ARIZONA STATE UNIVERSITY

MAE 503

FINITE ELEMENTS IN ENGINEERING

Professor- Dr. Jay Oswald

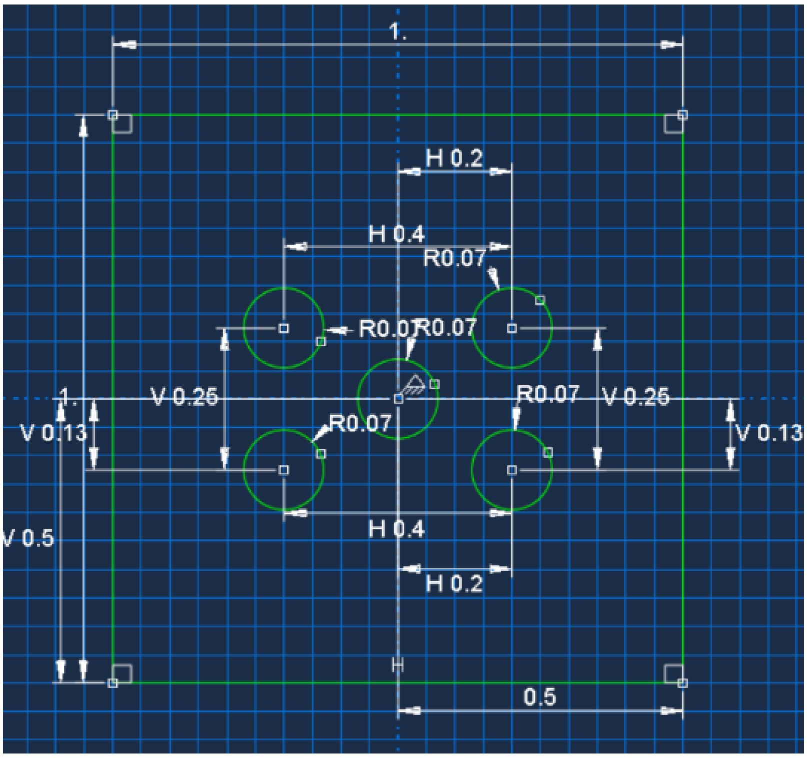
Onkar Chavan | ochavan2 | 1223314248

Mangirish Kulkarni | mkulka17 | 1223229852

Kaustaubh Nalawade | kunalawa | 1222645996

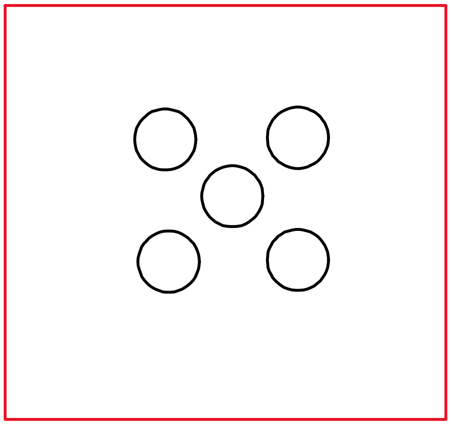
Problem Statement

We are solving a problem where we need to calculate the heat flux and temperature fields of a ceramic housing which has load as heat flux generated by nuclear fuel rods. Their decay results in a heat flux of $20\ kW/m^2$. Ceramic housing has thermal conductivity of 17 Watts/(meter-Kelvin) and its dimensions are 1 m x 1 m and it has 5 nuclear fuel rods of radius 0.07 m. Its exact location is as shown in figure below. Boundary conditions of the housing are that the outer side exerts heat which is converted at a rate of $h = 10\ W/m^2K$ and ambient as 0 K.



- First, we modelled the problem in Abaqus and then determined the temperature and heat flux contour plots of the ceramic housing.
- We compared the plots between results of MATLAB and Abaqus side by side.
- For Richardson extrapolation:
  1. First determined the order of convergence p
  2. Calculate Richardson extrapolation with 2 finest grids to estimate solution at h = 0
  3. Calculated the Grid convergence Index
  4. Asymptotic range of convergence.

Friday, April 15, 2022 9:57 PM



Suppose the system is under steady state condition.

$$\oint_{\Gamma_i} \vec{q} \cdot \vec{n} \, d\Gamma = \text{heat flowing out of } \Omega_i \qquad (\vec{q} = \text{power/area})$$

$$\int_{\Omega_i} s \, d\Omega = \text{heat generated within } \Omega_i \qquad \left(s = \frac{\text{power}}{\text{volume or area}}\right)$$

$$\dot{w} = \int_{\Omega_i} s \, d\Omega - \oint_{\Gamma_i} \vec{q} \cdot \vec{n} \, d\Gamma = 0 \quad \text{(steady state)}$$

$$= \int_{\Omega_i} s \, d\Omega - \int_{\Omega_i} \vec{\nabla} \cdot \vec{q} \, d\Omega = \int_{\Omega_i} s - \vec{\nabla} q \, d\Omega$$

Must Hold for any $\Omega_i \in \Omega$

$$\Rightarrow \vec{\nabla} \vec{q} - s = 0$$

In this case

$$\vec{\nabla}\left(-h(T-T_\infty) + \vec{q}\right) - 0 = 0$$

... Since heat generated within the system is 0.
& $\vec{q}$ = heat supplied to the system

Strong Form

$$\vec{\nabla}\left(-h(T-T_\infty) + \vec{q}\right) = 0 \quad \text{on } \Omega$$

$$\vec{q} = -k \nabla T \quad \text{Fourier's Law}$$

Strong form

Energy Balance	$\vec{\nabla}\left(-h(T-T_\infty) - k\nabla T\right) = 0$
Fourier's Law	$\vec{q} = -k\nabla T$
natural BC	$q_n = \vec{q} \cdot \vec{n} = \bar{q}(x,y)$ on Γ_q
essential BC	$T = \bar{T}(x,y)$ on Γ_T

Units -

We have used SI Units for the entire project. This consistency has been maintained for heat flux, thermal conductivity, heat transfer co-efficient and radius of system for both MATLAB and Abaqus simulations.

Name	Units
Length	Meters (m)
Temperature	Kelvin (K)
Heat Flux	Watt/meter^2 (W/m^2)
Convective heat transfer co-efficient	Watt/meter^2*Kelvin (W/m^2.K)
Thermal Conductivity	Watt/meter*Kelvin (W/m.K)

Getting weak form

$$\int_\Omega w\left(\vec{\nabla}\left(-k\vec{\nabla}T\right) - h(T-T_\infty)\right)d\Omega = 0$$

Integrating by parts we get

$$\int_\Omega \left(\vec{\nabla}\left(wk\vec{\nabla}T\right)\right)d\Omega + \int_\Omega\left(\vec{\nabla}w\cdot k\vec{\nabla}T\right)d\Omega - \int_\Omega (whT)\,d\Omega + \int (whT_\infty)\,d\Omega = 0$$

$$\int_\Omega \left(\vec{\nabla}\cdot\left(wk\vec{\nabla}T\right)\right)d\Omega = \int_\Gamma \left(\vec{w}\,k\vec{\nabla}T\cdot\vec{n}\right)d\Gamma - \int_\Omega (whT)\,d\Omega + \int whT_\infty\,d\Omega$$

As $w=0$ on Γ_T, & using divergence theorem $\int_\Omega k\vec{\nabla}T\,d\Omega = \oint_\Gamma k\vec{\nabla}T\cdot\vec{n}\,d\Gamma$

$$\boxed{\int_\Omega\left(\vec{\nabla}w\cdot k\vec{\nabla}T\right)d\Omega = \int_\Gamma (w\bar{q})\,d\Gamma + \int_\Omega whT\,d\Omega - \int_\Omega whT_\infty\,d\Omega}$$

Discretization:

$$T(x,y) \approx T^h(x,y) = d_i^e N_i^e$$

$$w(x,y) \approx w^h(x,y) = w_i^e N_i^e$$

$$d^e = L^e d$$

$$w^e = L^e w$$

$$\nabla T \approx d_i^e \nabla N_i^e$$

$$\nabla w \approx w_i^e \nabla N_i^e$$

Substituting the discretized functions in to the weak form gives

$$w^T \sum \left[(L^e)^T \int_{\Omega^e} (B^e)^T D^e B^e\,d\Omega\, L^e\right] d + w^T \sum (L^e)^T \left[\int_{\Gamma_q} (N^e)^T \bar{q}\,d\Gamma - \int_{\Omega^e} (N^e)^T h N^e\,d\Omega + \int_{\Omega^e} \left((N^e)^T h T_\infty\right) d\Omega\right] = 0$$

$$K = \sum (L^e)^T K^e L^e \text{ where } K^e = \int_{\Omega^e} (B^e)^T D^e B^e\,d\Omega$$

$$f_T = \sum (L^e)^T f^e \text{ where } f^e = -\int_{\Gamma_q} (N^e)^T \bar{q}\,d\Gamma$$

$$H = \sum (L^e)^T \int_{\Omega^e} \left((N^e)^T h N^e\right) d\Omega$$

$$f_\Omega = \sum (L^e)^T \int_{\Omega^e} \left((N^e)^T h\cdot T_\infty\right) d\Omega$$

$$w^T\left[Kd + Hd - f_\Omega - f_\Gamma\right] = 0$$

$$(K+H)d = f_\Omega + f_\Gamma$$

Abaqus Report

Below are the detailed steps followed for Abaqus software Simulation-

- **Part Creation**

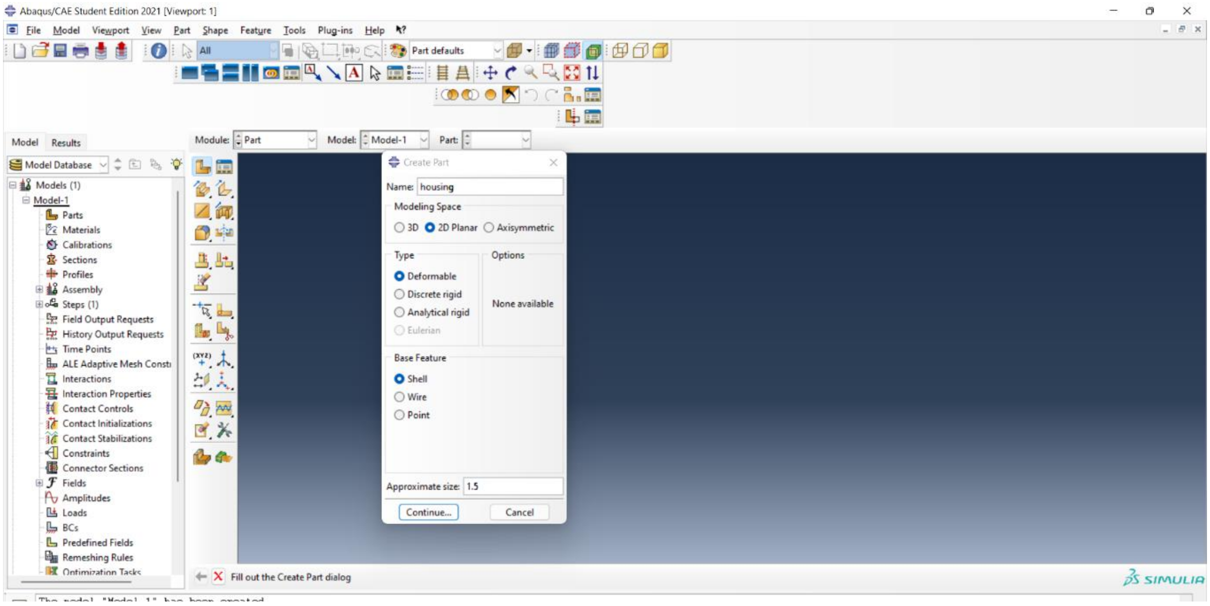
In module selected Part → Create Part

Here Select   Modelling Space → 2D Planar

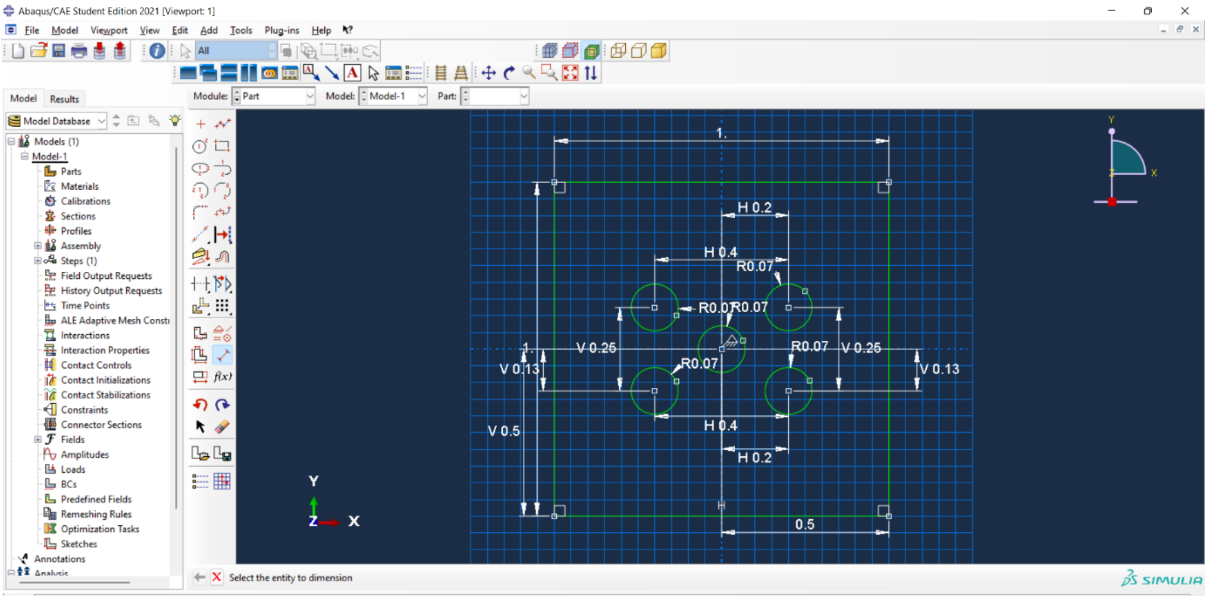
Type → Deformable

Shape → Shell

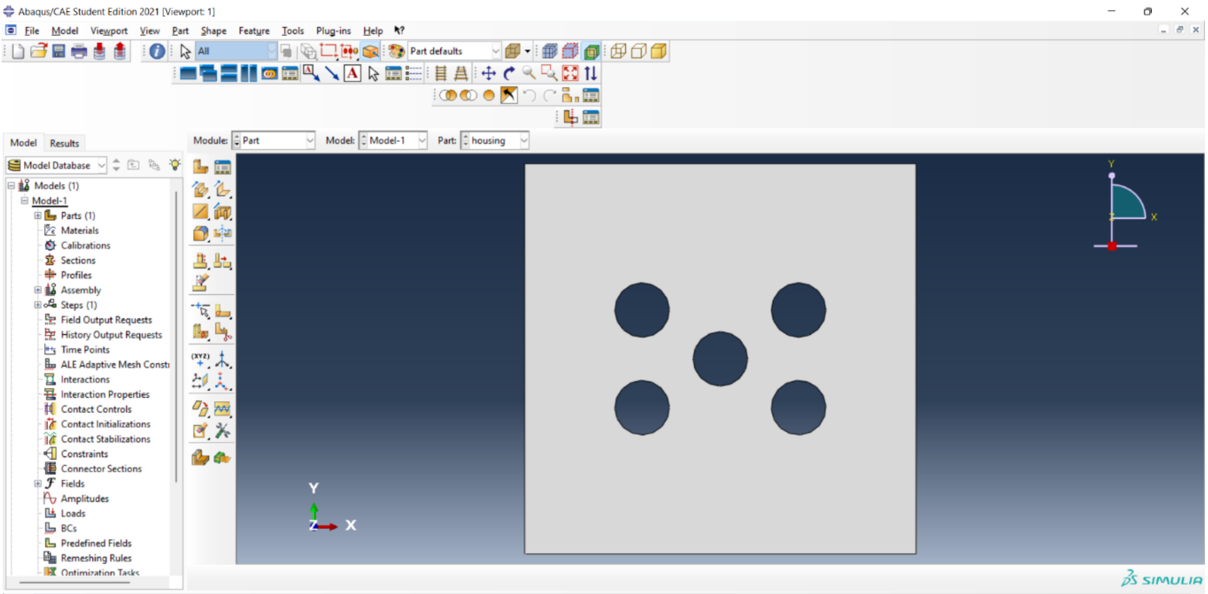
Approximate Size → 1.5



Created the part by given dimensions in the problem.



Exit the sketch and a required 2D design is formed



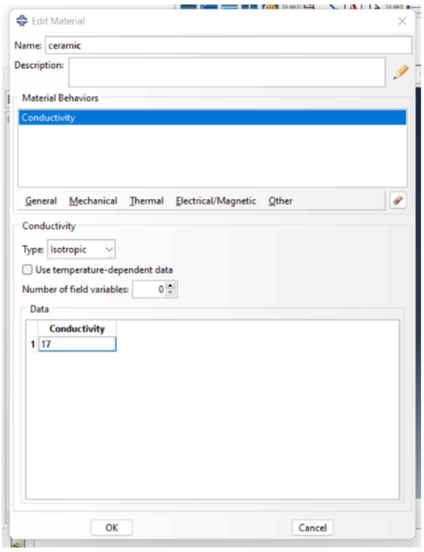
- **Material Creation**

    1. Create Material

In module select Property → Select create Material

Here we entered the given properties of material as below-

Thermal → Conductivity → 17 Watts/(meter-Kelvin)



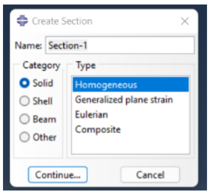
2. Create Section

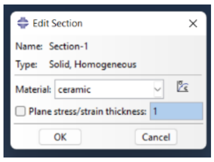
In module select Property ⟶ Create Section

Category ⟶ Solid

Type ⟶ Homogeneous

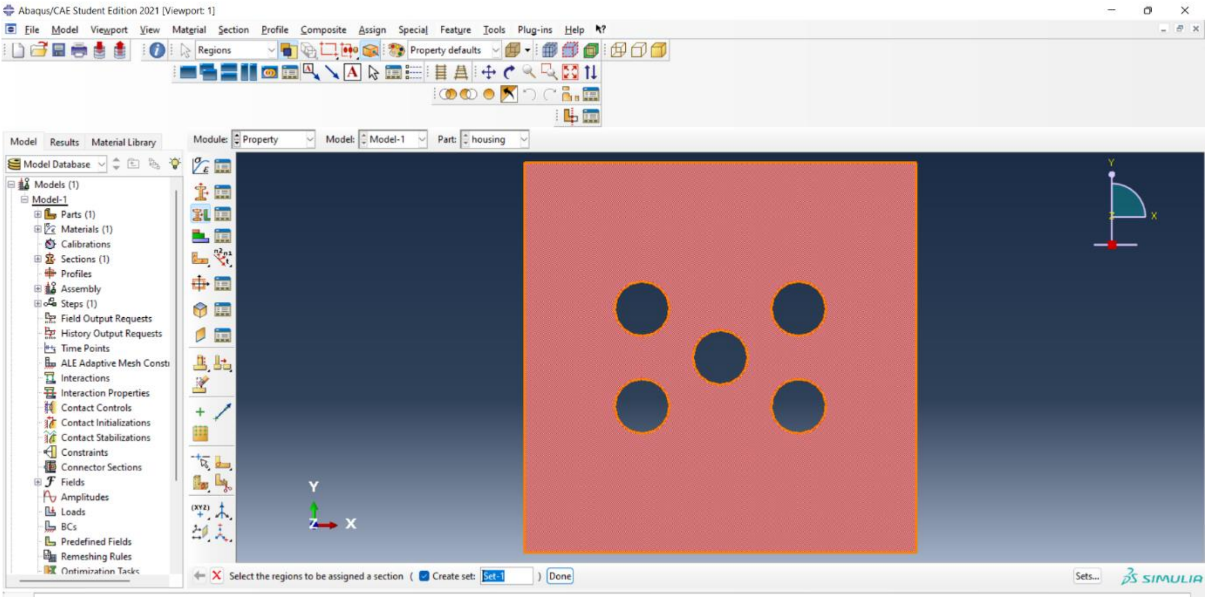
Material ⟶ Ceramic





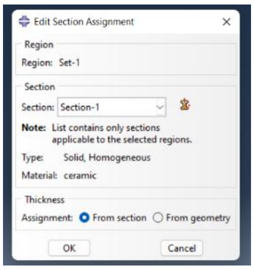
3. In module select Property ⟶ Assign Section

Select the part.



Section → Section-1

Thickness → from Section

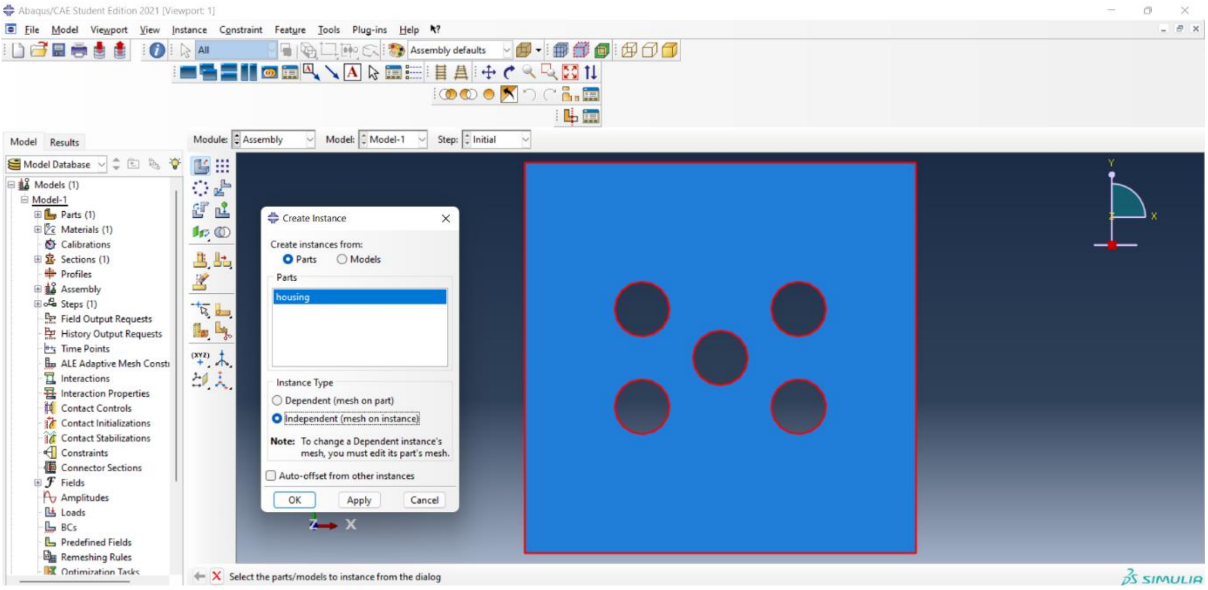


- **Creating Assembly**

Module → Assembly

Create instance → Parts → housing

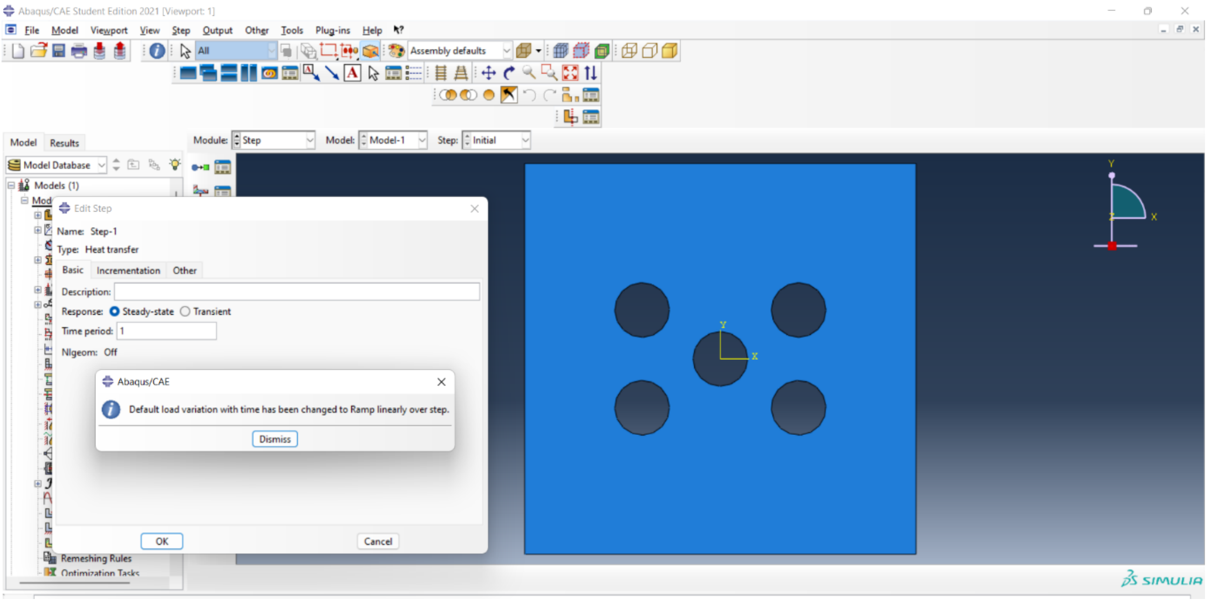
Instance Type → Independent (mesh on instance)



- **Creating Step**

Module → Step → Create Step

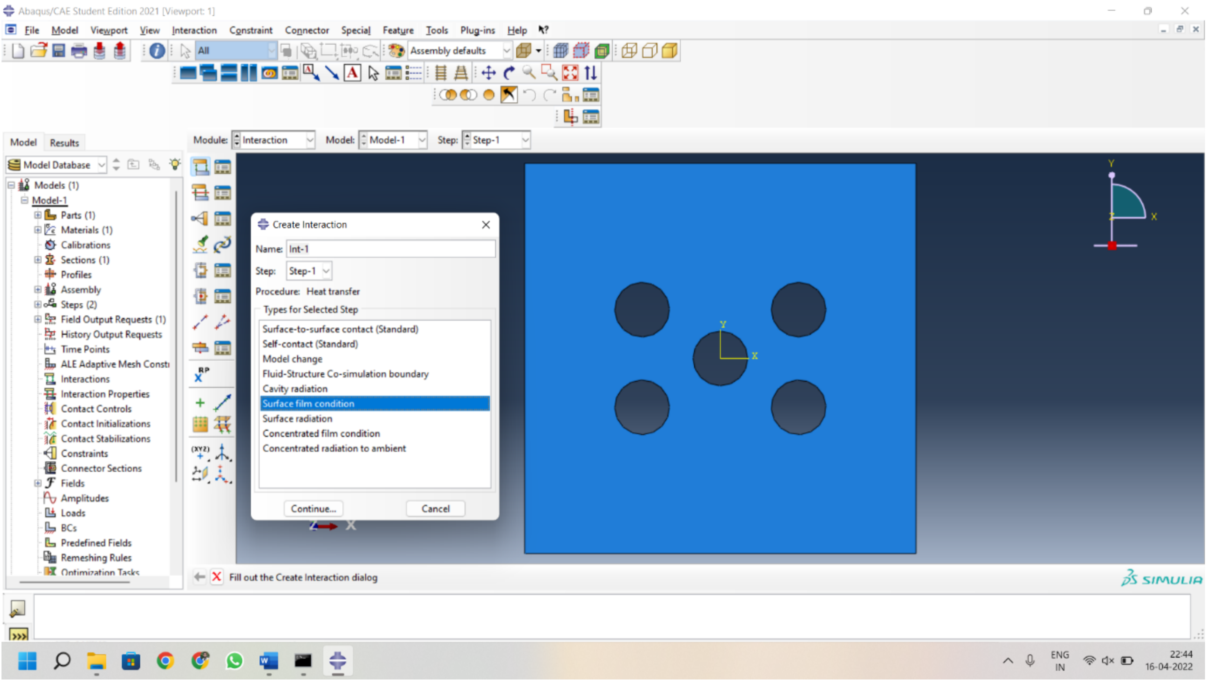
Response → Steady State



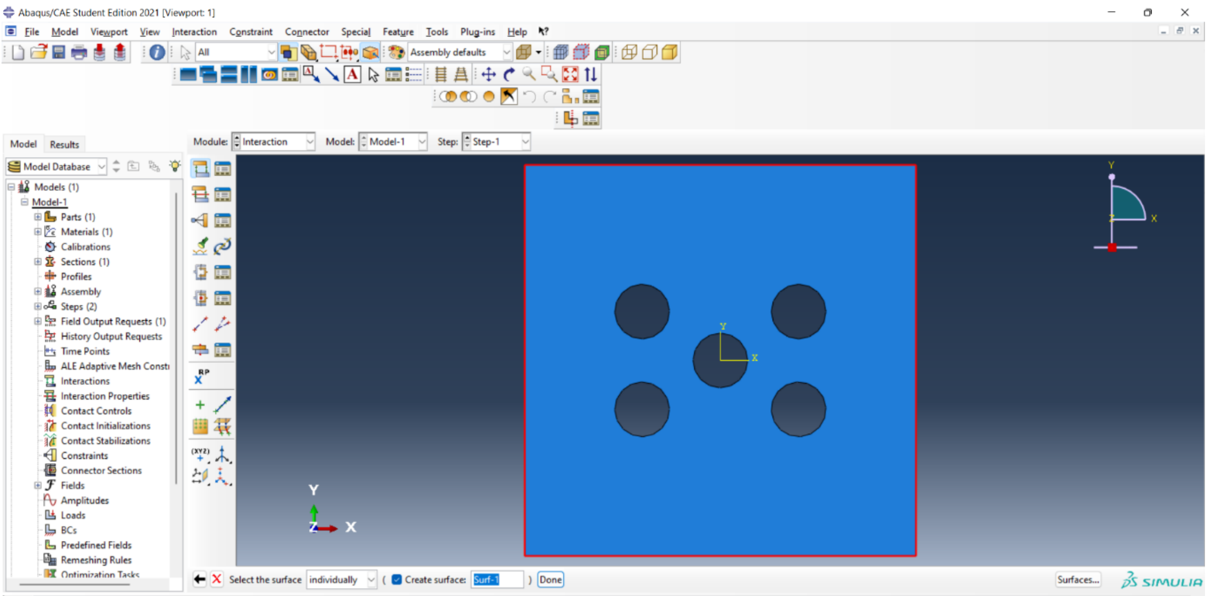
- **Creating Interaction**

Module → Interaction

Types of Selected Steps → Surface film condition

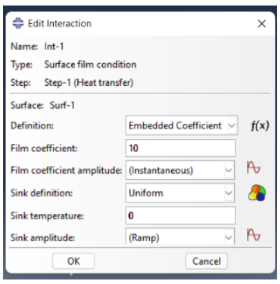


Select all the sides of the plane as boundary conditions act on them.

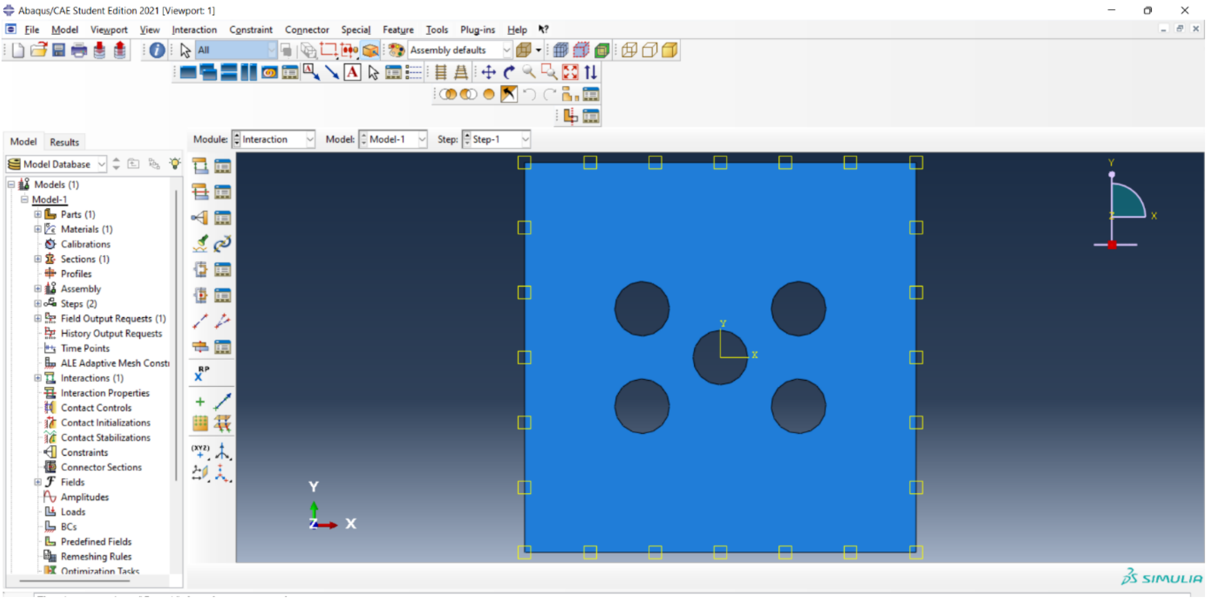


Film coefficient $\rightarrow$ 10 (as $h = 10\ W/m^2K$)

Sink Temperature $\rightarrow$ 0 (assumed to be 0 K)



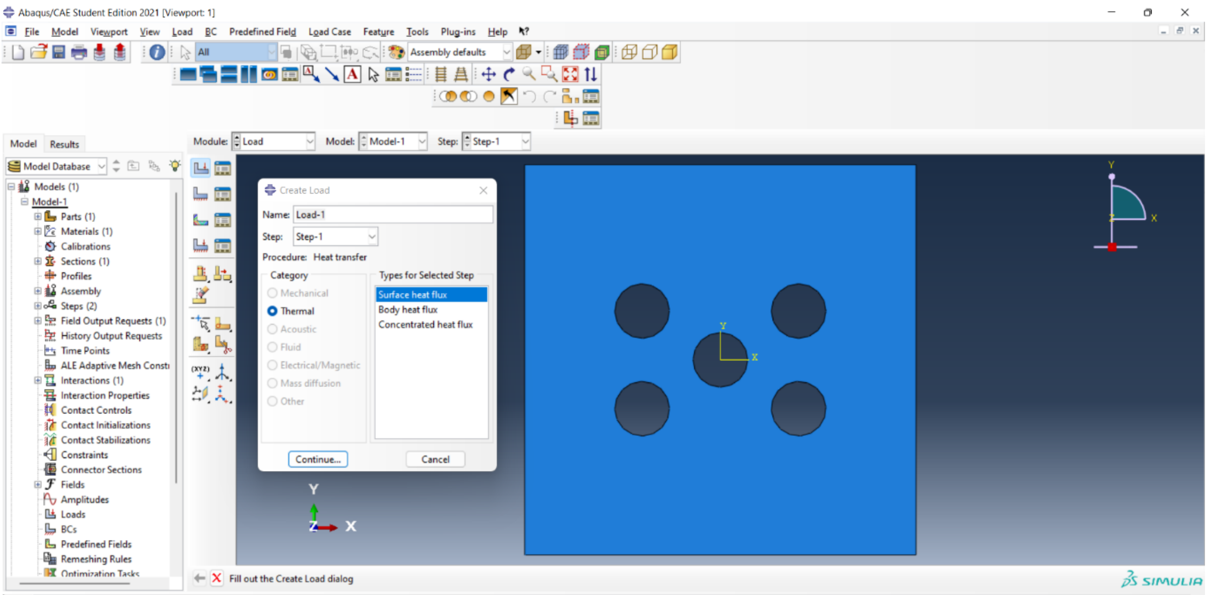
Click OK and it looks like screenshot shown below.



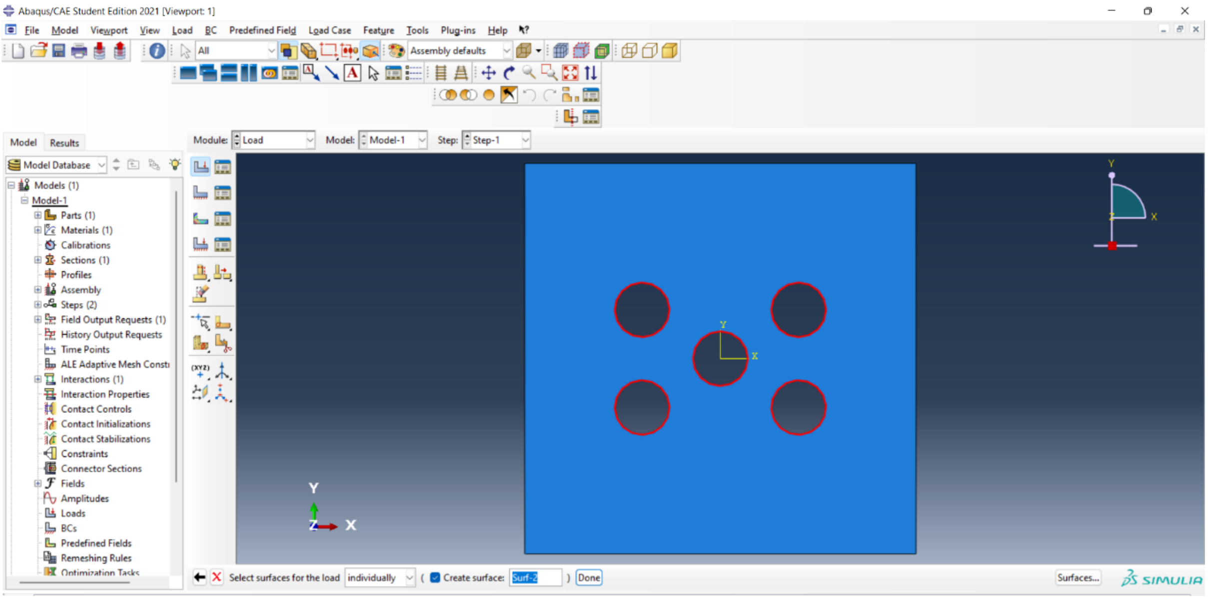
- **Applying Load**

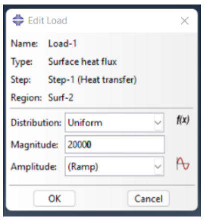
Module → Load

Types for Selected Step → Surface heat flux



Selected the all (5) circular surface for the load and entered the magnitude of 20000 as heat flux is $20\ kW/m^2$

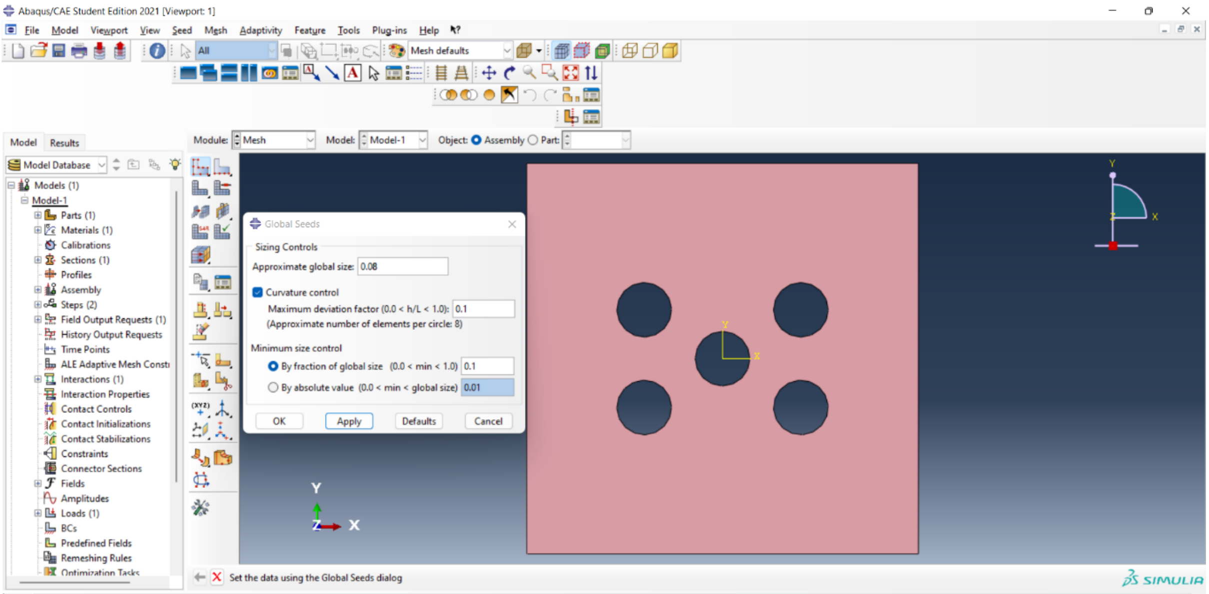


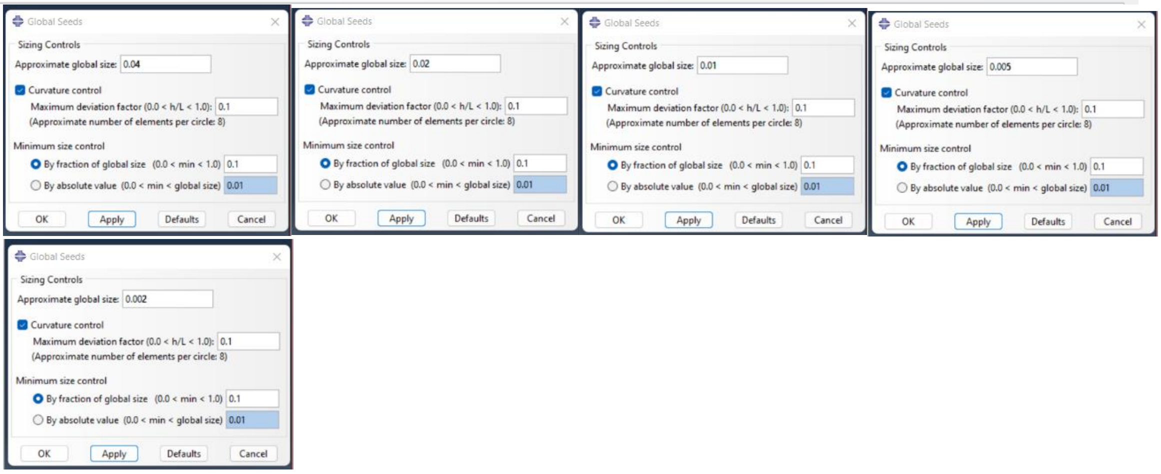


- **Creating Mesh**

Module → Mesh

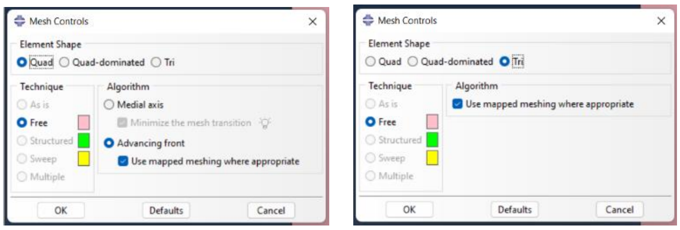
Seed Part → Approximate global size→ 0.08; also performed on 0.002, 0.005, 0.01, 0.02 and 0.04



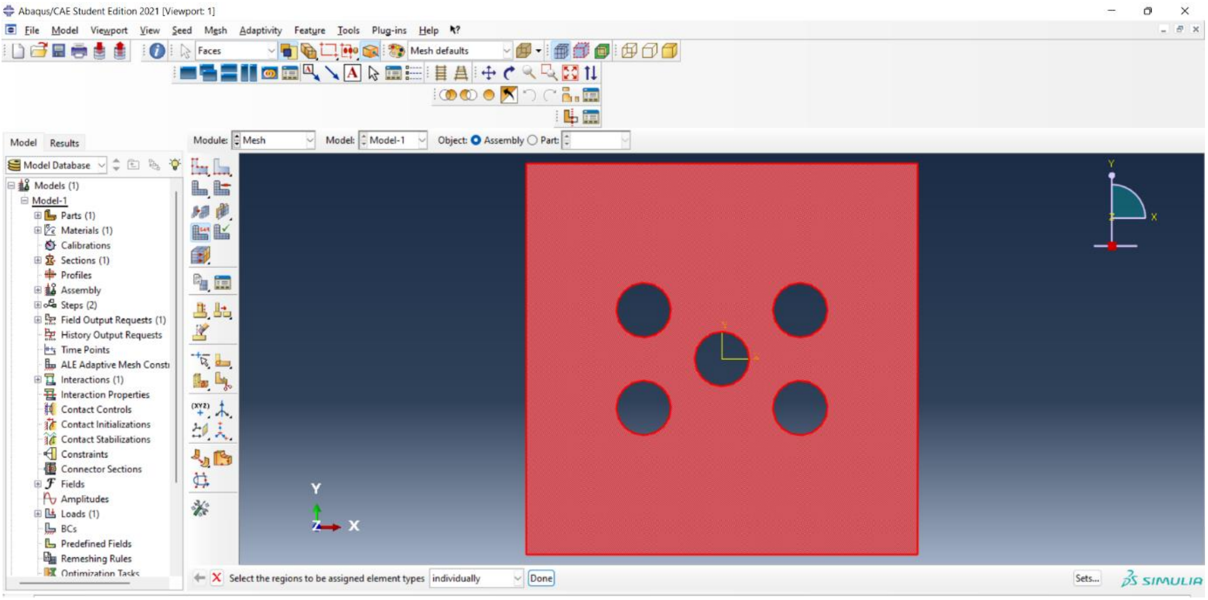


Assign Mesh→ Element Shape→ Quad (for 4 node quadrilateral)

→Tri (for 3 and 6 node triangle)



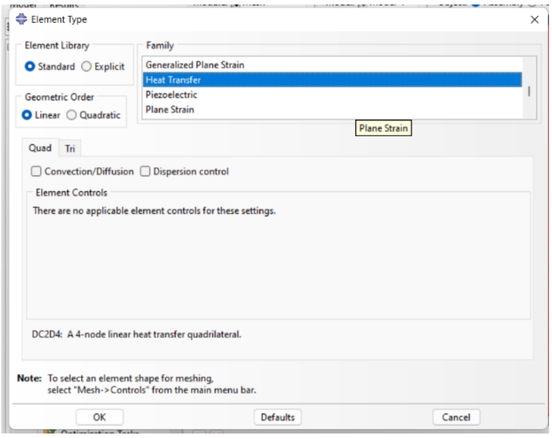
Select Element Type → selected all the regions

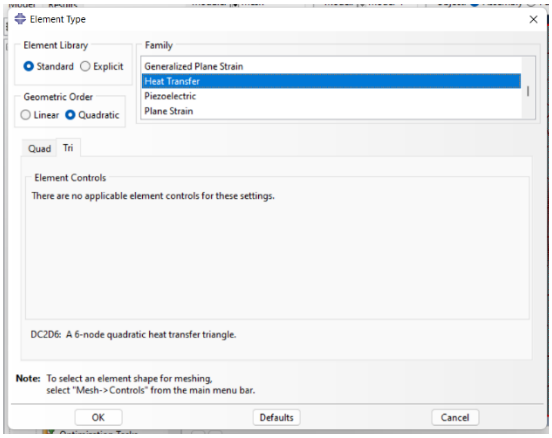


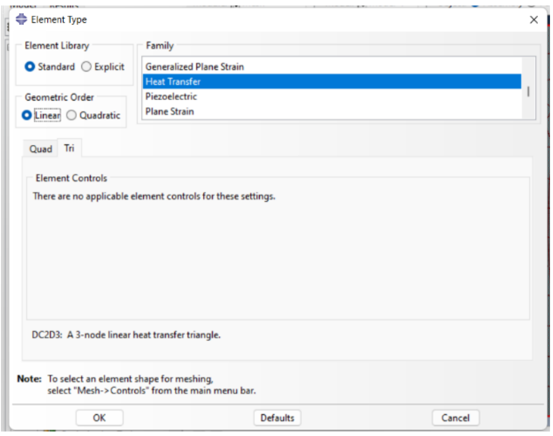
Element Type → Family → Heat Transfer

Geometric Order → Linear (for 4 node quadrilateral and 3 node triangle)

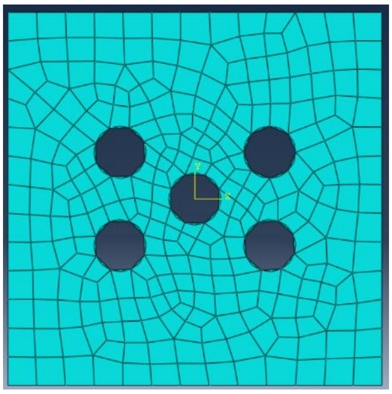
→ Quadratic (for 6 node triangle)

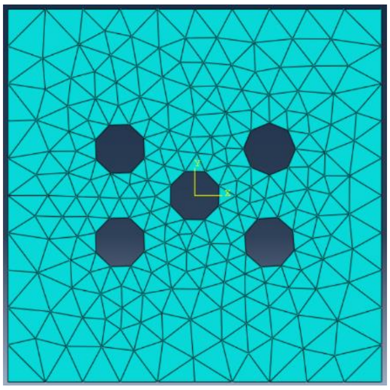


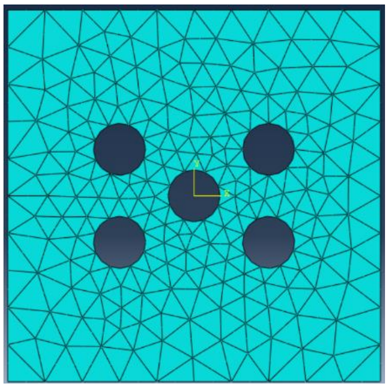




Module → Mesh → Mesh part instance





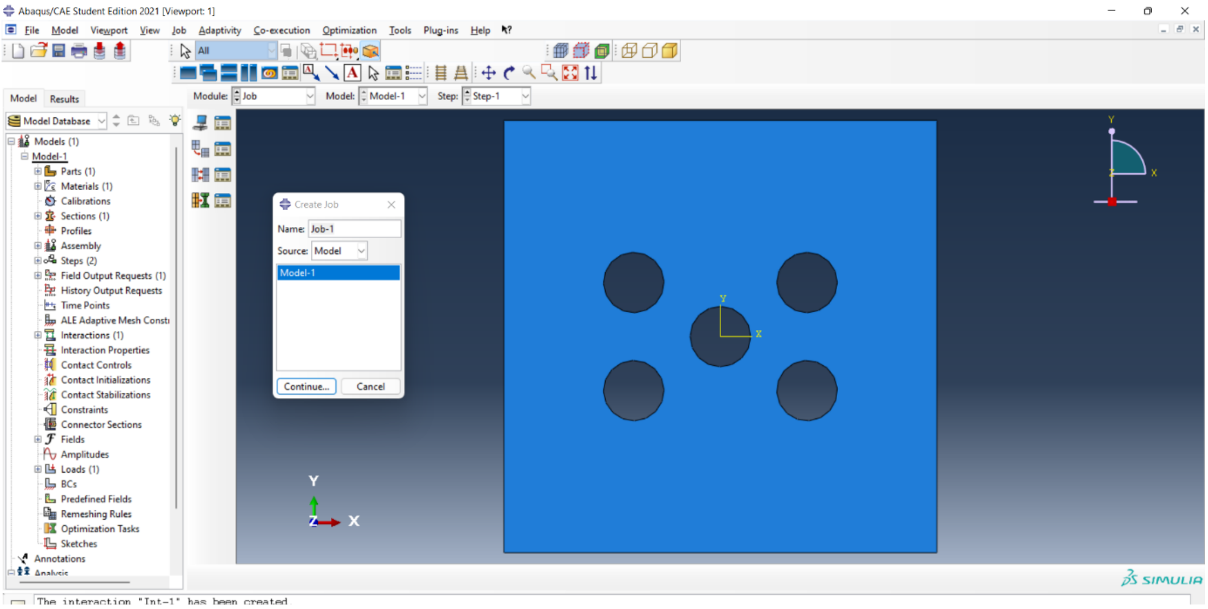


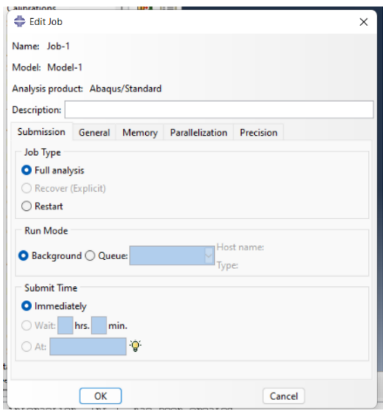
4 node quadrilateral   3 node triangle   6 node triangle

- Creating Job

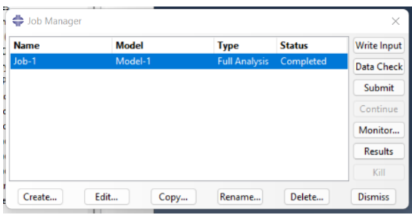
Module → Job

Create new job → Name, Source





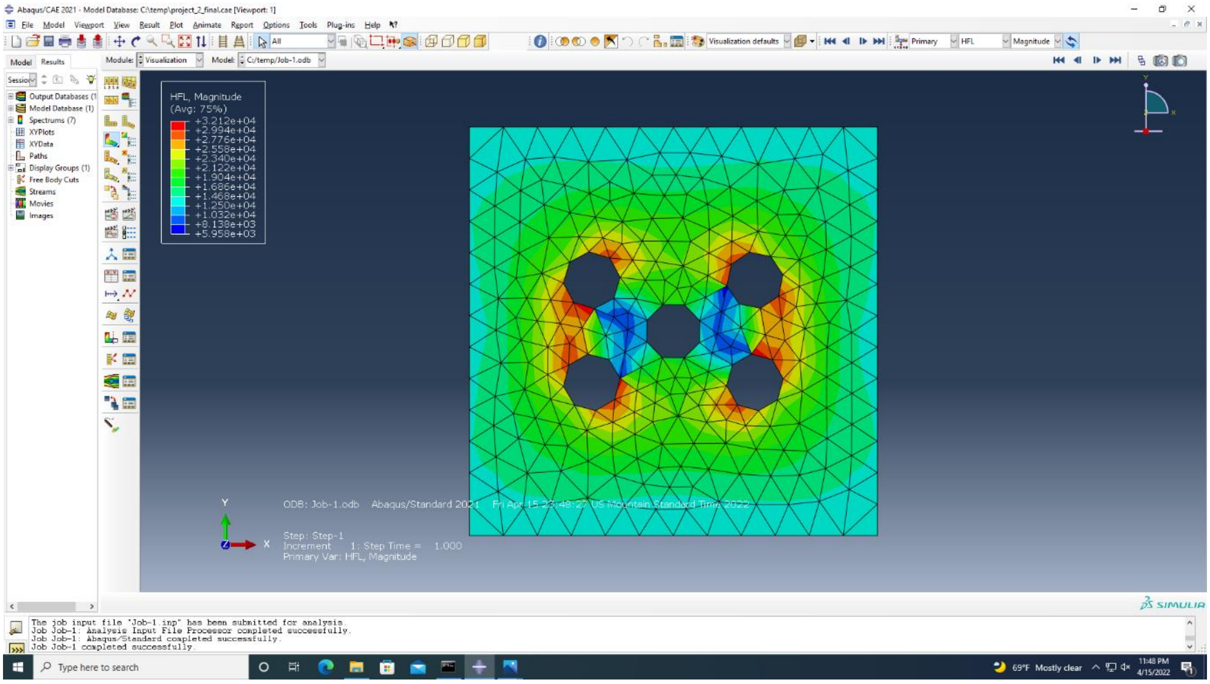
Job Manager → Submit



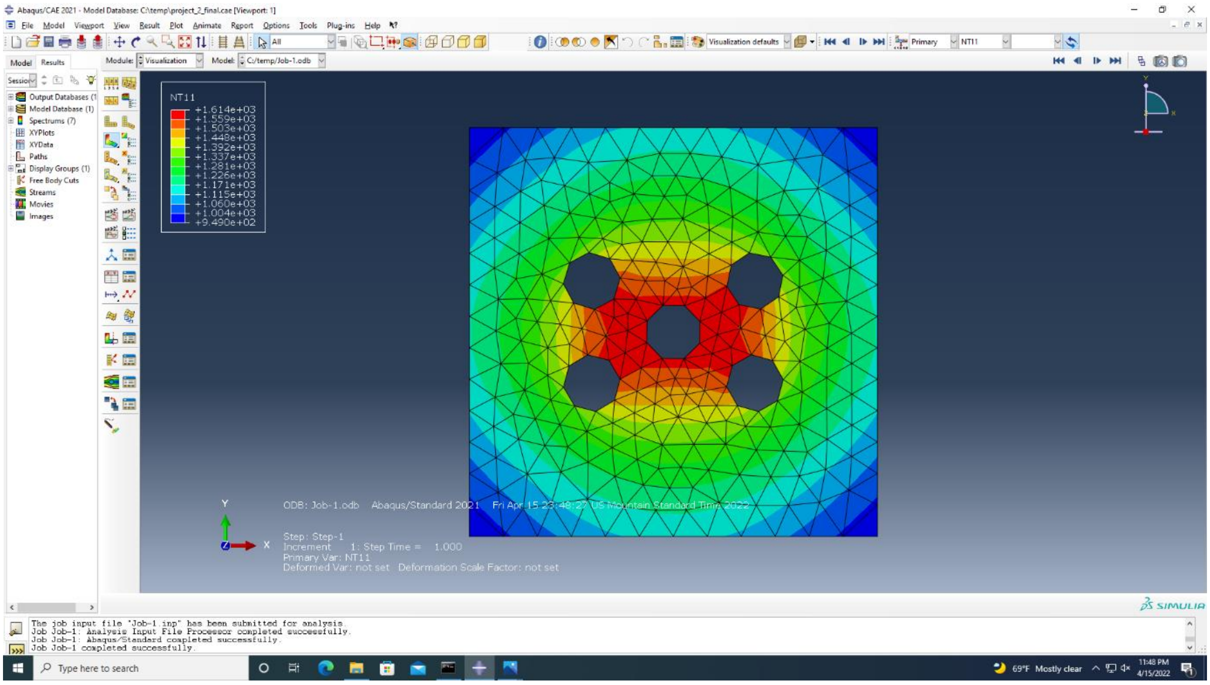
Results → HFL (for heat flux)

→ NT11 (for Temperature)

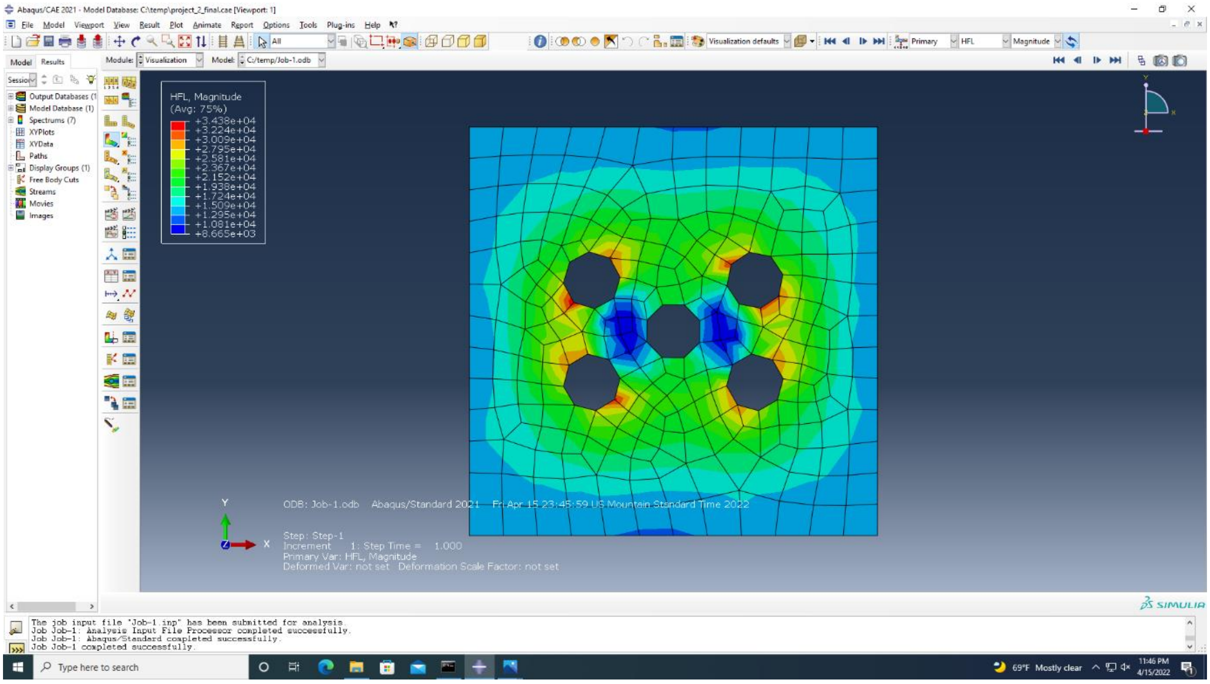
3 node triangle- heat flux



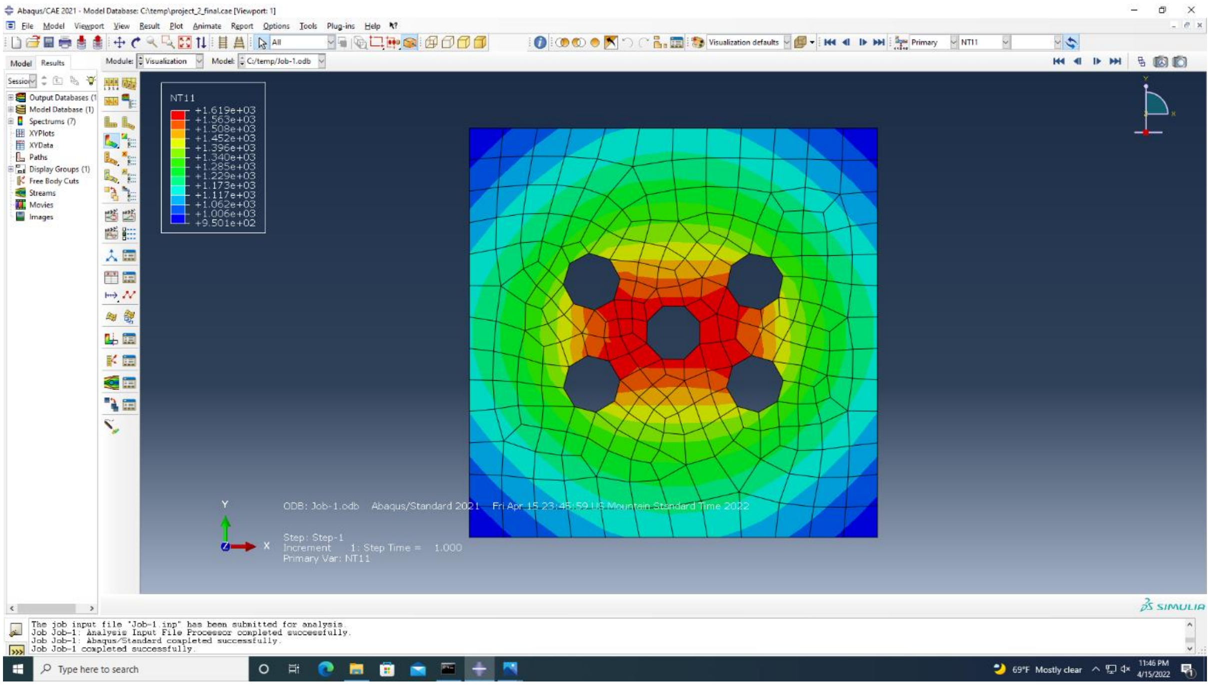
3 node triangle- temperature



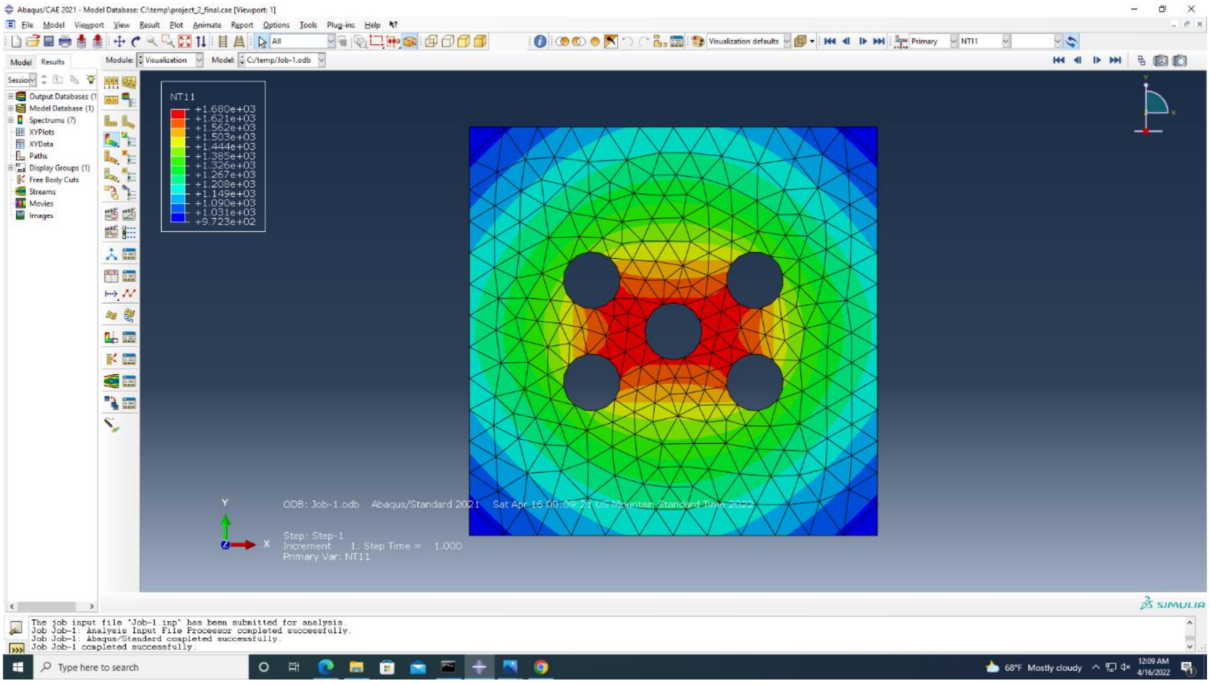
4 node quadrilateral- heat flux



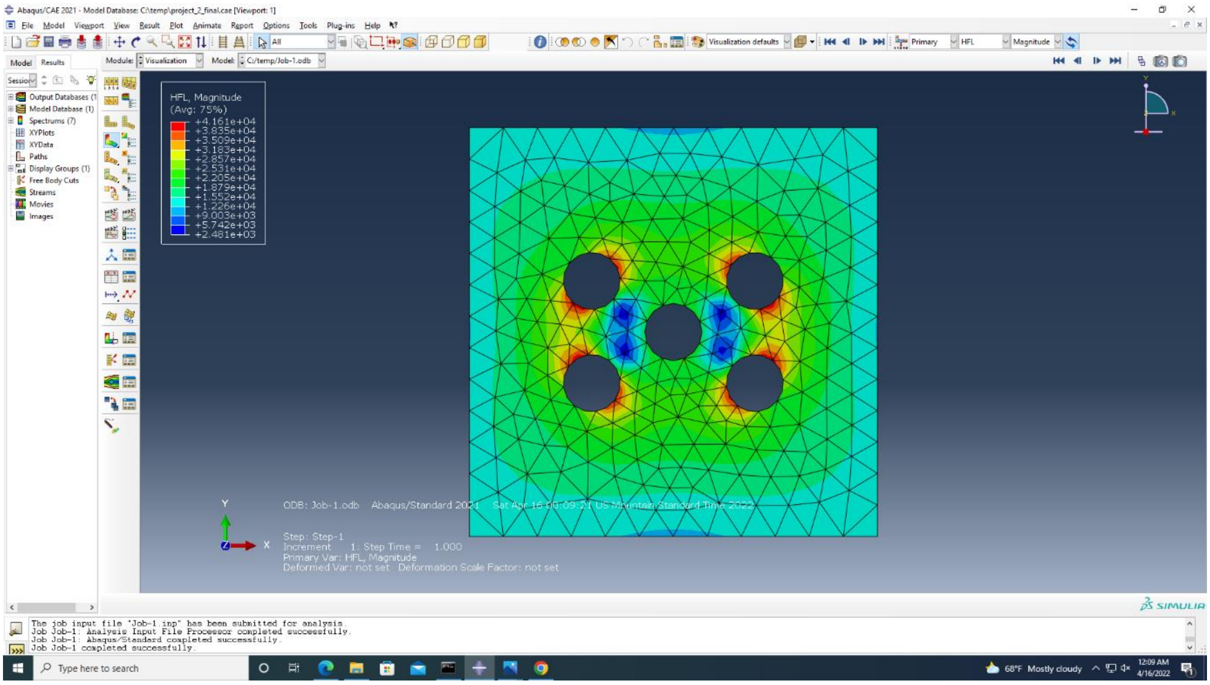
4 node quadrilateral- temperature



6 node triangle- temperature



6 node temperature- heat flux

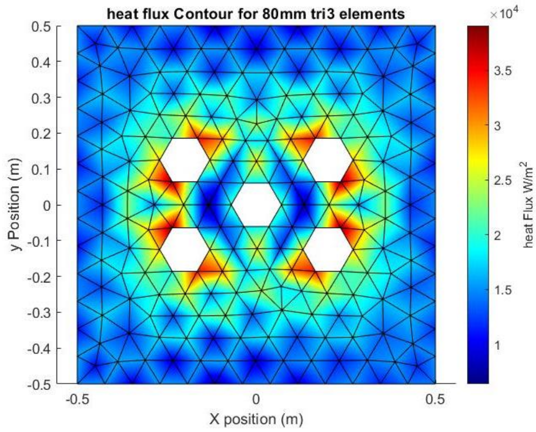


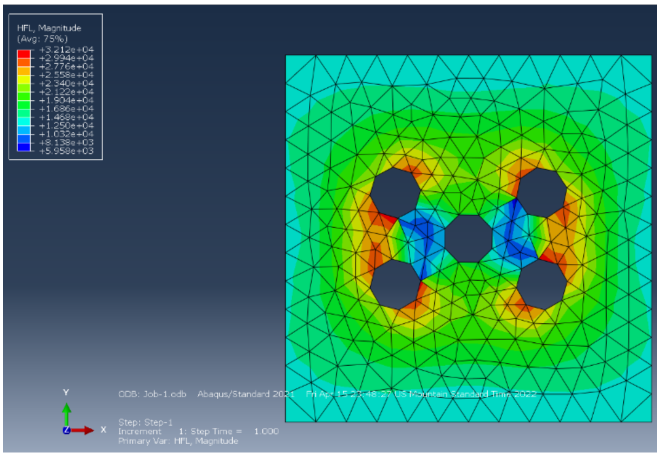
Comparing contour plots of heat flux and temperature.

Below mentioned left side contours plots are from MATLAB and right side contours are from Abaqus.

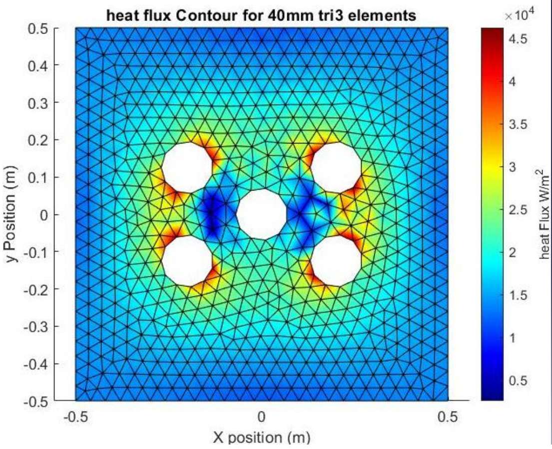
- 3 node triangle Heat Flux

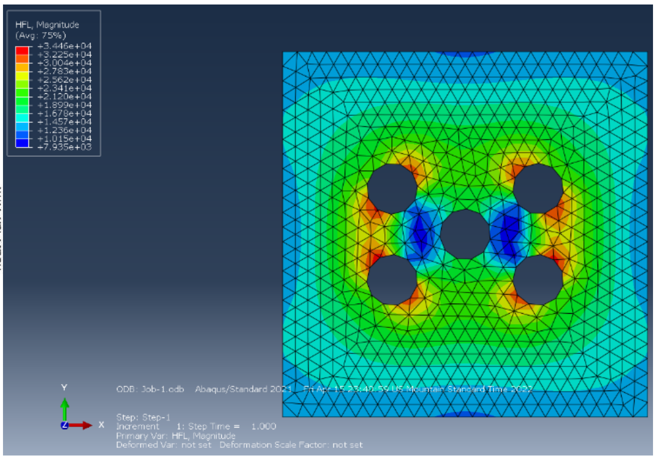
1. 3 node triangle H80



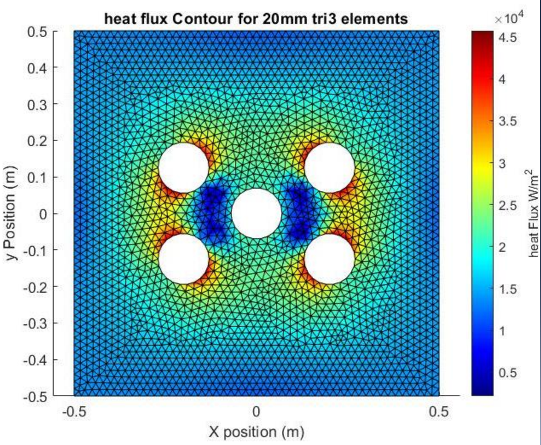


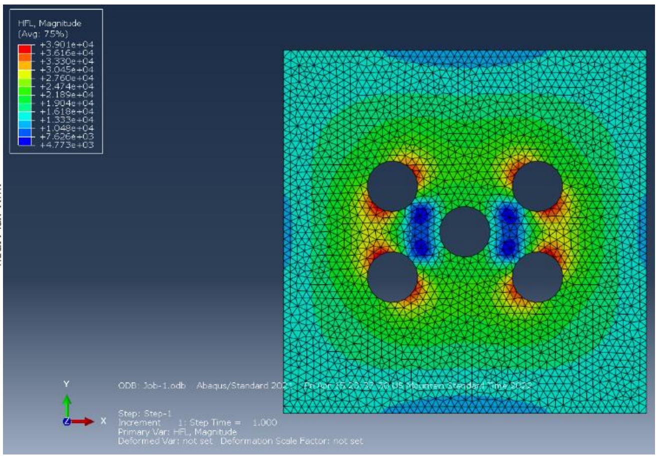
2. 3 node triangle H40



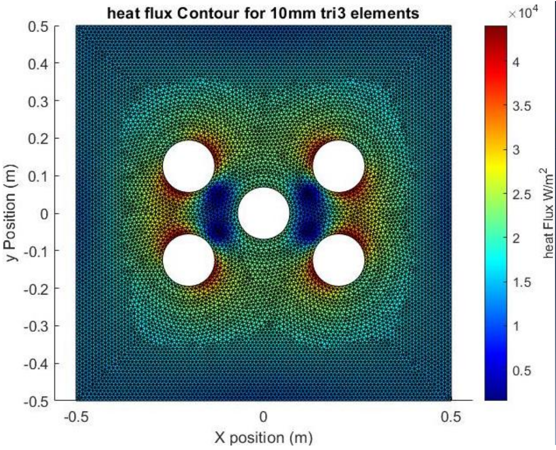


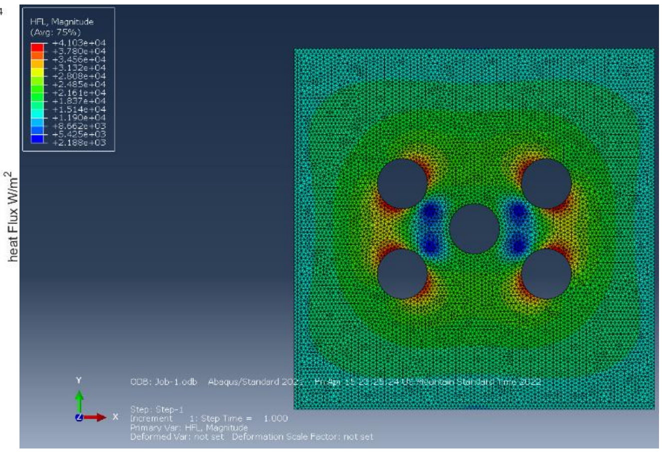
3. 3 node triangle H20



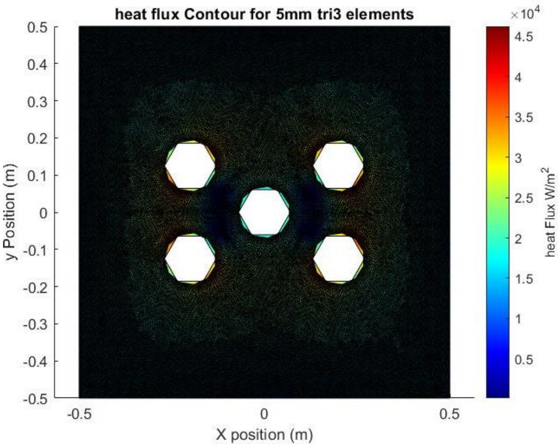


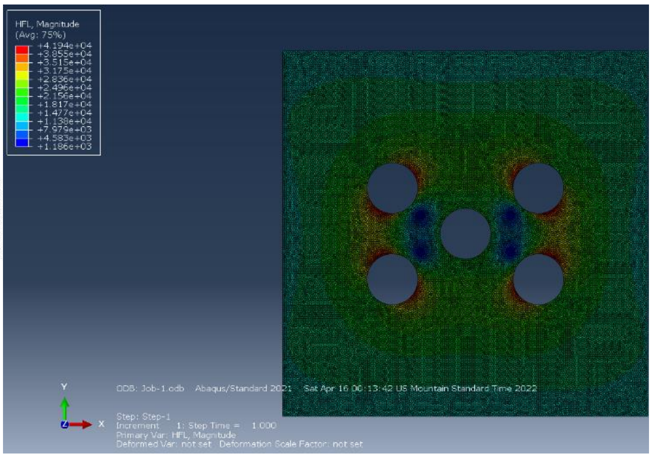
4. 3 node triangle H10



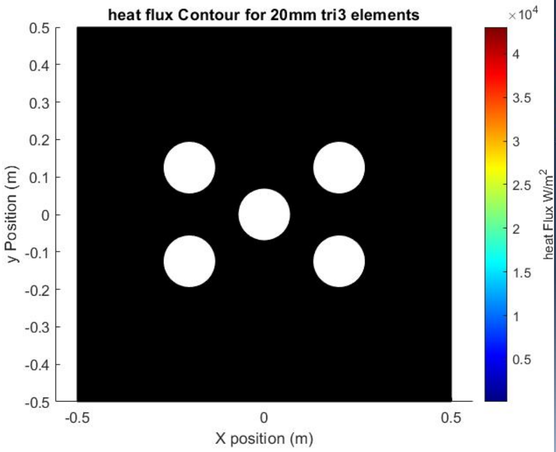


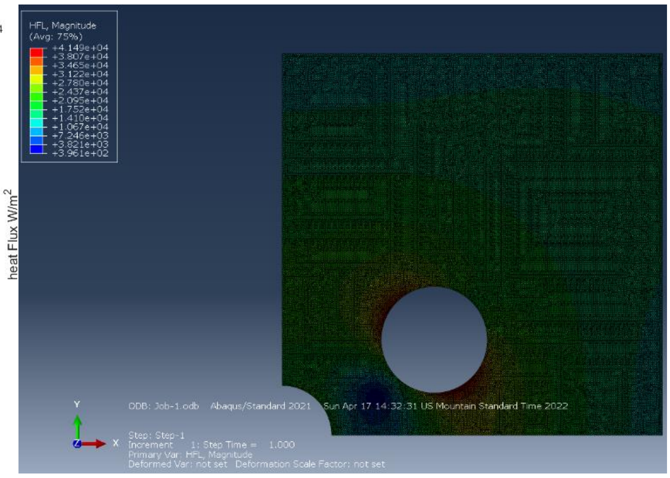
5. 3 node triangle H5





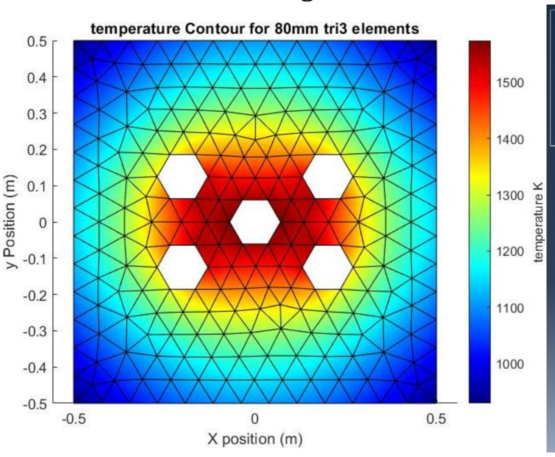
6. node triangle H2

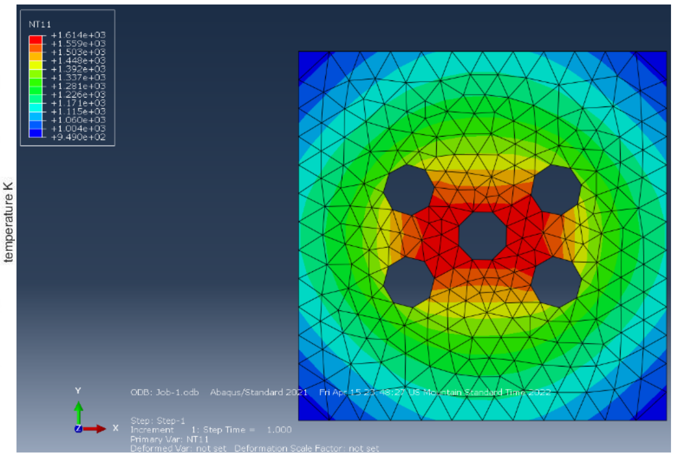




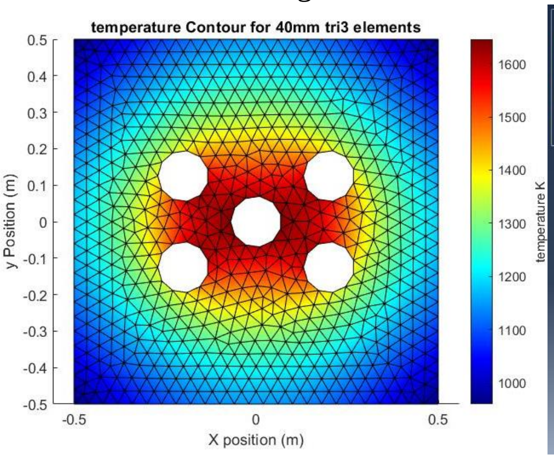
- 3 node Temperature Contours

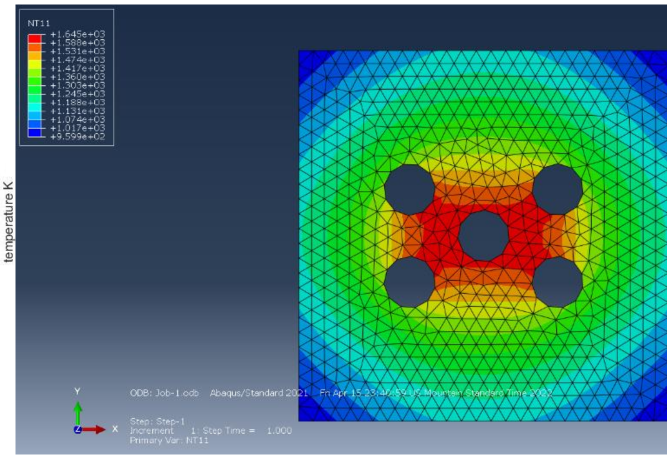
1. 3 node triangle H80



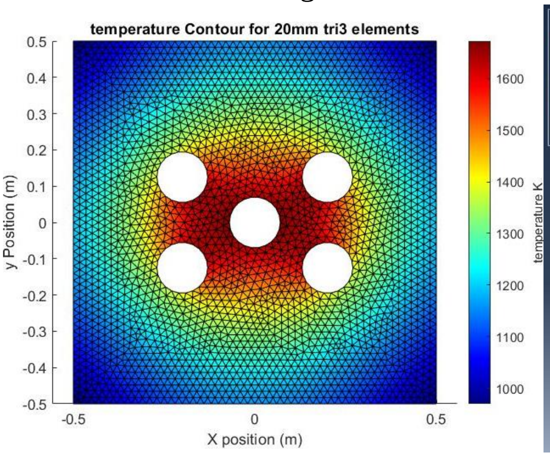


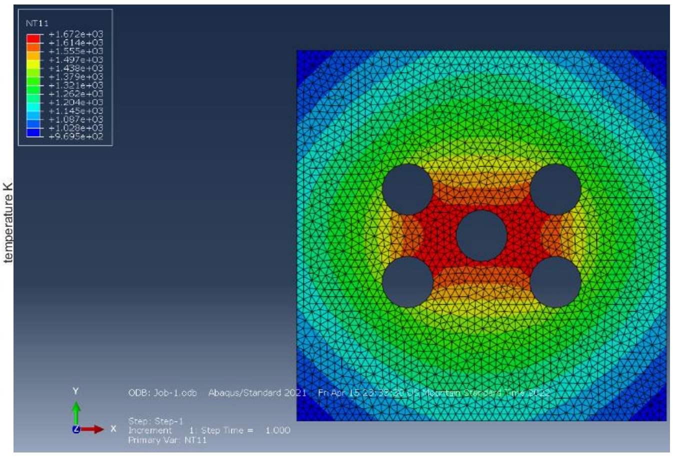
2. 3 node triangle H40



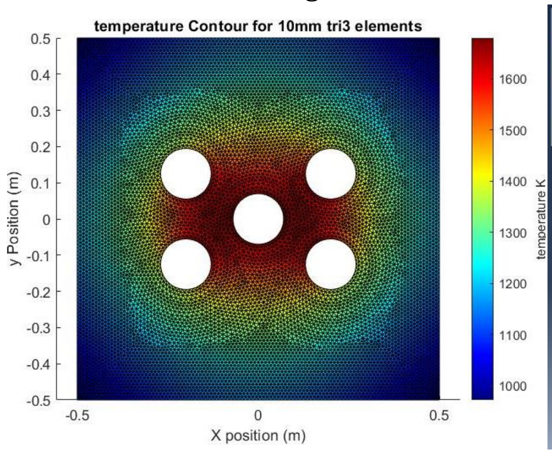


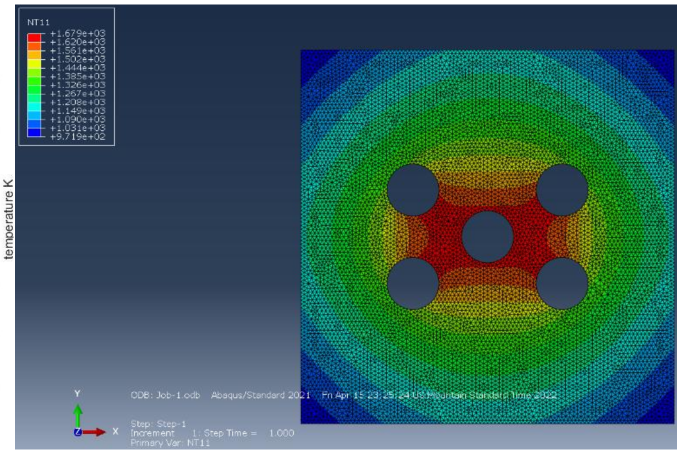
3. 3 node triangle H20



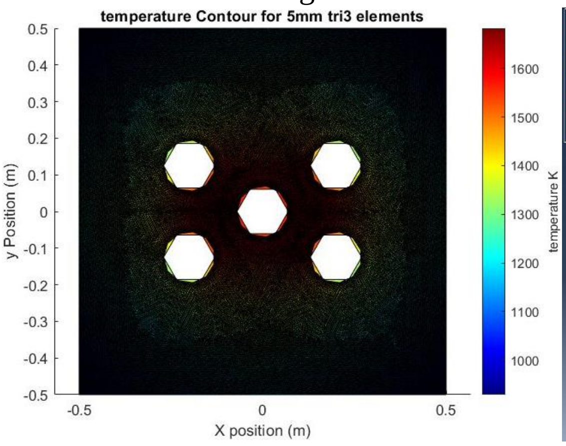


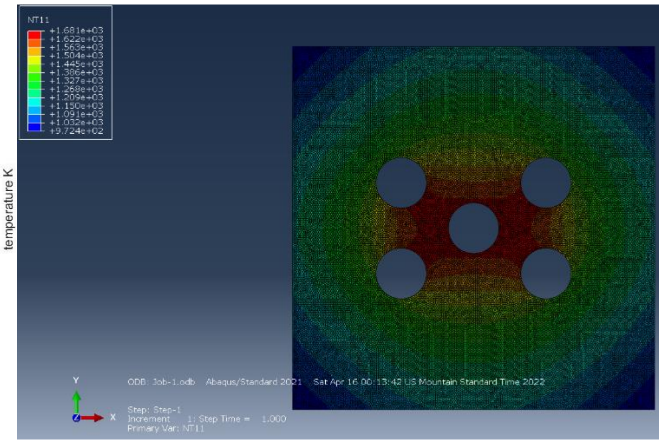
4. 3 node triangle H10



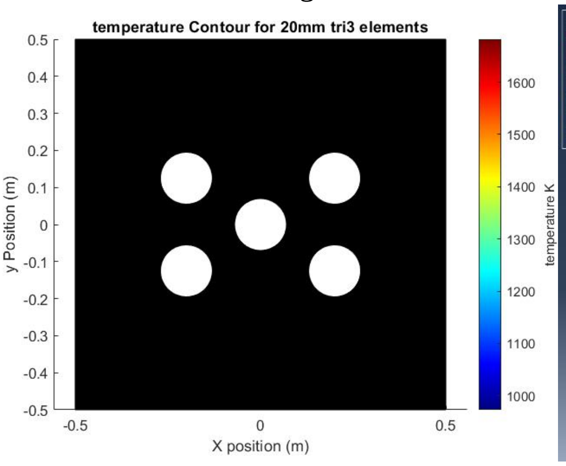


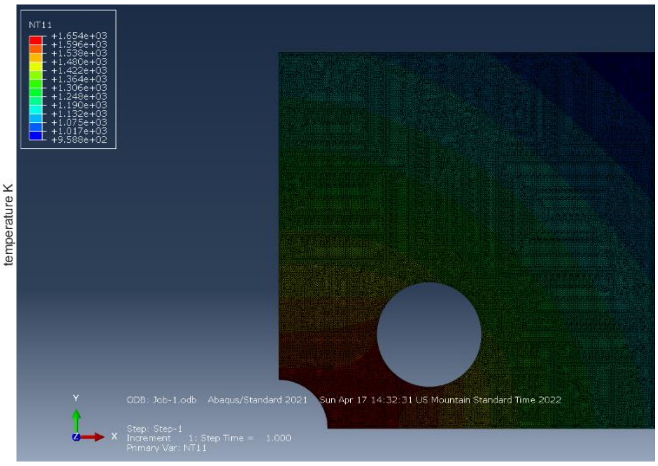
5. 3 node triangle H5





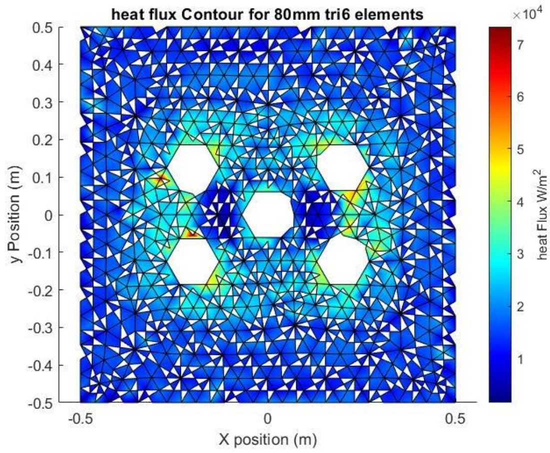
6. 3 node triangle H2

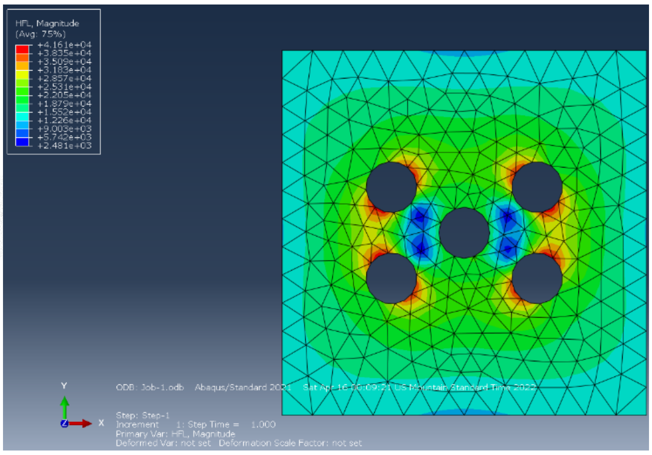




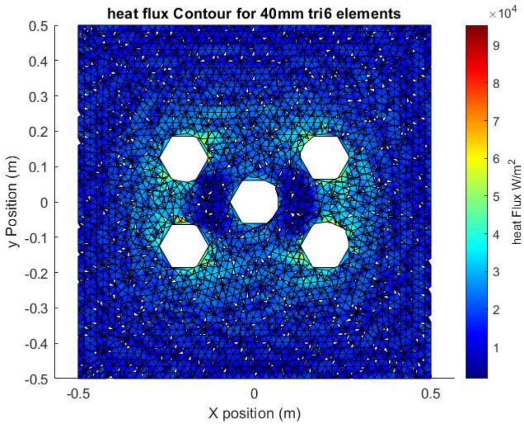
- 6 node triangle Heat Flux

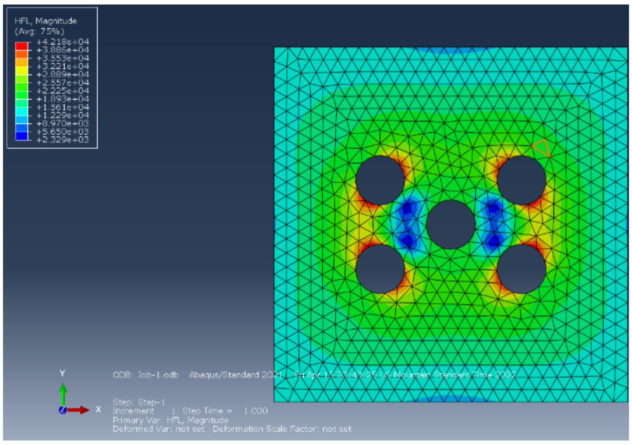
1. 6 node triangle H80



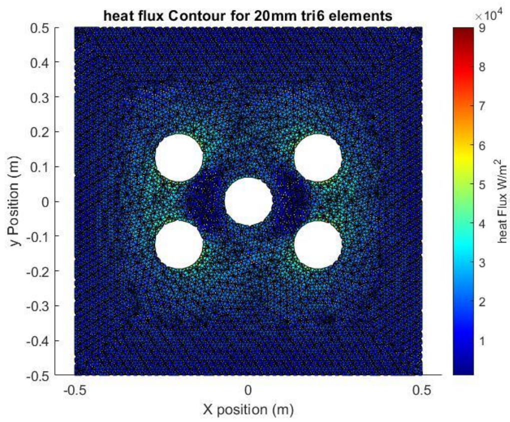


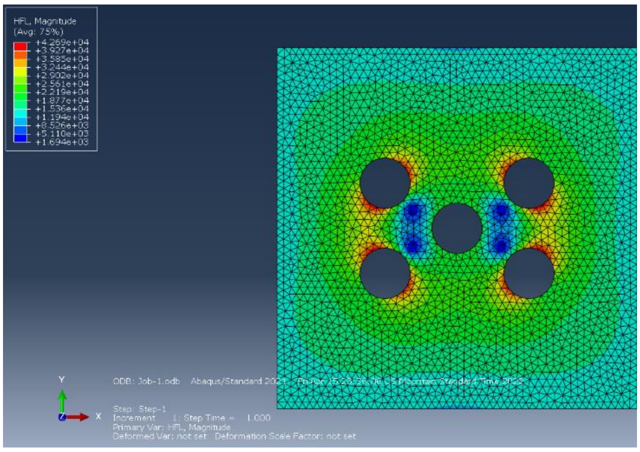
2. 6 node triangle H40



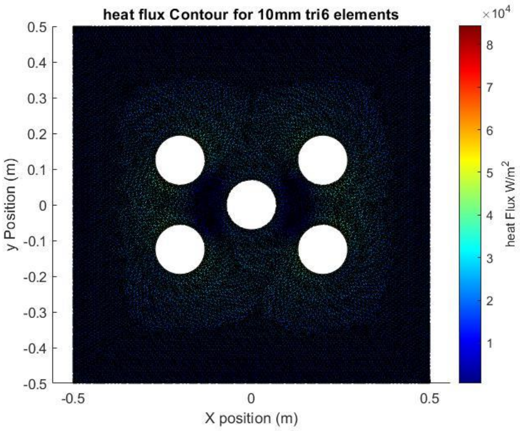


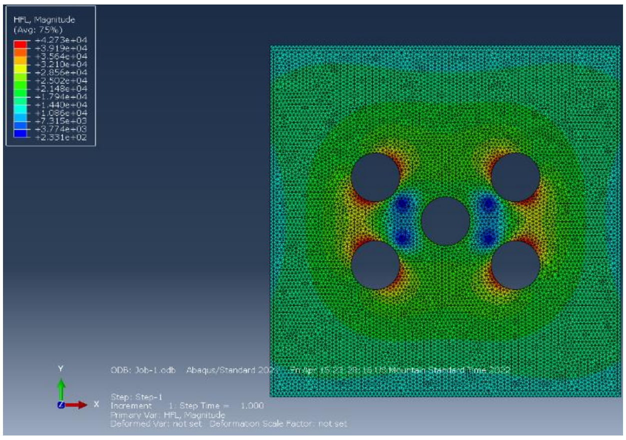
3. 6 node triangle H20



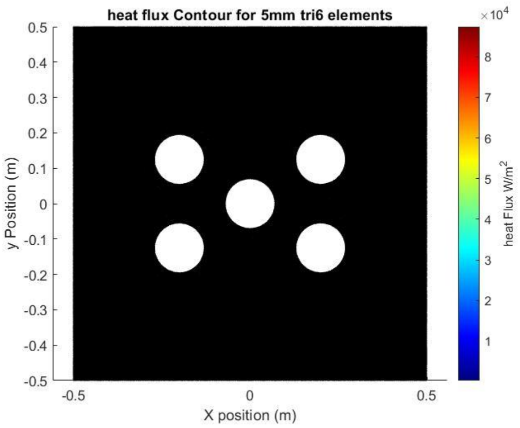


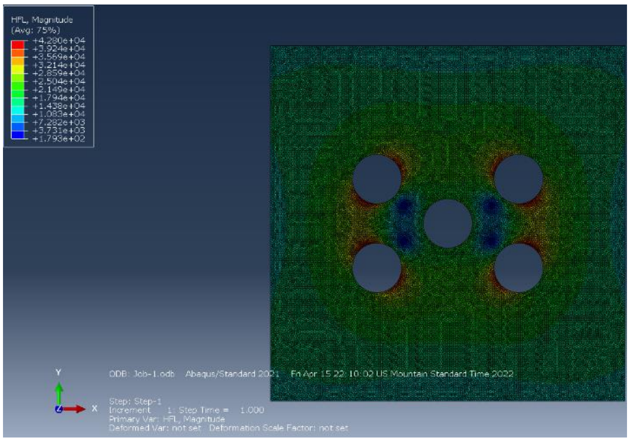
4. 6 node triangle H10



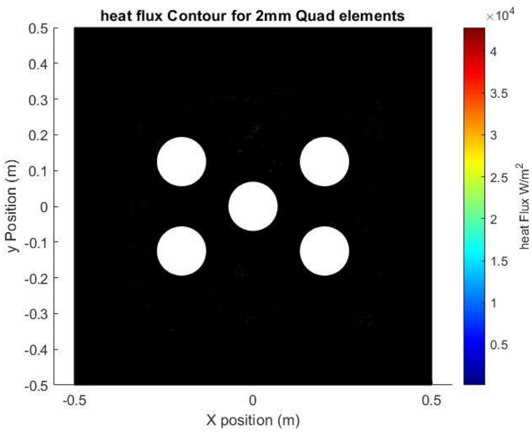


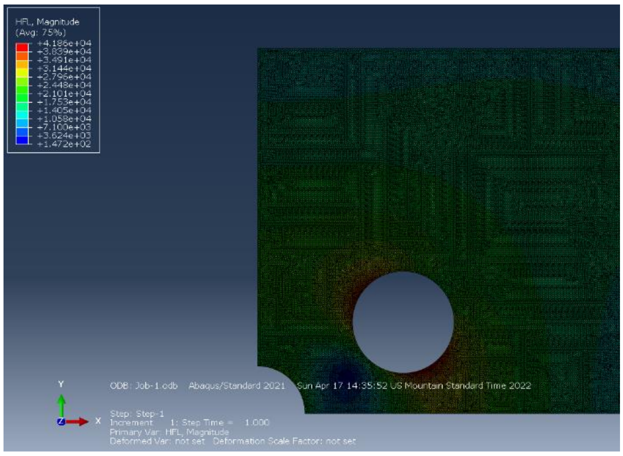
5. 6 node triangle H5





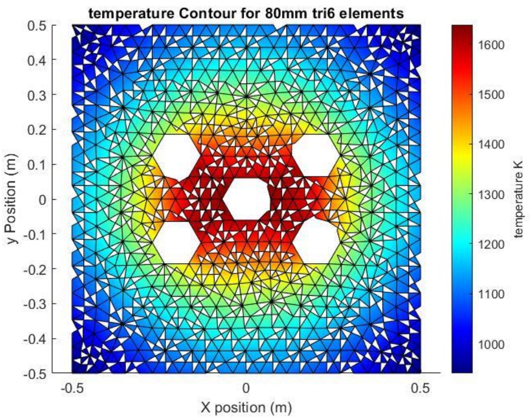
6. 6 node triangle H2

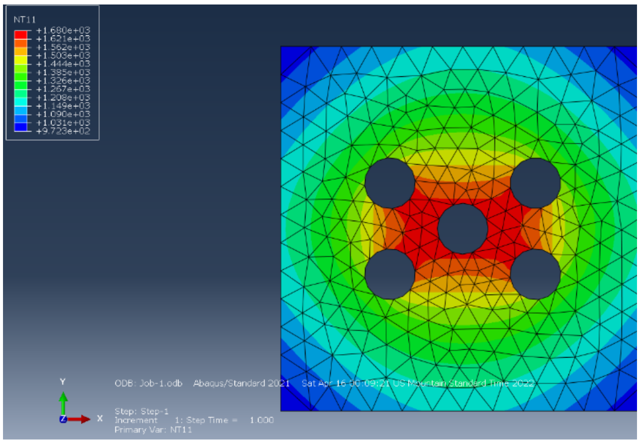




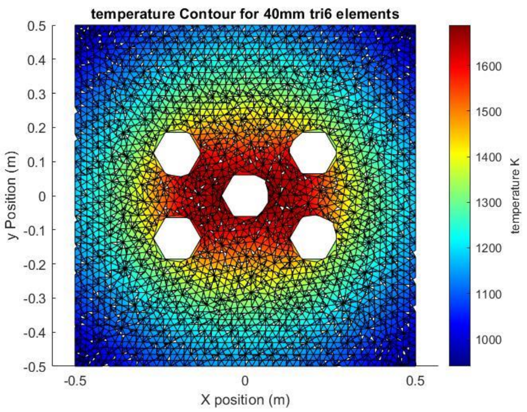
- 6 node triangle Temperature Contours

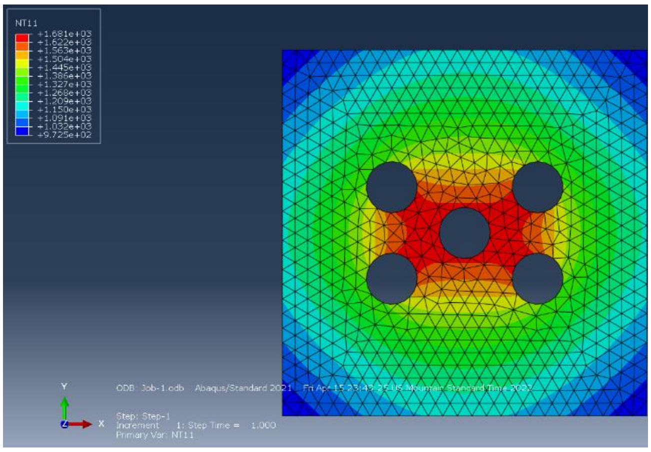
1. 6 node triangle H80



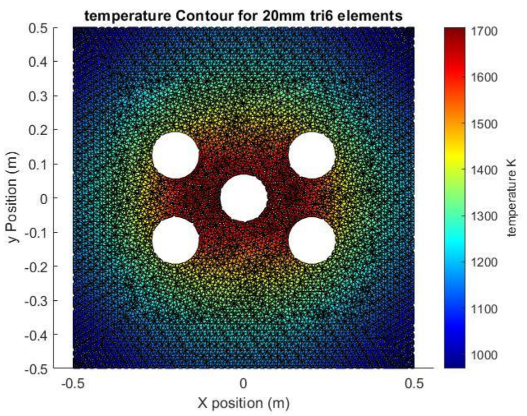


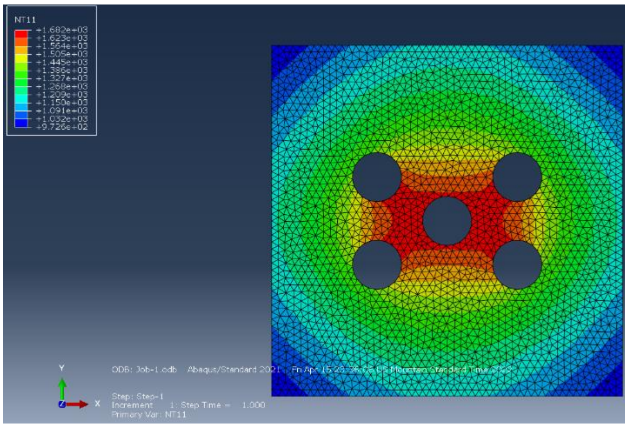
2. 6 node triangle H40



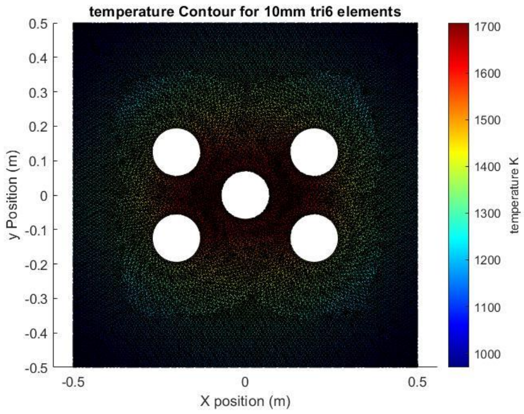


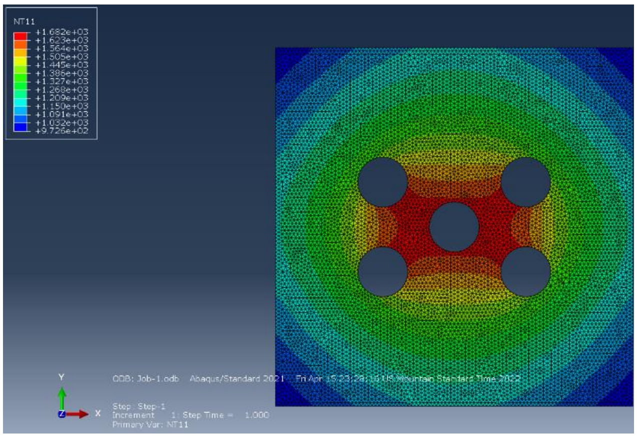
3. 6 node triangle H20



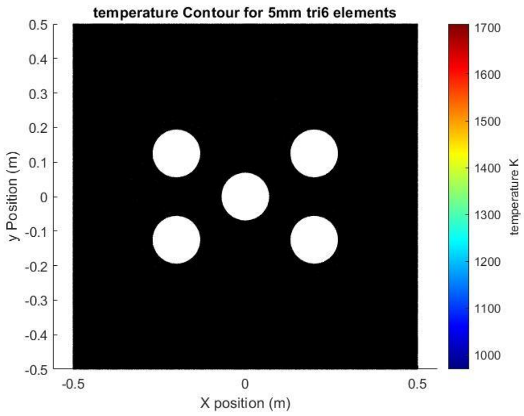


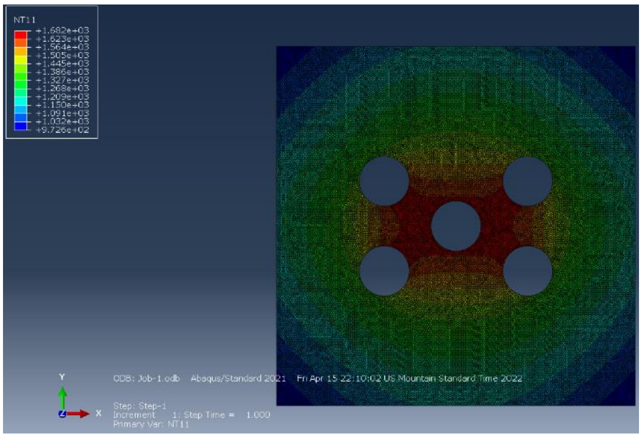
4. 6 node triangle H10



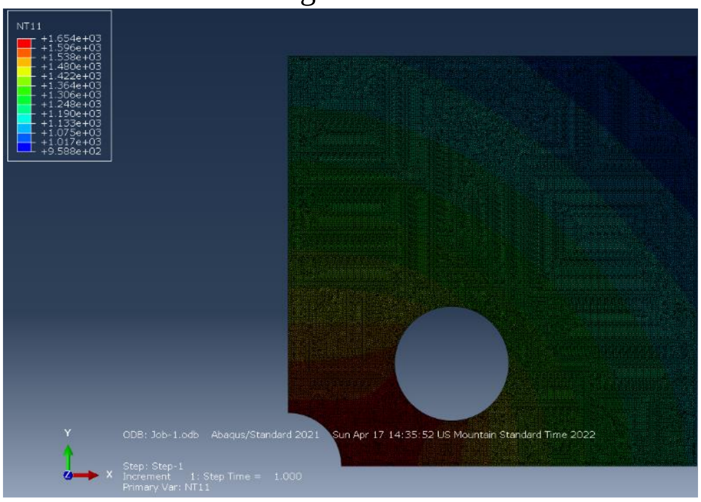


5. 6 node Triangle H5



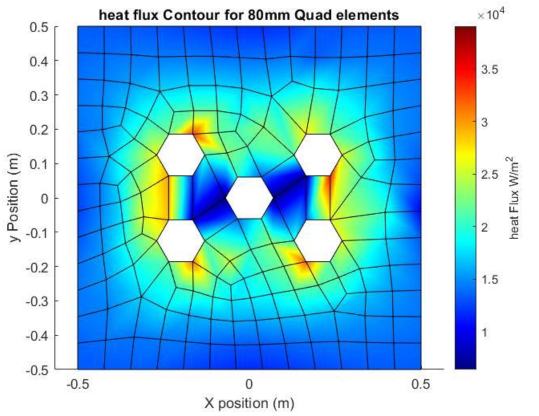


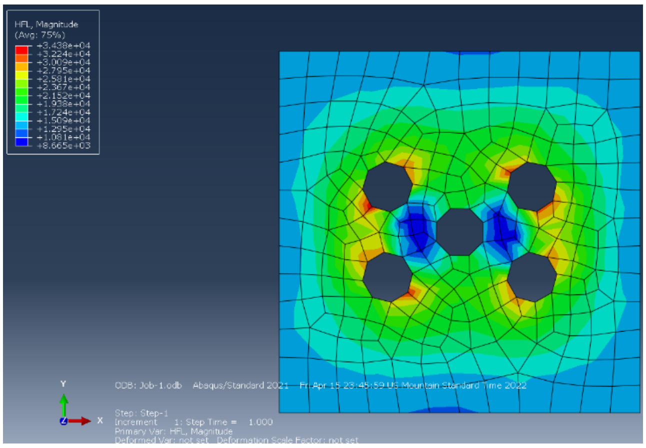
6. 6 node triangle H2



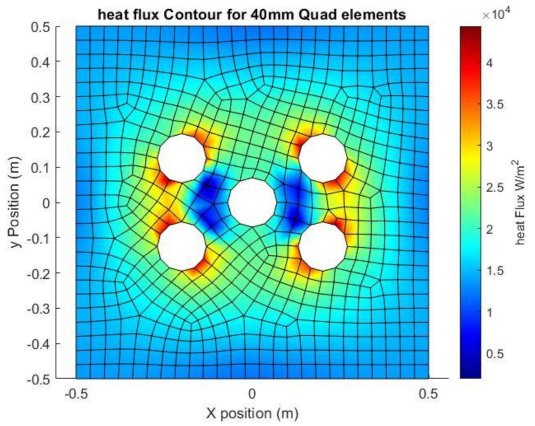
- 4 node Quadrilateral Heat Flux

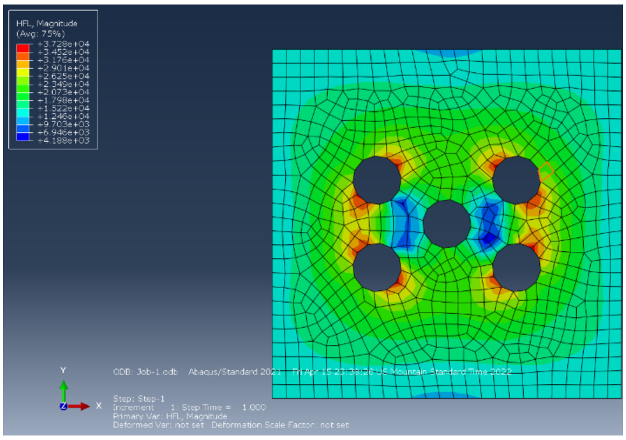
1. node quad H80



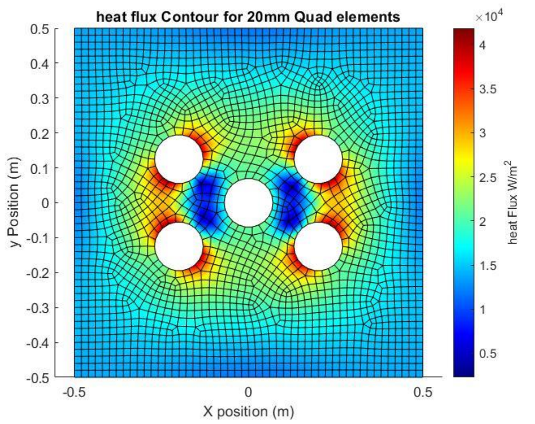


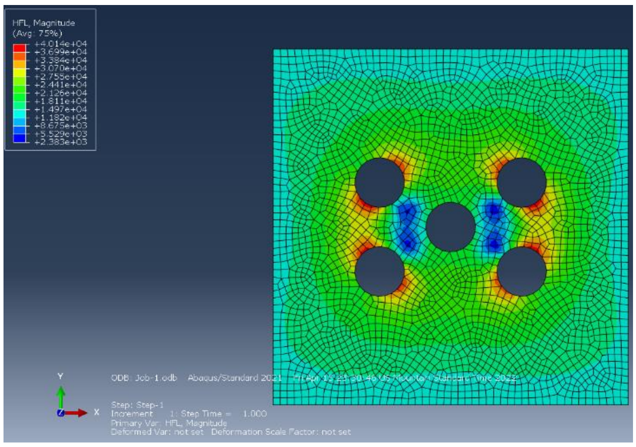
2. Quad 4 node H40



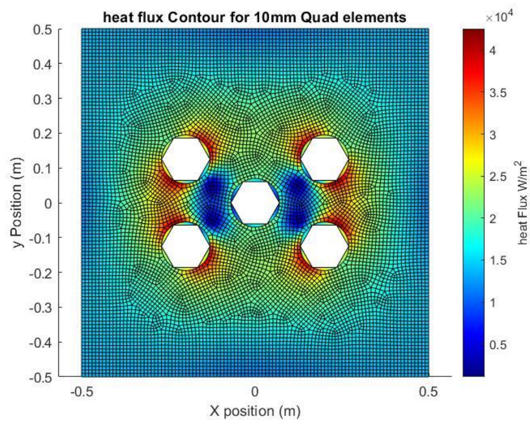


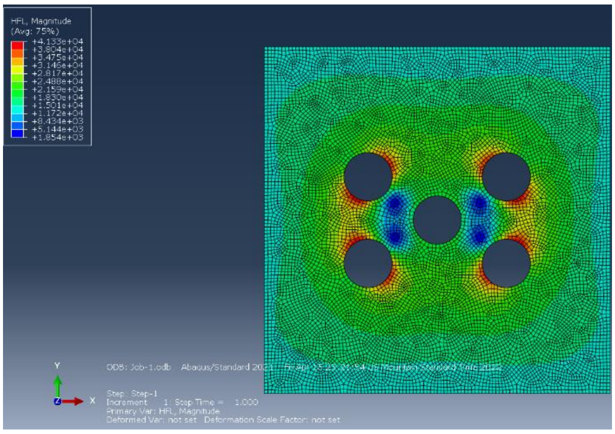
3. Quad 4 node H20



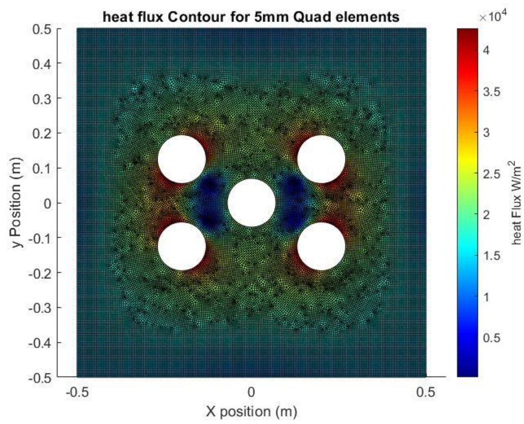


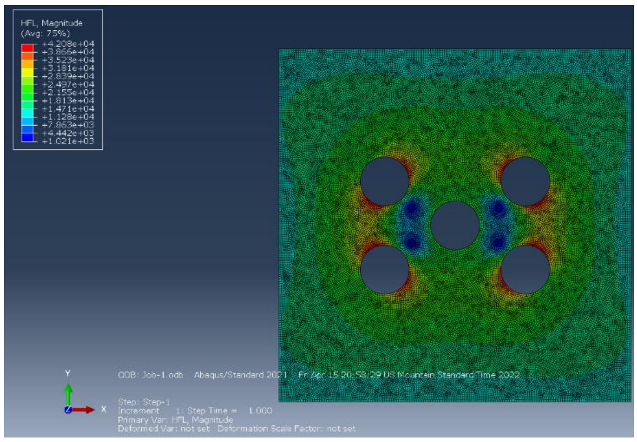
4. Quad 4 node H10



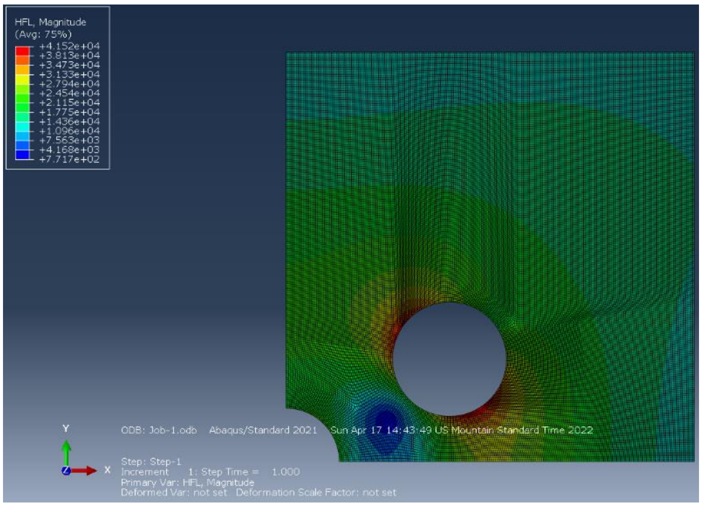


5. Quad 4 node H5



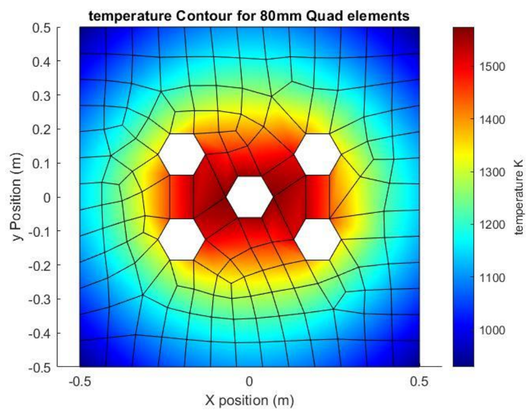


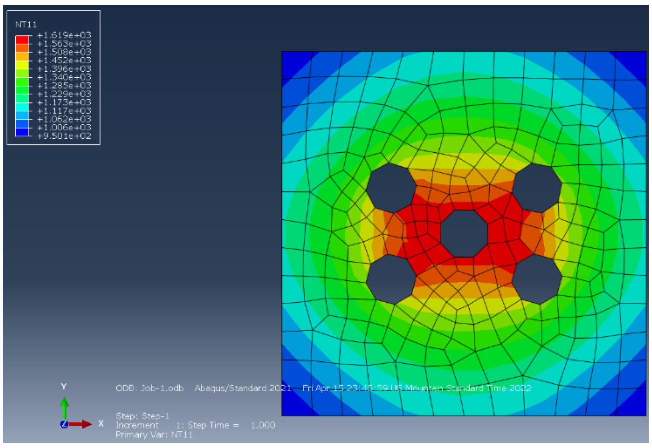
6. Quad 4 node H2



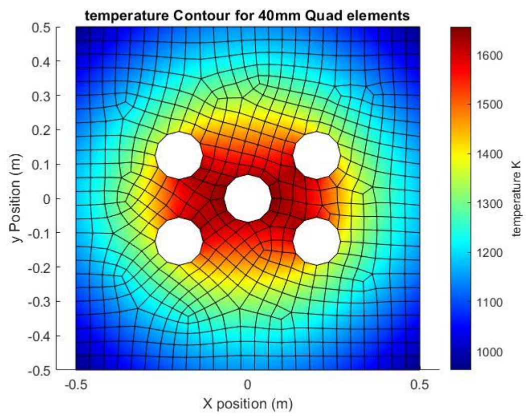
- 4 node quad temperatures

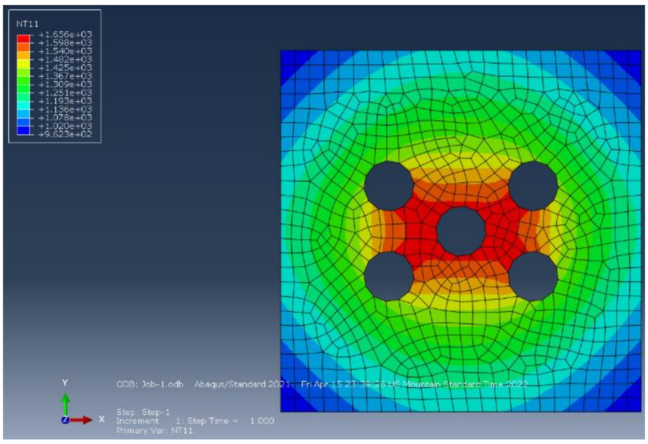
1. Quad 4 node H80



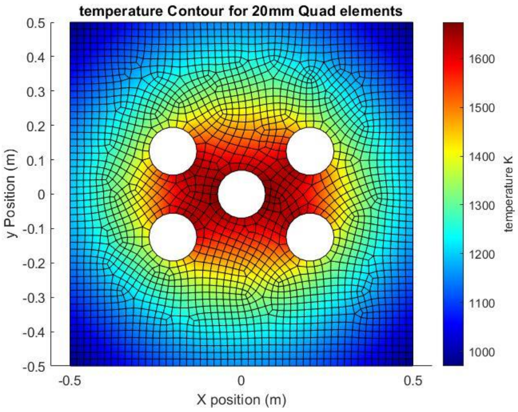


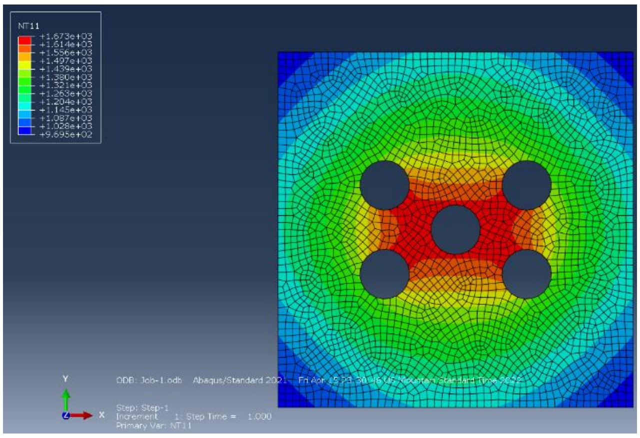
2. Quad 4 node H40



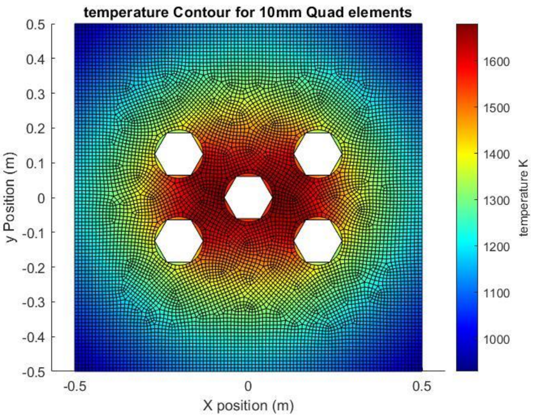


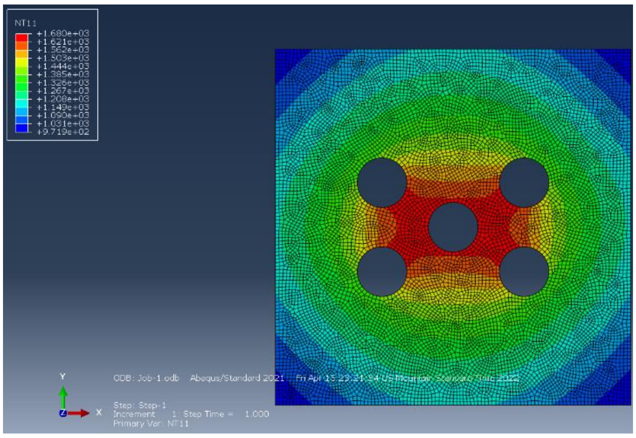
3. Quad 4 node H20



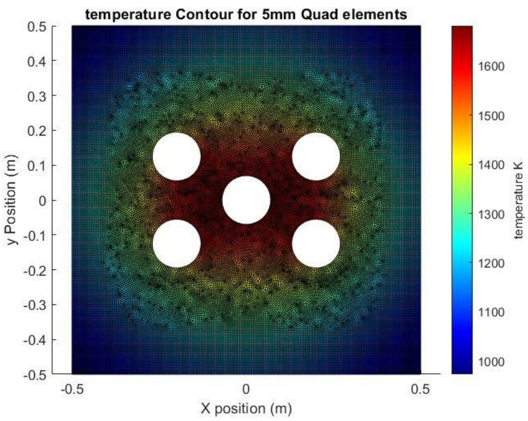


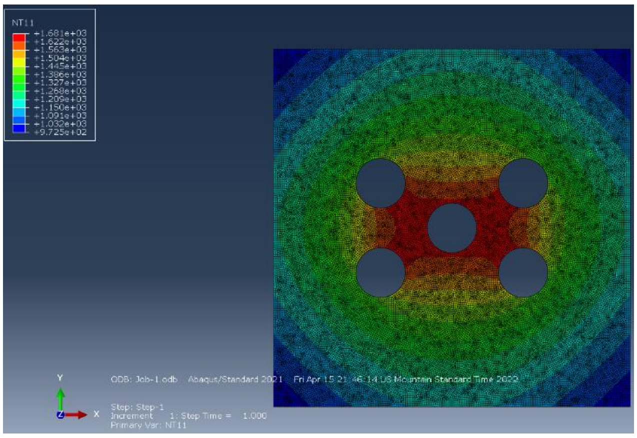
4. Quad 4 node H10



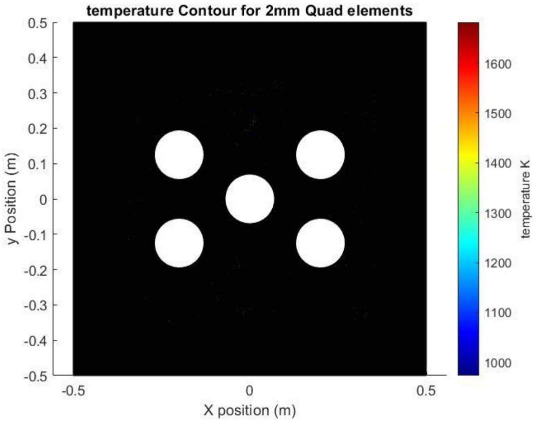


5. Quad 4 node H5





6. Quad 4 node H2



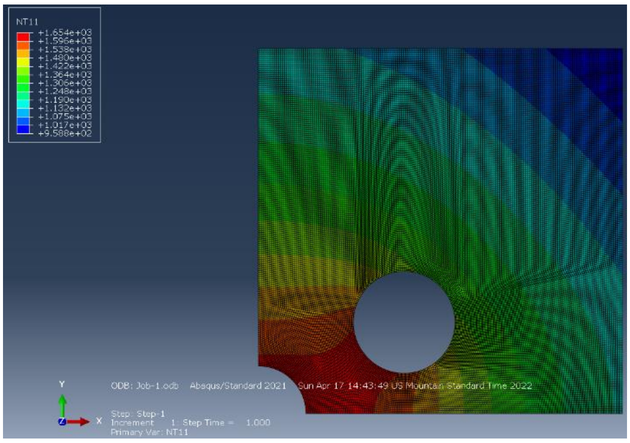


Table for Maximum Temperature and Heat flux for Different element sizes for 3 element types.

Element type	Element Size	Maximum temperature in MatLab	Maximum temperature in Abaqus	Maximum Heat Flux in MatLab	Maximum Heat Flux in ABAQUS
T3	2	1.6823E+03	1681	4.30E+04	41921
	5	1.6818E+03	1681	4.33E+04	41920
	10	1.6799E+03	1679	4.40E+04	40840
	20	1.6727E+03	1672	4.57E+04	38690
	40	1.6460E+03	1645	4.62E+04	34200
	80	1.5745E+03	1618	3.91E+04	30450
Q4	2	1.6823E+03	1681	4.28E+04	41942
	5	1.6819E+03	1681	4.26E+04	42140
	10	1.6804E+03	1680	4.25E+04	41390
	20	1.6745E+03	1674	4.18E+04	40040
	40	1.6563E+03	1655	4.43E+04	35960
	80	1.5710E+03	1626	3.44E+04	35510
T6	2	1.7064E+03	1682		42782
	5	1.7067E+03	1682	8.74E+04	42780
	10	1.7075E+03	1682	1.71E+03	42710
	20	1.7071E+03	1682	9.01E+04	42580
	40	1.6883E+03	1681	9.53E+04	42510
	80	1.5297E+04	1681	7.33E+04	41650

As we can see in the Table above, The finer the mesh the consistency of results increases for both Matlab and Abaqus for all element types. The coarse mesh i.e 40 and 80 shows significant difference in all T3, T6 and Q4 element types. Thus appropriate element sizes should be selected so that the exact solution will be converged. From observations the results of heat flux dosen’t seem to be converged. Difference is large as mesh gets coarse. But from observations, The finer the mesh the solution could get results as close to exact solutions.

Richardson Extrapolation Method:

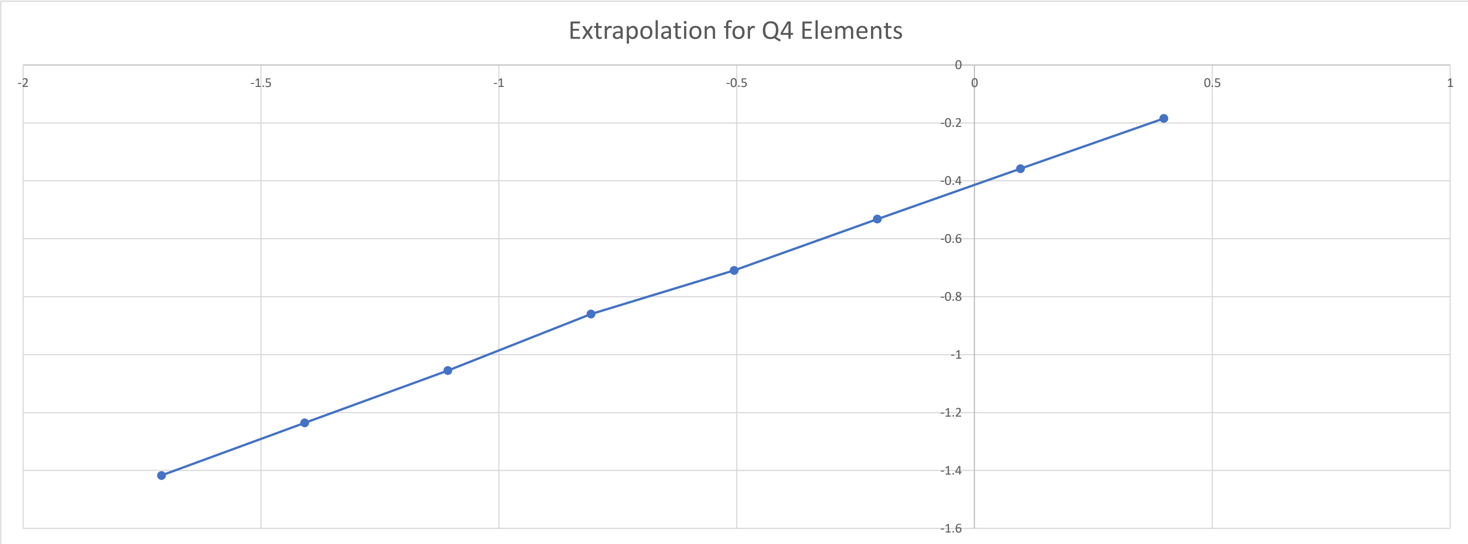
Element Type	Element Size	Max Temperature	T_diff	p value	Th0	Grid Convergence 12	Grid Convergence 34	range
T3	2	1.6823E+03	-5.3562E-01	1.7730E+00	1.6815E+03	1.64677E-02	3.9530E-02	9.99709E-01
	5	1.6818E+03	-1.8305E+00	1.9731E+00	1.6812E+03	4.65482E-02	1.1163E-01	0.999103
	10	1.6799E+03	-7.1867E+00	1.8970E+00	1.6809E+03	1.97116E-01	4.7080E-01	0.996745
	20	1.6727E+03	-2.6767E+01	1.4164E+00	1.6818E+03	1.21775E+00	2.8604E+00	0.992738
	40	1.6460E+03	-7.1447E+01					
	80	1.5745E+03						

Element Type	Element Size	Max Temperature	T_diff	p value	fh0	Grid Convergence 12	Grid Convergence 34	range
Q4	2	1.6823E+03	-4.0000E-01	1.9069E+00	1.6824E+03	1.08103E-02	2.5936E-02	0.999643
	5	1.6819E+03	-1.5000E+00	1.9758E+00	1.6824E+03	3.80388E-02	9.1184E-02	0.998874
	10	1.6804E+03	-5.9000E+00	1.6252E+00	1.6832E+03	2.11263E-01	5.0440E-01	0.996182
	20	1.6745E+03	-1.8200E+01	2.2286E+00	1.6794E+03	3.72556E-01	8.8182E-01	0.983291
	40	1.6563E+03	-8.5300E+01					
	80	1.5710E+03						

For Q4 elements:

Size	Max_T	Delta	log-size	log-delta
0.01953125	0.0382727	-1.709269961	-1.417110899	
0.0390625	0.0581121	-1.408239965	-1.23573343	
0.078125	0.0881025	-1.10720997	-1.055011768	
0.15625	0.1381045	-0.806179974	-0.85979217	
0.3125	0.1953916	-0.505149978	-0.709094111	
0.625	0.2936892	-0.204119983	-0.532112024	
1.25	0.4386035	0.096910013	-0.357927907	
2.5	0.65349	0.397940009	-0.184761054	

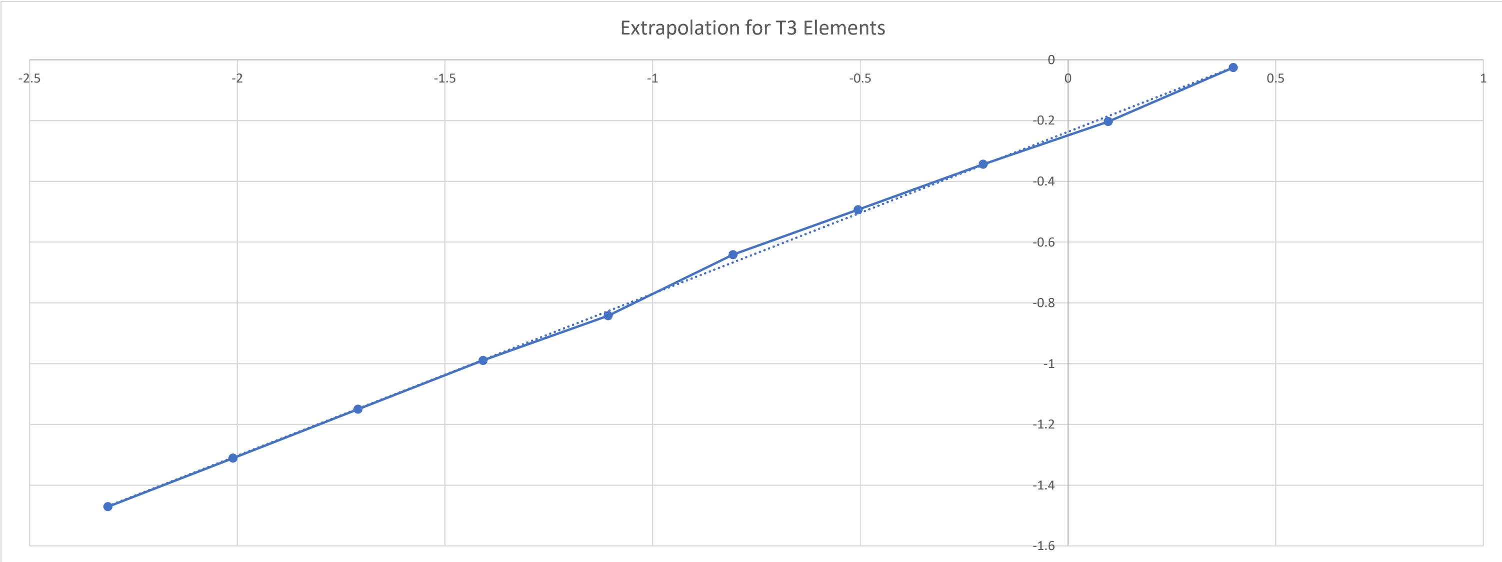
Element Size	Max Temperature	T_diff	Log Size	Log diff
2	1.6823E+03	-4.0000E-01	0.301029996	-0.397940009
5	1.6819E+03	-1.5000E+00	0.698970004	0.176091259
10	1.6804E+03	-5.9000E+00	1	0.770852012
20	1.6745E+03	-1.8200E+01	1.301029996	1.260071388
40	1.6563E+03	-8.5300E+01	1.602059991	1.930949031
80	1.5710E+03		1.903089987	



For T3 elements:

Size	Max_T	Delta	Log-size	Log_delta
0.004882813		0.03382974	-2.311329952	-1.47070134
0.009765625		0.048872488	-2.010299957	-1.310935551
0.01953125		0.070803626	-1.709269961	-1.149944501
0.0390625		0.102522536	-1.408239965	-0.989180661
0.078125		0.14373929	-1.10720997	-0.842424505
0.15625		0.228394828	-0.806179974	-0.641313735
0.3125		0.321043879	-0.505149978	-0.493435606
0.625		0.45298228	-0.204119983	-0.343918787
1.25		0.62599234	0.096910013	-0.203430981
2.5		0.942763624	0.397940009	-0.025597183

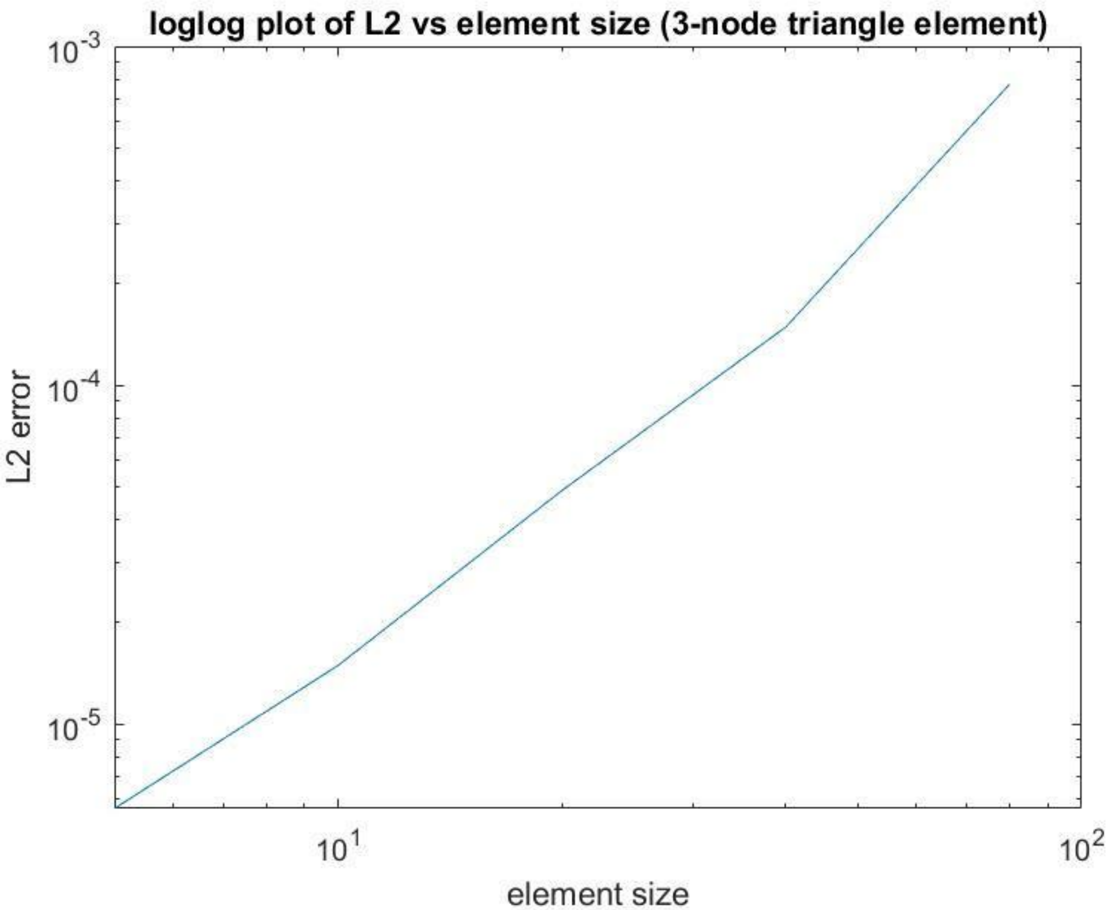
Size	Max_temp	Diff	Log-size	Log diff
2	1.6823E+03	-5.3562E-01	3.225903445	-0.271141719
5	1.6818E+03	-1.8305E+00	3.225765149	0.262569635
10	1.6799E+03	-7.1867E+00	3.225292188	0.856527283
20	1.6727E+03	-2.6767E+01	3.223430316	1.427593372
40	1.6460E+03	-7.1447E+01	3.216424714	1.853983137
80	1.5745E+03		3.197151973	


Extrapolation for T3 Elements
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6

Plot for L2 Error Norm and convergence-

The Rate of convergence obtained for 3-Node Trai Element is **1.901638**.

```
Read 234 nodes and 394 elements from C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-mesh-h80.inp.
Read 732 nodes and 1317 elements from C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-mesh-h40.inp.
Read 2703 nodes and 5104 elements from C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-mesh-h20.inp.
Read 10013 nodes and 19414 elements from C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-mesh-h10.inp.
Read 39433 nodes and 77634 elements from C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-mesh-h5.inp.
Rate of convergence for 3-node elements= 1.901638e+00>>
```



```
clc;
clear;
el_size = [80,40,20,10,5];
max_mat = [];
for i= 1:length(el_size)
    h = el_size(i);
    if h == 80
        mesh=abaqus_reader('C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-
mesh-h80.inp');
    elseif h == 40
        mesh=abaqus_reader('C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-
mesh-h40.inp');
    elseif h == 20
        mesh=abaqus_reader('C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-
mesh-h20.inp');
    elseif h == 10
        mesh=abaqus_reader('C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-
mesh-h10.inp');
    elseif h == 5
        mesh=abaqus_reader('C:\Users\ochavan2\Downloads\meshes\T3-meshes\t3-
mesh-h5.inp');
    end
    m = Mt_Mq(mesh);
    max_mat = [max_mat;m];
end

function m = Mt_Mq(mesh)

kappa = 17; % W/mk

h = 10; % W/m2-K

q_bar = 20000; % W/m

T_ambient = 0; % Kelvin

F = zeros(length(mesh.x),1);

K = spalloc(length(mesh.x),length(mesh.x),9*length(mesh.x));

quad=[1/3;1/3;1/2];

    for c = mesh.conn
      Ke = zeros(length(c));
      xe = mesh.x(:,c);
      fe=zeros(length(c),1);
      for q = quad
          [N, dNdp] = shape_tri3(q);
          J = xe*dNdp;
          B = dNdp/J;
          Ke = Ke+ B*kappa*B'*det(J)*q(end);
          fe=fe+N*kappa*det(J)*q(end);
```

```
      end
      K(c,c) = K(c,c) + Ke;
      F(c)=F(c)+fe;
   end

perimeter_edge_conn = mesh.edge_connectivity('perimeter');
hole_edge_conn = mesh.edge_connectivity('holes');

for ch = hole_edge_conn
    xh = mesh.x(:,ch);
    he = zeros(length(ch),1);
    for g = quadrature(2)
        Ne = 0.5*[1-g(1); 1+g(1)];
        dNdpe = [-0.5; 0.5];
        J2 = xh * dNdpe;
        he = he + Ne*q_bar*norm(J2)*g(end);

    end
    F(ch)=F(ch)+he;
end

for cp = perimeter_edge_conn
    xp = mesh.x(:,cp);
    fg = zeros(length(cp),1);
    for u = [0;2]
        Np = 0.5*[1-u(1); 1+u(1)];
        dNdpp = [-0.5; 0.5];
        J3 = xp * dNdpp;
        fg = fg + Np*h*Np'*norm(J3)*u(end);
    end
    K(cp,cp)=K(cp,cp)+fg;
end

T=K\F;
max_temperature = max(T);

% Heat flux
A = spalloc(length(mesh.x),length(mesh.x),9*length(mesh.x));
y = zeros(length(mesh.x),2);
for c = mesh.conn
    xe3 = mesh.x(:,c);
    de = T(c)';
    Ae = zeros(length(c));
    for q = quad
        [Nq,dNdpq] = shape_tri3(q);
        J3 = xe3*dNdpq;
        dNdxq = dNdpq/J3;
        qflux = -kappa*de*dNdxq;
        Ae = Ae + Nq*Nq'*det(J3)*q(end);
        y(c,:) = y(c,:) + Nq*qflux*det(J3)*q(end);
    end
    A(c,c) = A(c,c) + Ae;
end
qflux = A\y;
```

```
for c = 1:length(qflux)
    qfluxMag(c) = sqrt(qflux(c,1)^2+qflux(c,2)^2);
end
maxFlux = max(qfluxMag);

m = [max_temperature;maxFlux];

% Plotting Contour Plots

% figure(1)
% p.faces = mesh.conn';
% p.vertices = mesh.x';
% p.facecolor = 'interp';
% p.facevertexcdata = T;
% patch(p)
% c1=colorbar;
% colormap(jet(256))
% c1.Label.String='temperature K';
% xlabel('X position (m)');
% ylabel('y Position (m)');
% title('temperature Contour for 3-node triangular elements');
% axis equal;
%
% figure(2);
% p.faces = mesh.conn';
% p.vertices = mesh.x';
% p.facecolor = 'interp';
% p.facevertexcdata = qfluxMag';
% patch(p)
% c2=colorbar;
% colormap(jet(256))
% c2.Label.String='heat Flux W/m^2';
% xlabel('X position (m)');
% ylabel('y Position (m)');
% title('heat flux Contour for 3 node triangular elements');
% axis equal;
end

function [N,dNdp] = shape2(p)
    N   = (1/2)*[1-p;1+p];

    dNdp= [-0.5;0.5];

end
function[N,dNdp]=shape_tri3(p)
    N = [p(1);p(2);1-p(1)-p(2)];
    dNdp =[1, 0; 0, 1; -1, -1];
end
%
% function[N,dNdp]=shape_tri6(p)
%
%     N = [p(1)*(2*p(1)-1); p(2)*(2*p(2)-1); (1-p(1)-p(2))*(2*(1-p(1)-
p(2))-1); 4*p(1)*p(2); 4*p(2)*(1-p(1)-p(2)); 4*p(1)*(1-p(1)-p(2))];
%
```

```
%     dNdp = [4*p(1) - 1,0; 0,4*p(2) - 1; 4*p(1)-3+4*p(2),-3+4*p(1)+4*p(2);
 4*p(2), 4*p(1); -4*p(2), 4-4*p(1)-8*p(2);  4-8*p(1)-4*p(2), -4*p(1)];
%
% end
```

```
Read 234 nodes and 394 elements from C:\Users\ochavan2\Downloads\meshes\T3-
meshes\t3-mesh-h80.inp.
Read 732 nodes and 1317 elements from C:\Users\ochavan2\Downloads\meshes\T3-
meshes\t3-mesh-h40.inp.
Read 2703 nodes and 5104 elements from C:\Users\ochavan2\Downloads\meshes\T3-
meshes\t3-mesh-h20.inp.
Read 10013 nodes and 19414 elements from C:\Users\ochavan2\Downloads\meshes
\T3-meshes\t3-mesh-h10.inp.
Read 39433 nodes and 77634 elements from C:\Users\ochavan2\Downloads\meshes
\T3-meshes\t3-mesh-h5.inp.
```

Published with MATLAB® R2021b