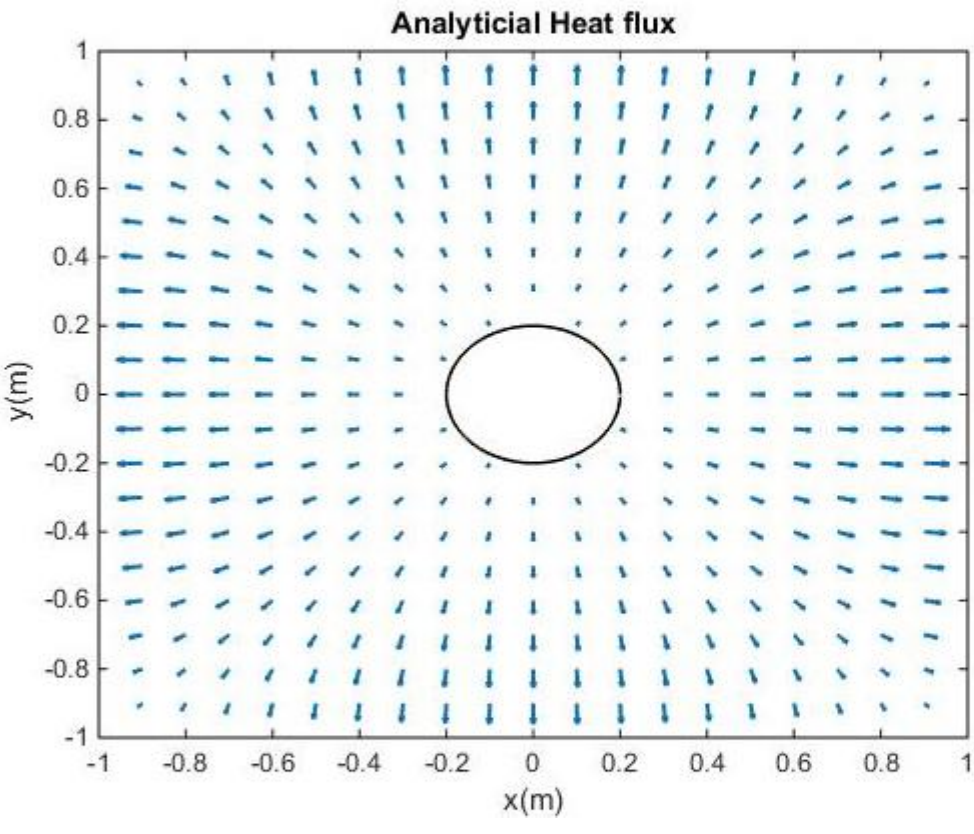
Hence, the source variation is given as,

$$\boxed{\boldsymbol{s = -1080\left(x^2 + y^2 - \frac{L^2}{2}\right)}}$$

3. CONTOUR PLOT OF ANALYTICAL HEAT SOURCE FIELD

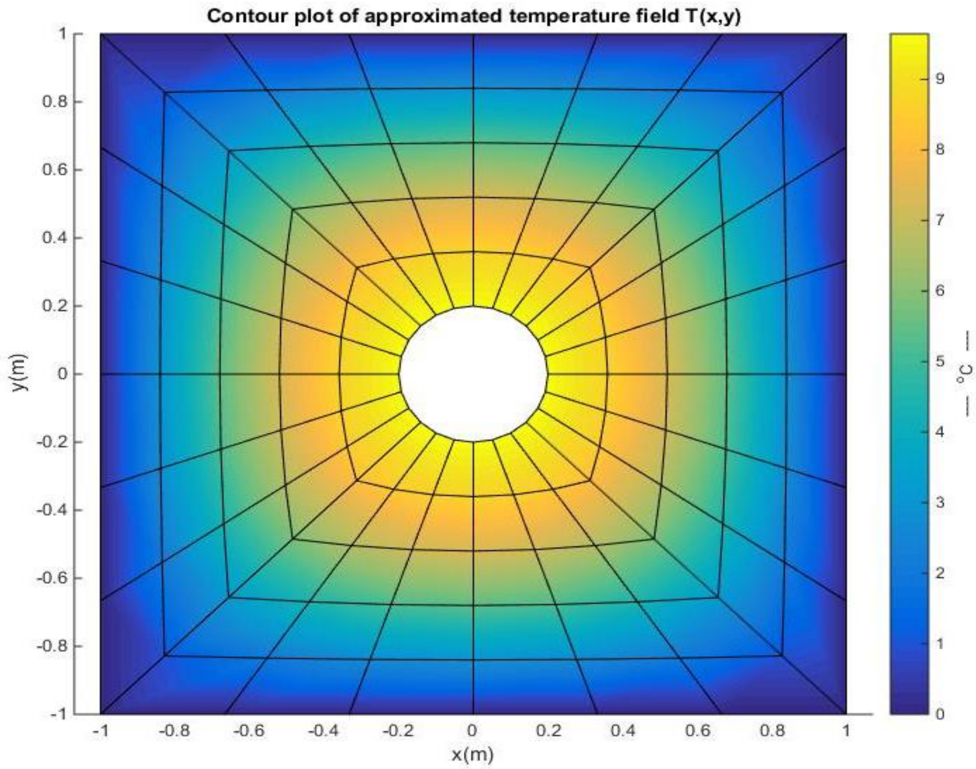


4. QUIVER PLOT OF THE ANALYTICAL HEAT FLUX FIELD

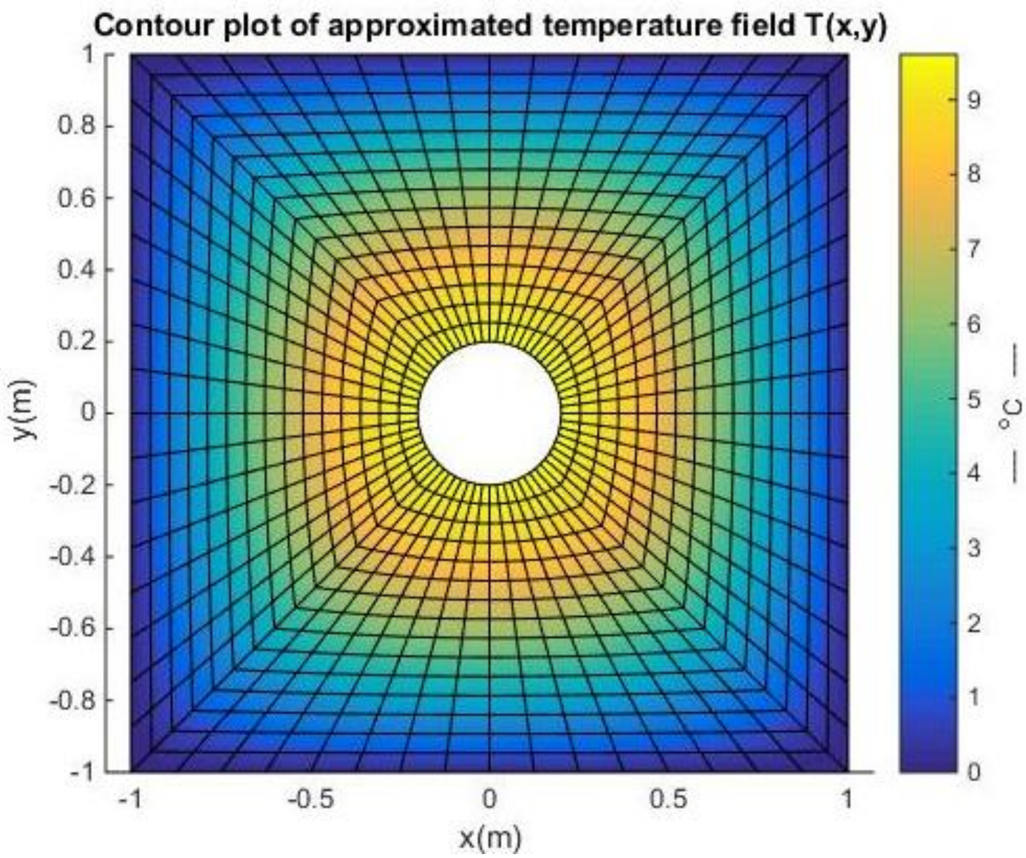


MATLAB SOLUTION

1. CONTOUR PLOT OF APPROXIMATED TEMPERATURE FIELDS FOR TWO MESH SIZES

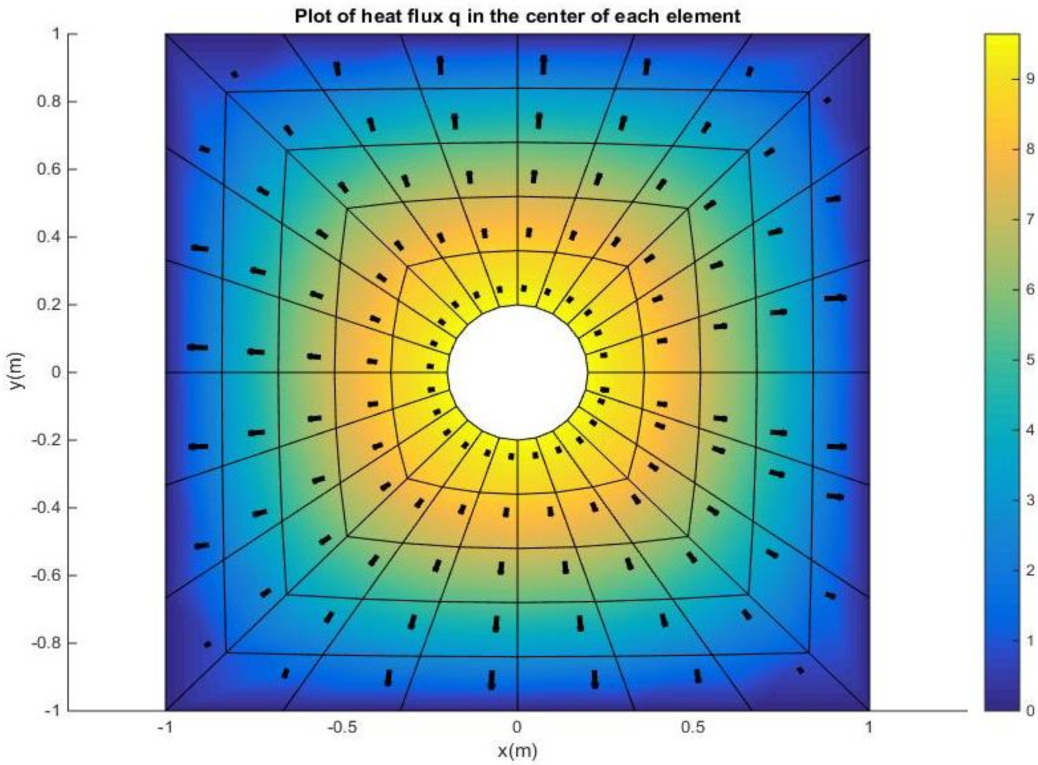


MESH SIZE: 3



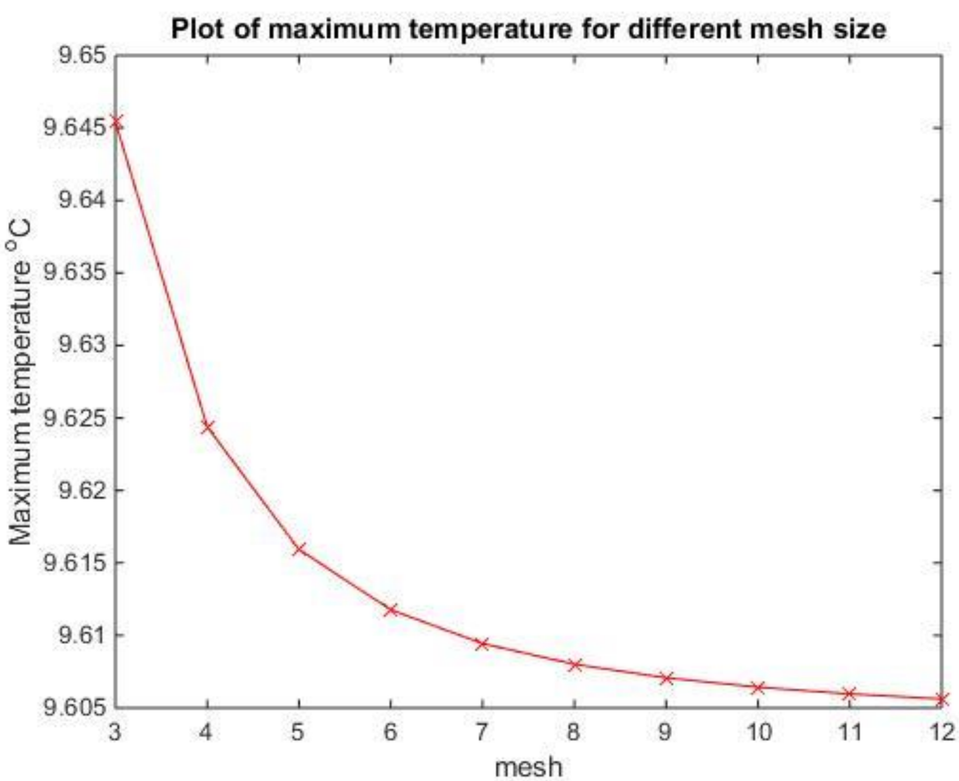
MESH SIZE: 8

2. PLOT OF HEAT FLUX AT THE CENTER OF EACH ELEMENT (using quiver command)

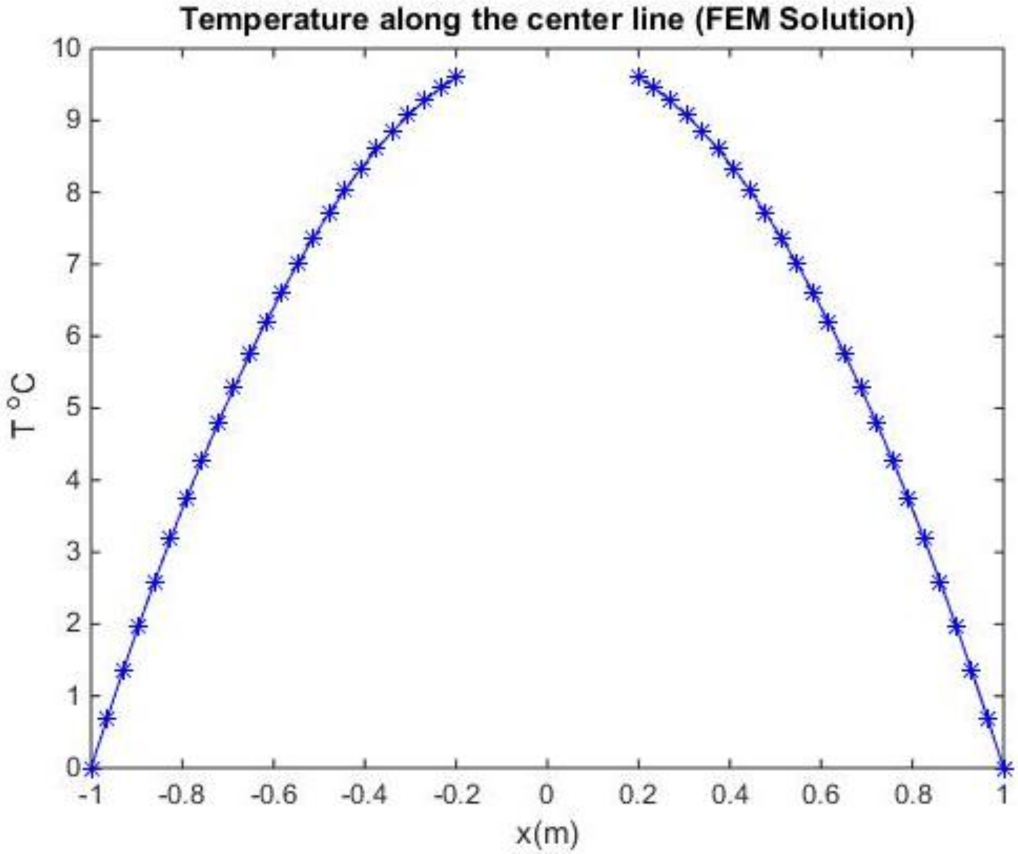


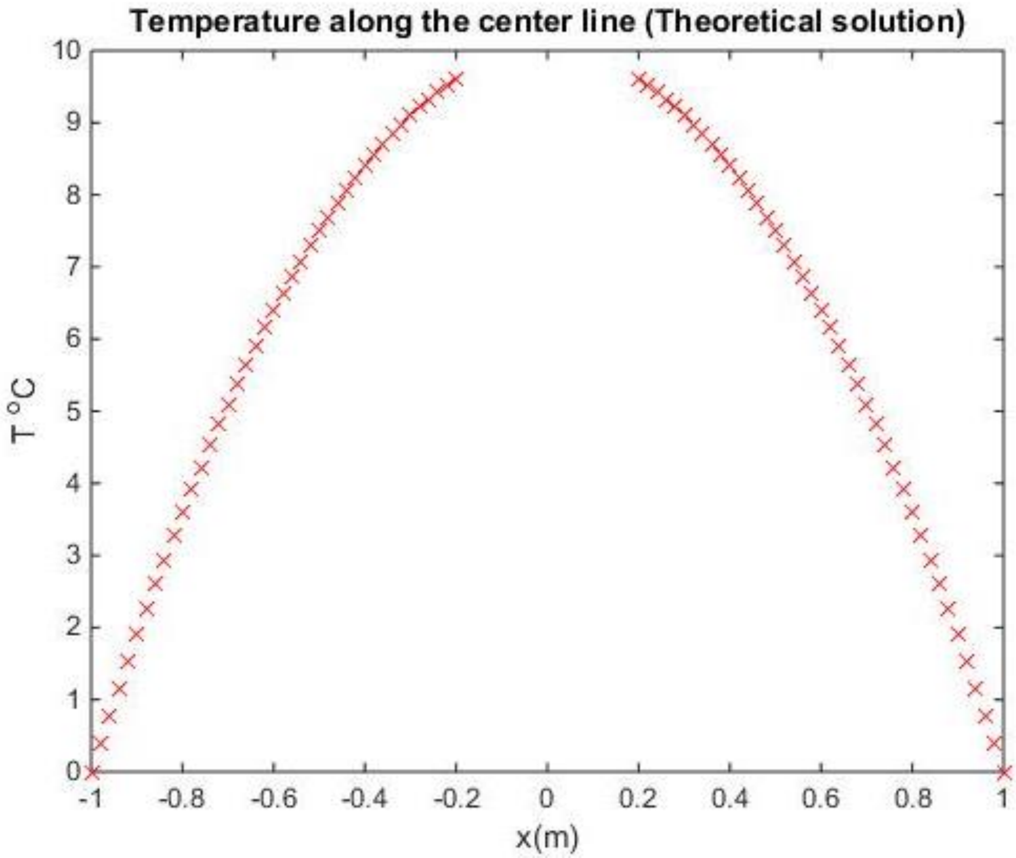
MESH SIZE: 3

3. ANALYZE THE MAXIMUM TEMPERATURE UNDER SEVERAL MESH SIZES

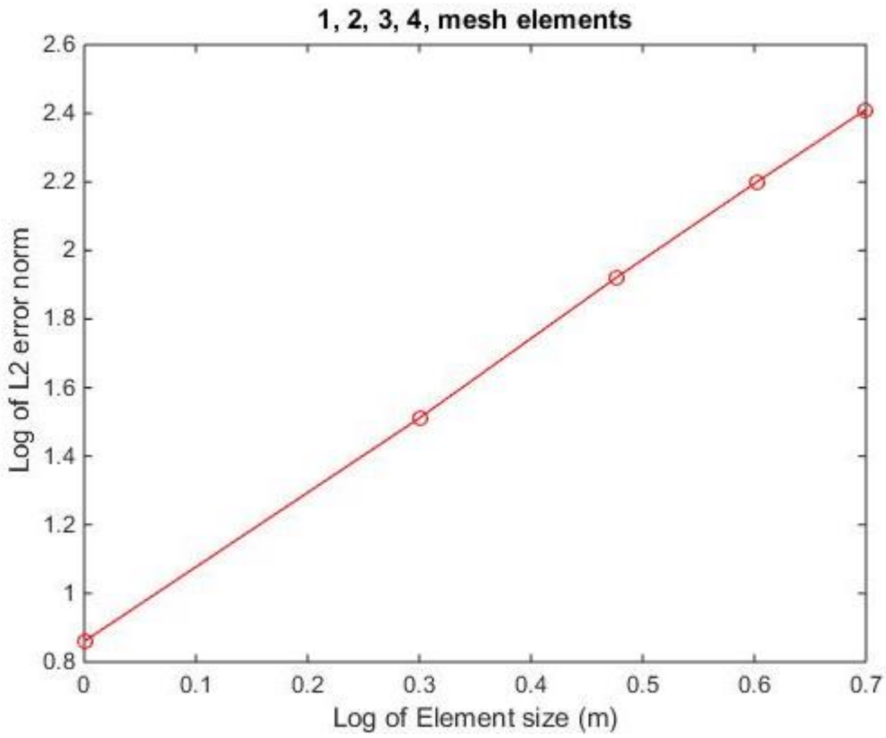


4. <u>PLOT TEMPERATURE ALONG A CENTER LINE (y=0, -L/2 < x < L/2) AND COMPARE THE RESULT WITH THEORETICAL SOUTION</u>





5. PLOT L2 TEMPERATURE ERROR NORM VS MESH SIZE ON A LOG-LOG PLOT AND COMPUTE THE RATE OF CONVERGENCE



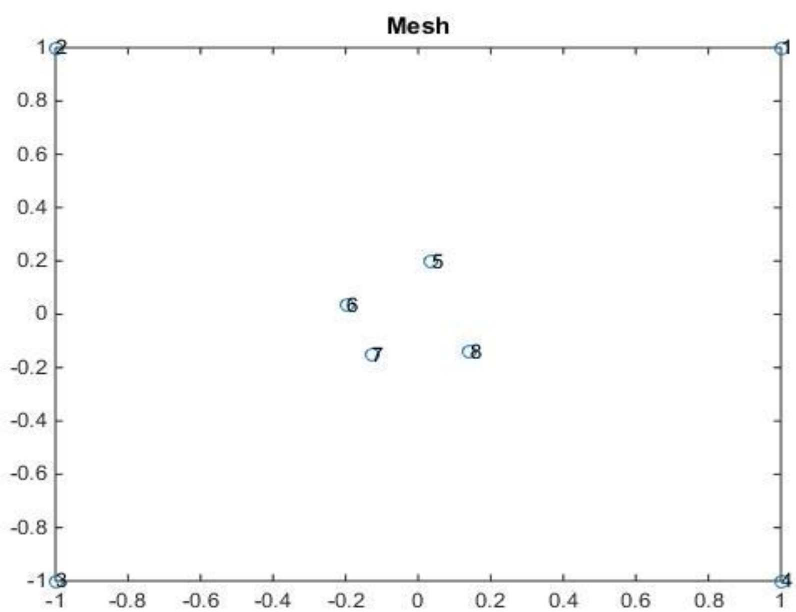
RATE OF CONVERGENCE ≅ 2

6. DEMONSTRATION OF CODE PASSING PATCH TEST

Generated mesh:

Total number of nodes: 8

Total number of elements: 4



The assumed temp solution is T = 2 +4x +5y

PATCH TEST PASS

%Nodal temperature values (FEM Solution)

```
d =
  11.000000000000002
   1.000000000000000
  -7.000000000000003
   3.000000000000001
   2.961499999999999
   1.153799999999999
   0.744199999999999
   2.141400000000000
```

%Nodal temperature values (Assumed analytical solution)

```
T =

  11.000000000000000
   1.000000000000000
  -7.000000000000000
   3.000000000000000
   2.961500000000000
   1.153800000000000
   0.744200000000000
   2.141400000000000
```

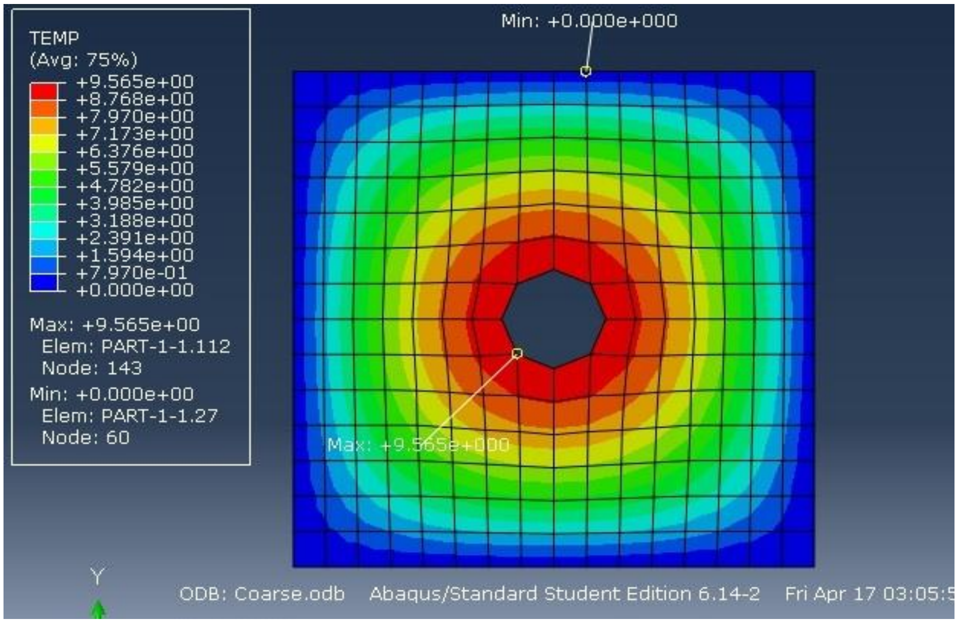
For the MATLAB code, see appendix.

ABAQUS SOLUTION

1. PLOT OF TEMPERATURE FIELDS FOR TWO DIFFERENT ELEMENT SIZES

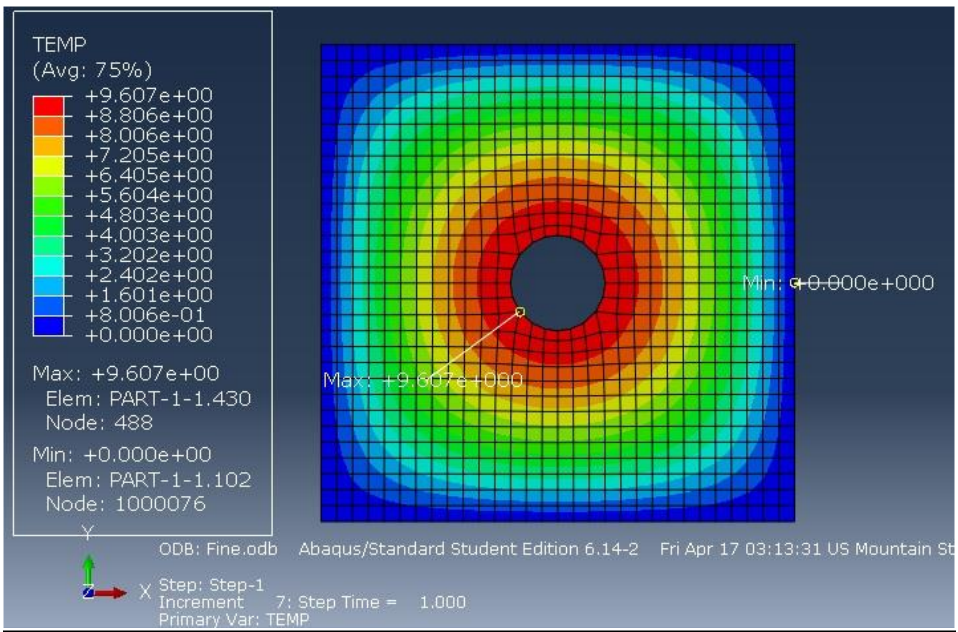
Plot of Temperature contours for coarser mesh:

For the coarser mesh conditions, the plate was meshed with global element size of 0.12m. The total number of elements generated was 220 with 254 nodes._The solution is found to converge to a fair degree of accuracy (with maximum temperature being shown as 9.565°C at the hole edges) with the analytical solution in this case.



Plot of Temperature contours for finer mesh:

For the finer mesh conditions, the plate was meshed with global element size of 0.06m. The total number of elements generated were 876 with 976 nodes.



The solution is found to converge perfectly with the analytical solution in this case.

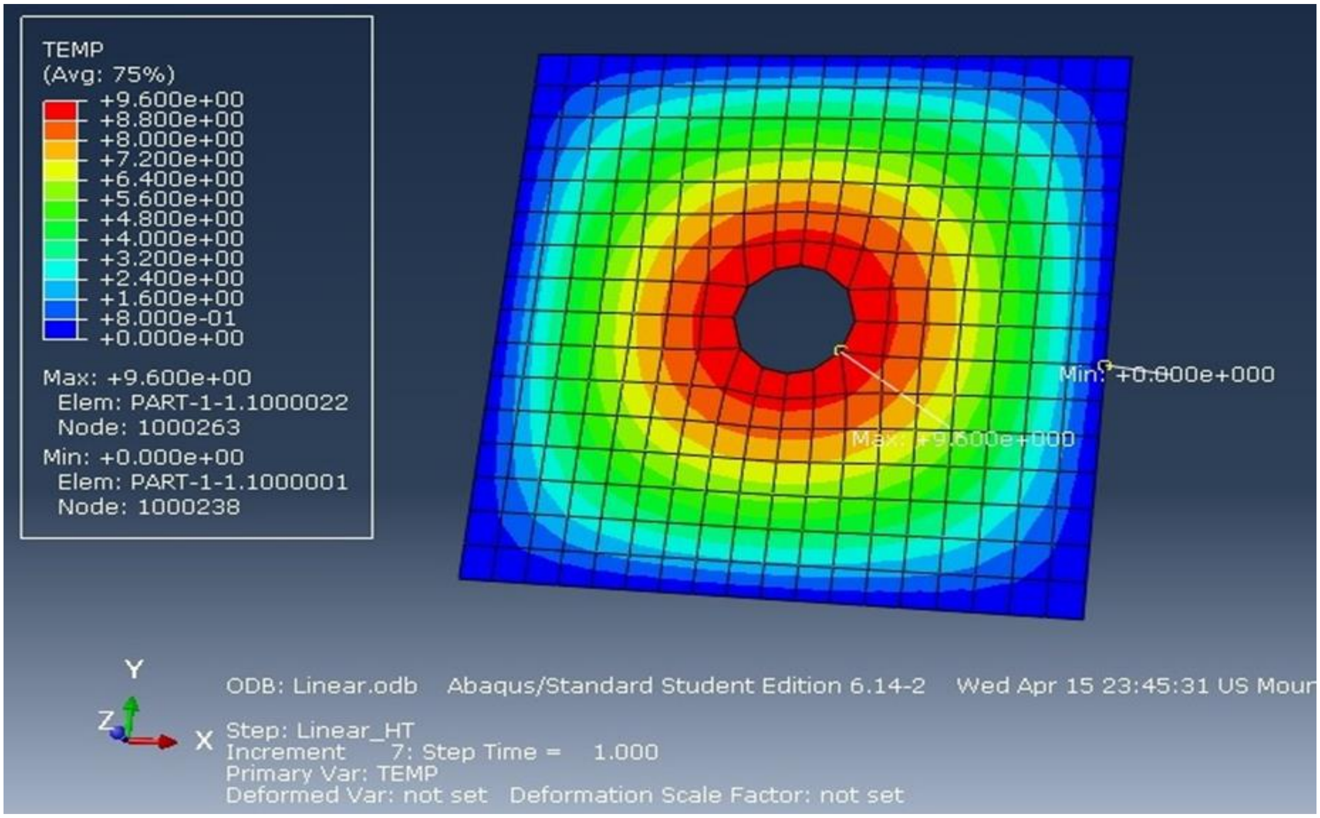
2. COMPARISON OF SOLUTIONS USING LINEAR AND QUADRATIC ELEMENTS

In order to compare solutions with different types of elements to check which elements yield better results for same number of elements, we need to mesh the model using the exactly the same mesh pattern with same no. of elements, however in the first case as 1st order elements (linear) while in the second case as 2nd order elements (quadratic). The following images show the results.

1st Order (Linear) Elements:

In the first case, the model was meshed with 280 elements and 380 nodes quadrilateral elements. A heat source term given in the analytical solution was applied over the entire body. The boundary conditions were heat flux term on the inner boundary of the plate and 0°C at the outer boundary.

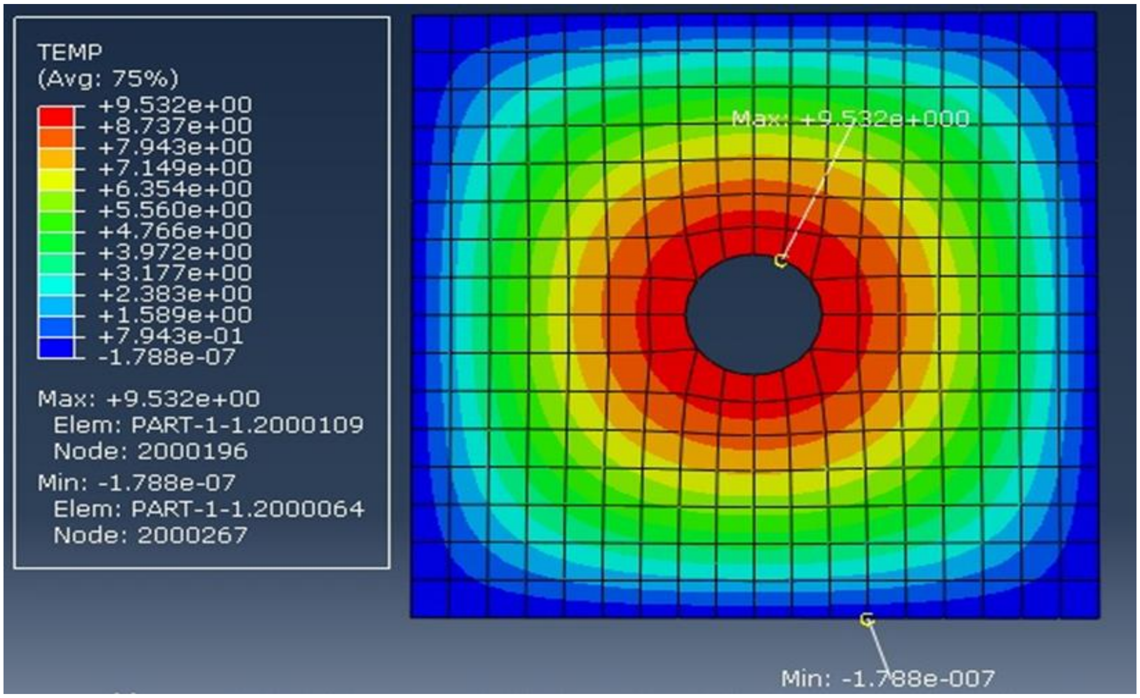
The following image shows the temperature contours obtained. As expected, the maximum temperature is found at the inner boundary and minimum temperature at the outer boundary. The linear elements give a perfect solution even for 280 elements. Hence, the linear elements give a very good solution even for a very basic no. of elements.



2nd order (Quadratic) Elements:

In the second case, the model was meshed maintaining the same mesh pattern with 280 no of elements and 920 no. of nodes. A heat source (see analytical solution) was applied over the entire body. The boundary conditions were heat flux on the inner boundary of the plate and 0°C at the outer boundary.

The following image shows the temperature contours obtained. As expected, the maximum temperature is found at the inner boundary and minimum temperature at the outer boundary. However, in this case, solution does not converge perfectly and the maximum temperature found at the inner boundary is only about 9.532°C. This leaves an error margin of 0.7% in the solution.



Conclusion:

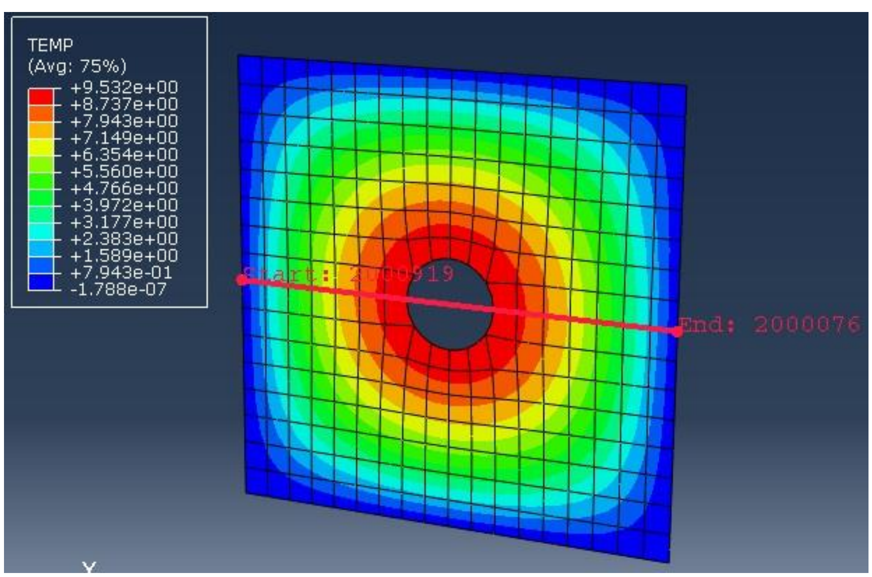
For same no. of elements, solutions using linear elements took about 0.30 seconds to complete while solution using quadratic elements took about 0.44 seconds to complete. Also, the linear element mesh performs better in case of convergence compared to the quadratic element.

This leads us to conclude that given the limitation of number of elements, linear elements should be preferred over quadratic because of better convergence rate and for reduced solution time (about 25% reduction in solution time).

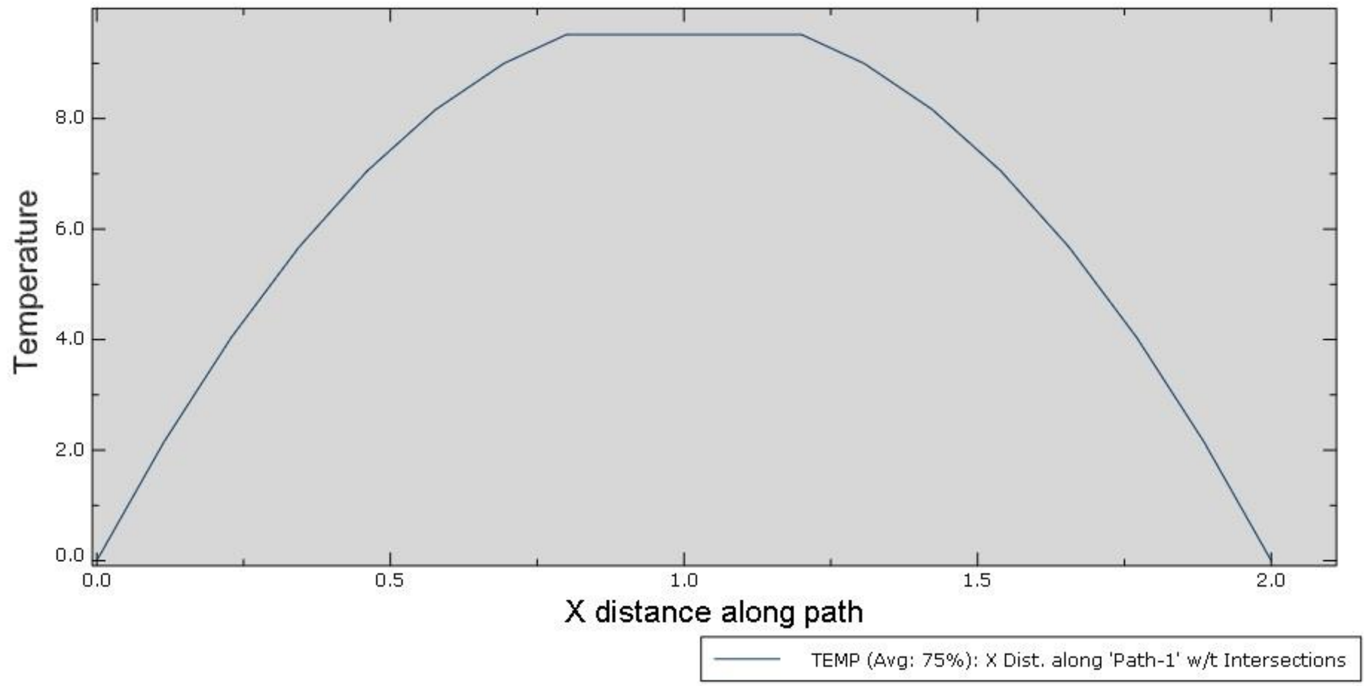
3. JUSTIFICATION FOR THE ABAQUS SOLUTION

The following points justify that the Abaqus solution is reasonable.

1. The Abaqus solution converges perfectly with our analytical and Matlab FEM solutions. The maximum and minimum temperature locations are captured perfectly by Abaqus.
2. So, a node path along X axis was created in Abaqus in order to plot temperature variations as shown below. The temperature variations along X on the plate are plotted along this nodal path.



The following graph shows the variation of temperature along the nodal path (along X axis) between co-ordinates (-1, 1). As expected, temperature varies along a parabola (because of the x^2 term in source) expression from (-1, -0.2) until hole (-0.2, 0.2) where it encounters a discontinuity in the values and then again from (0.2, 1), it varies as a parabola. Similar graph is obtained in the Y direction for temperature variations.

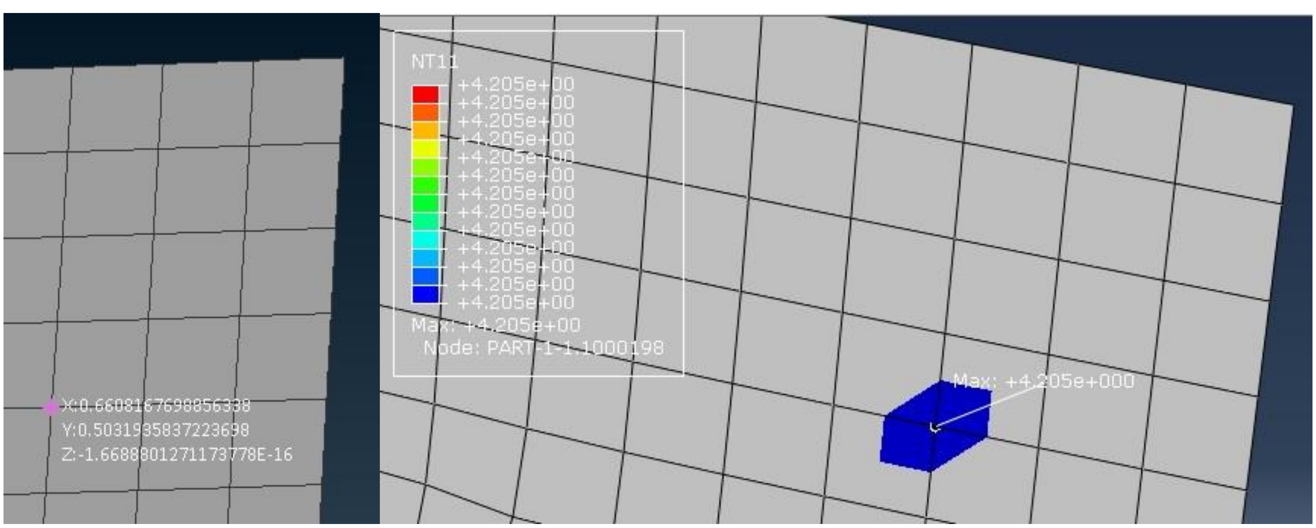


3. We will also compare the temperatures obtained by analytical solution and Abaqus solutions at an intermediate location on the plate.

   Consider the location: (0.661, 0.503) – nodal co-ordinates for any random node

   Temperature obtained by analytical solution: 4.206°C

   Temperature obtained by Abaqus solution: 4.205°C (as shown below)



APPENDIX: MATLAB code(s)

```
%Program to solve 2D Heat transfer equation
%
%  ∇.q - s = 0; on Ω  where q = -k∇T
%
%  With b.c
%  T = 0 on Γ_T & q on Γ_q
%  No. Of elements along half side 'nel' is the input argument.
%  Returns the computed temperature distribution d, the norm errors
%  and the field  plots.

function [h] = solve_fem2(nel)
%PREPROCESSING
a = 2;                                            %Dimension of the plate
r = 0.2;                                          %Radius of the hole
k = 54;                                           %Conductivity W/(mK)
Tbar = 0;                                         %Boundary temperature
D = k*eye(2);

%Set counter = n to compute max temperature for 'n' consecutive mesh values
counter = 1;

%ASSEMBLY
for j = 1:counter
    m = make_mesh(nel,a,r);
    h.en = length(m.conn);                        %Total number of elements
    h.nn = length(m.x);                           %Total number of nodes
    connflux = [(h.nn+1-8*nel):h.nn;
        (h.nn+2-8*nel):h.nn,(h.nn+1-8*nel)];%Flux boundary connectivity

    qpts = [[-1, 1, 1,-1;
        -1,-1, 1, 1] / sqrt(3.0);
        1, 1, 1, 1];                              %Quadrature points

    f = zeros(h.nn,1);
    K = zeros(h.nn);
    fq = zeros(h.nn,1);

    for c = m.conn
        xe = m.x(:,c);
        for q = qpts
            [N,dNdp] = shape(q);
            xq = xe*N';
            J = xe*dNdp';
            B = dNdp'/J;
            s = -10*k*(2*xq(1)^2+2*xq(2)^2-a^2);    %in W/m^2.
            K(c,c) = K(c,c) + B*D*B'*det(J)*q(3);   %Global Stiffness
            f(c) = f(c) + N'*s*det(J)*q(3);         %Force due to heat source
        end
    end
    m.x(:,end)
    m.x(:,end-1)
    le = norm((m.x(:,end)-m.x(:,end-1)));           %Flux boundary length
    for c = connflux
        xe = m.x(:,c);
        for q =[-1 1]/sqrt(3)
            N = 0.5*[1+q 1-q];
            xq = N*xe';
```

```
            qq = -k*20*[xq(1)*(xq(2)^2-a^2/4);xq(2)*(xq(1)^2-a^2/4)];
            n = [xq(1);xq(2)]/r;
            fq(c) = fq(c) + N'*(qq'*n*le/2);       %Force due to heat flux
        end
    end
    f = f + fq;                                    %Global force vector

    %BOUNDARY CONDITIONS
    % Enforce essential boundary condition (nodes on the plate edges)
    for i = 1:(8*nel)
        K(i,:) = 0.0;
        K(i,i) = 1.0;
        f(i) = Tbar;
    end

    %SOLUTION
    h.d = K\f;
    S(j) = max(h.d);  %Max. temperature values for different mesh
    nh(j) = nel;      %Different mesh inputs
    nel = nel+1;
end

%POST PROCESSING

figure(1),plot(m.x(1,1:end), m.x(2,1:end), 'o')
title('Mesh')

% %Contour plot of approximated temperature field
for c = m.conn
    de = h.d(c);
    xe = m.x(:,c);
    hold on
    figure(2),patch(xe(1,:),xe(2,:),de);
    xlabel('x(m)');ylabel('y(m)');
    title('Contour plot of approximated temperature field T(x,y)')
    colorbar; axis equal; ylabel(colorbar,'----   ^oC   ----');
end

%Heat flux in the center of each element
qq = [];
xx = [];
for c = m.conn
    xe = m.x(:,c);
    de = h.d(c);
    [N,dNdp] = shape([0.5; 0.5]);
    J = xe*dNdp';
    B = dNdp'/J;
    qq(end+1,:) = -k*h.d(c)'*B;
    xx(end+1,:) = xe*N';
    hold on
    figure(3),patch(xe(1,:),xe(2,:),de);
    colorbar; axis equal; ylabel(colorbar,'----   ^oC   ----');
end
figure(3)
quiver(xx(:,1),xx(:,2),qq(:,1),qq(:,2),0.25,'linewidth',3,'color','k');
xlabel('x(m)');ylabel('y(m)');
title('Plot of heat flux q in the center of each element')

%Plot of maximum temperature vs mesh
```

```
%%%Set counter to 10 or 11 and suppress all the other plots
figure(4),plot(nh,S,'rx-')
xlabel('mesh');ylabel('Maximum temperature ^oC');
title('Plot of maximum temperature for different mesh size')

% %Plot of temperature along the center y=0 || -l/2 < x < l/2
kk = find(abs(m.x(2,:))<= 1e-10);
left = kk(2:2:end); right = kk(end-1:-2:1);
figure(5),plot(m.x(1,left),h.d(left),'b*-',m.x(1,right),h.d(right),'b*-')
xlabel('x(m)');ylabel('T ^oC');
title('Temperature along the center line (FEM Solution)')

xx = [-a/2:0.02:-0.2, 0.2:0.02:a/2];
for x = xx
    T = 10*(x-a/2)*(x+a/2)*(-a/2)*(a/2);
    hold on
    figure(6),plot(x,T,'rx')
    xlabel('x(m)');ylabel('T ^oC');
    title('Temperature along the center line (Theoretical solution)')
end
end

%COMPUTES THE RATE OF CONVERGENCE
function [dconv] = convergence2
a = 2;                                  %Length of the rod
r = 0.2;                                %Radius of the hole
qpts = [[-1, 1, 1,-1;
        -1,-1, 1, 1]/sqrt(3.0);
         1, 1, 1, 1];                   %Quadrature points

L2 = zeros(1,5);                        %Soltuion values for L2 error norm
h = zeros(1,5);                         %Element size for each solution
m = 1;
mesh = [1 2 3 4 5];

for elm = mesh
    mm = solve_fem2(elm);
    err1 = 0; dnorm = 0;
    tt = make_mesh(elm,a,r);
    h(m) = sqrt((tt.x(1,2)-tt.x(1,1))^2 + (tt.x(2,2)-tt.x(2,1))^2);
    for c = tt.conn
        xe = tt.x(:,c);
        for q = qpts
            [N,dNdp] = shape(q);
            xq = xe*N';
            J = xe*dNdp';
            de = mm.d(c);
            %Exact solution
            T = 10*(xq(1)-a/2)*(xq(1)+a/2)*(xq(2)-a/2)*(xq(2)+a/2);
            %FEM solution
            Th = N*de;
            %L2 Error -Norm
            err1 = err1+((T-Th)^2)*det(J)*q(3);
            dnorm = dnorm +((T)^2)*det(J)*q(3);
        end
    end
    L2(m) = sqrt(err1)/sqrt(dnorm);
    m = m+1;
end
```

```matlab
figure(7),plot(abs(log10(h)), abs(log10(L2)), 'ro-')
title(sprintf('%d, %d, %d, %d, mesh elements',.../
    mesh(1),mesh(2),mesh(3),mesh(4)));
xlabel('Log of Element size (m)'); ylabel('Log of L2 error norm');
%Rate of Convergence for L2 error norm
dconv = abs(diff(log10(L2)))./abs(diff(log10(h)));
end

function [N,dNdp] = shape(p)
N = 0.25*[(1-p(1))*(1-p(2)),(1+p(1))*(1-p(2)),(1+p(1))*(1+p(2)),.../
    (1-p(1))*(1+p(2))];
dNdp = 0.25*[(p(2)-1),(1-p(2)),(p(2)+1),-(p(2)+1);
    (p(1)-1),-(p(1)+1),(p(1)+1),(1-p(1))];
end

function [d,T] = patchtest

% PATCH TEST %
Node and Element connectivity matrix for the mesh
x = [1 -1 -1 1 0.0347 -0.1970 -0.1286 0.1414;
    1 1 -1 -1 0.1970 0.0347 -0.1532 -0.1414];
conn = [1 2 3 4 5;2 3 4 1 6;6 7 8 5 7;5 6 7 8 8];

% Mesh
x2d = x(1,1:end);
y2d = x(2,1:end);
labels = cellstr( num2str([1:length(x)]') );
figure(1),plot(x2d, y2d, 'o')
text(x2d, y2d, labels)
title('Mesh')

%PREPROCESSING
D = 54*eye(2);              %Conductivity matrix W/(mK)
en = length(conn);          %Total number of elements
nn = length(x);             %Total number of nodes

%ASSEMBLY
qpts = [[-1, 1, 1,-1;
         -1,-1, 1, 1] / sqrt(3.0);
          1, 1, 1, 1];      %Quadrature points

f = zeros(nn,1);
K = zeros(nn);
T = zeros(nn,1);
for c =conn
    xe = x(:,c);
    for q = qpts
        [N,dNdp] = shape(q);
        J = xe*dNdp';
        B = dNdp'/J;
        K(c,c) = K(c,c) + B*D*B'*det(J)*q(3);
    end
end

%BOUNDARY CONDITIONS
% Enforce essential boundary condition (nodes on the plate edges)
f(1) = 11; f(2) = 1; f(3) = -7; f(4) = 3;
for i = 1:4
```

```matlab
        K(i,:) = 0.0;
        K(i,i) = 1.0;
    end
    format long

    %SOLUTION
    d = K\f;

    %TEST
    i = 1;
    for j = 1:8
        T(i) = 2+5*x(1,j)+4*x(2,j);
        i = i+1;
    end
        if abs(T(:) - d(:)) >= 10^-8
            fprintf('\nPatch test fail\n\n\n');
        else
            fprintf('\nPatch test pass\n\n\n');
        end
    end

    %Shape function & its derivate for a 4-node quadrilateral element
    function [N,dNdp] = shape(p)
    N = 0.25*[(1-p(1))*(1-p(2)),(1+p(1))*(1-p(2)),(1+p(1))*(1+p(2)),.../
                (1-p(1))*(1+p(2))];
    dNdp = 0.25*[(p(2)-1),(1-p(2)),(p(2)+1),-(p(2)+1);
             (p(1)-1),-(p(1)+1),(p(1)+1),(1-p(1))];
    end

    %run analytical(a,r,k) for (2,0.2,54)

    function [] = analytical(a,r,k)

    %Analytical Solution
    %%heat source and heat flux plot
    x=-1:0.1:1;
    y=-1:0.1:1;
    [x1,y1]= meshgrid(x,y);
    for l=1:length(x)
        for m=1:length(y)
            s(l,m) = -10*k*(2*x(l)^2 + 2*y(m)^2 - a^2);
            if (x(l)^2+y(m)^2 >= r^2)
                qx(l,m)=-20*k*x(l)*((y(m)^2)-1);
                qy(l,m)=-20*k*y(m)*((x(l)^2)-1);
            end
            hold on
        end
    end
    figure(2),contourf(x1,y1,s);colorbar
    xlabel('x(m)');ylabel('y(m)');title('s(x,y) Heat source field')
    ylabel(colorbar,'----   W/m^2   ----');
    ang=0:0.05:2*pi;
    xp=r*cos(ang); yp=r*sin(ang);
    figure(3),plot(xp,yp,'k','linewidth',1);
    xlabel('x(m)');ylabel('y(m)');title('Analyticial Heat flux')
    hold on
    figure(3),quiver(y1,x1,qx,qy,0.5,'linewidth', 1.5);
    hold on
    rectangle('Position',[-1 -1 2 2]); axis([-1 1 -1 1]);
```