MAE 404 Project 2

Sherwin Varghese Mathew, Surya Shekhar

TOTAL POINTS

110 / 110

QUESTION 1

1 Problem statement 5 / 5

✓ - **0 pts** Correct

- **2 pts** problem statement has missing, incomplete, or incorrect boundary conditions

QUESTION 2

2 Representative solution 15 / 15

✓ - **0 pts** **Both temperature and flux appear accurate**

- **0 pts** temperature not localized in center of fin (probable unit conversion error)
- **5 pts** temperature field incorrect
- **5 pts** heat flux incorrect
- **15 pts** plots copied from another group
- **2 pts** missing temperature and/or heat flux units
- **3 pts** Flux projection for QUAD8 incorrect

QUESTION 3

3 Convergence test (MAE 598 only) 30 / 30

✓ - **0 pts** Correct

- **5 pts** report does not state what the manufactured solution is
- **10 pts** convergence of Q4 elements is incorrect
- **5 pts** convergence of Q8 elements is incorrect
- **10 pts** missing computed rates of convergence
- **30 pts** all group members enrolled in MAE 404

QUESTION 4

4 Mesh refinement test 20 / 20

✓ - **0 pts** Correct

- **5 pts** missing estimate of error for each mesh
- **5 pts** does not show convergence in Tmax and/or qmax

QUESTION 5

5 Comparison to commercial FEM code 20 / 20

✓ - **0 pts** Correct

- **5 pts** missing comparison of heat flux
- **10 pts** abaqus heat flux not prescribed correctly
- **3 pts** big discrepancy in heat flux
- **20 pts** missing
- **5 pts** not showing matlab solution in comparison

QUESTION 6

6 Optimal design parameters 15 / 15

✓ - **0 pts** Correct

- **2 pts** did not explore sensitivity of beta parameter
- **3 pts** insufficient description of procedure used
- **15 pts** missing

QUESTION 7

7 Appendix - code 5 / 5

✓ - **0 pts** Correct

- **5 pts** no appendix

MAE 598: Finite Elements in Engineering

Spring 2019 Project 2

By

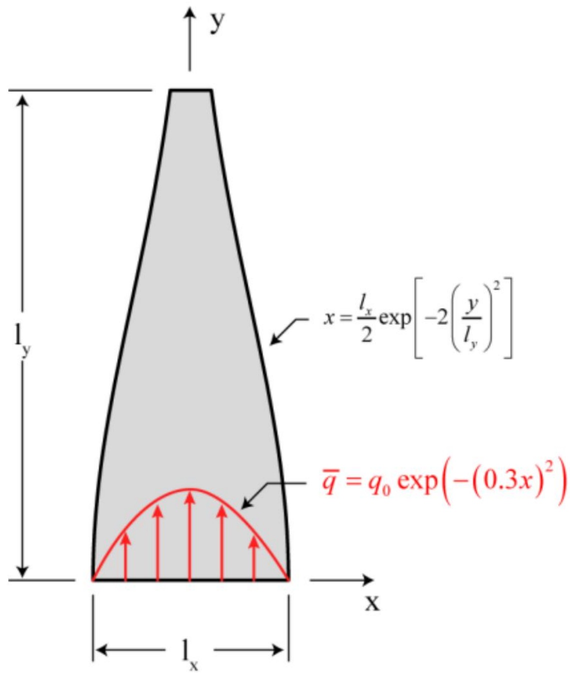
Sherwin Varghese Mathew
(ASU ID # 1214299540)

&

Surya Shekhar
(ASU ID # 1215338475)

Index

1. Problem Statement



Geometry of cooling fin.

The problem statement given is to design a fin cross section that could be manufactured from minimum material and can give maximum heat dissipation. The design parameters will be the height of the fin ‘Ly’ and ‘β’ that determines the edge profile.

The given parameters are:

1. A surface heat flux entering the body from the bottom surface with a function of
$$\overline{q} = q_0 \exp(-x^2)$$
Where: q_0=225 kW/m^2
2. Amount of heat that the film should dissipate from the edges per sq unit of area. surface film coefficient = .0001 W/ mm^2-K.
3. The ambient temperature = 0°C = 273 K
4. Thermal Conductivity of the material = .045 W/mm-k
5. The equation of the edge profile is
$$x = \left(\frac{Lx}{2}\right) * [\exp\{-2 * (y/Ly)\text{^}\beta\}]$$
6. Base length of the fin Lx=10mm

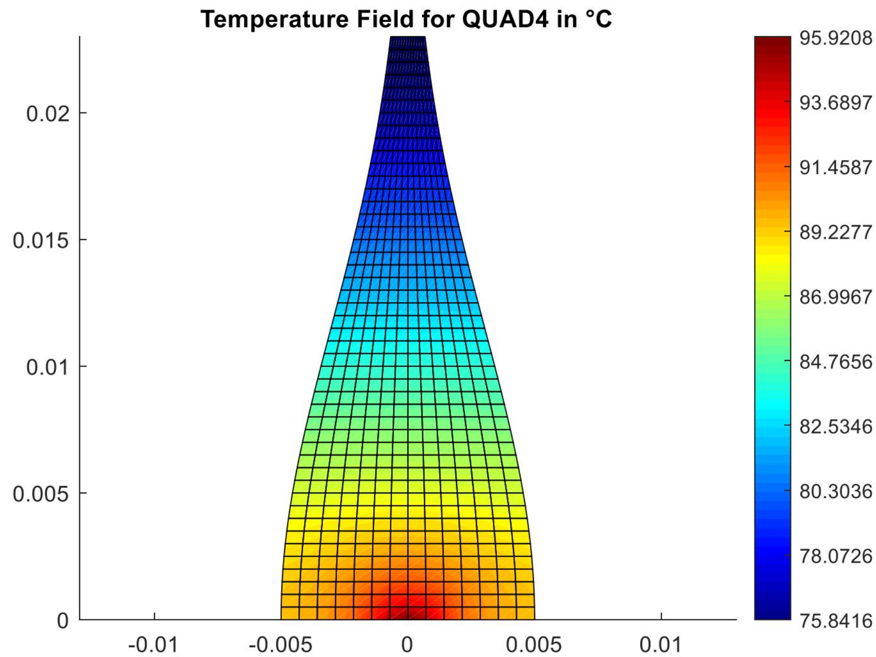
1 Problem statement **5 / 5**

✓ **- 0 pts** **Correct**

- 2 pts problem statement has missing, incomplete, or incorrect boundary conditions

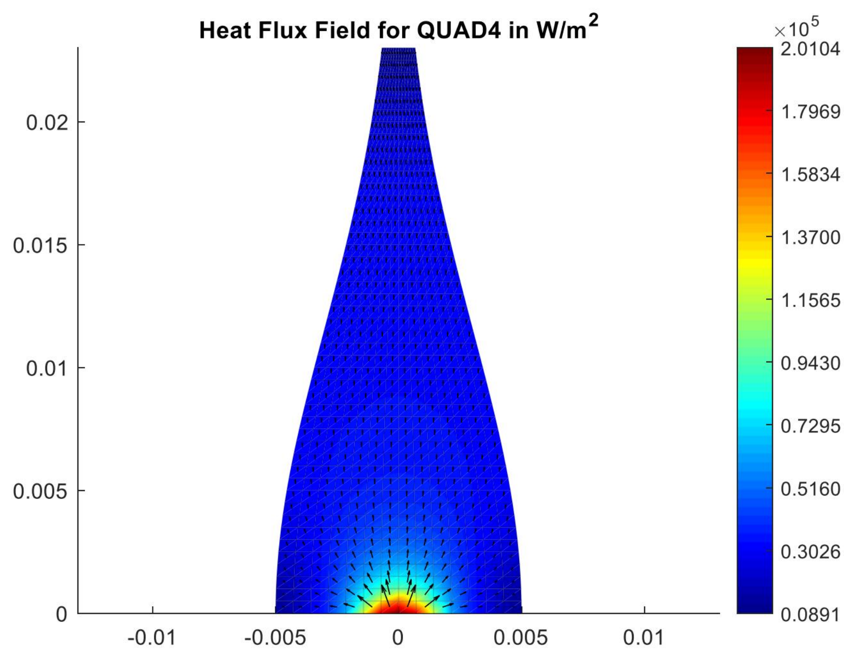
2. Representative Solution

2.1. Temperature field from MATLAB solution for QUAD4



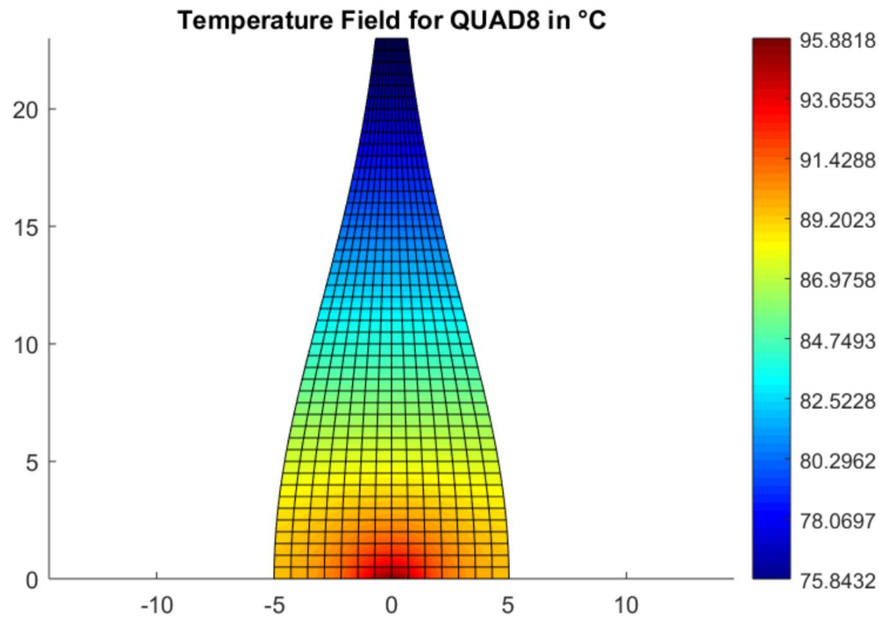
Maximum Temperature = 95.921ºC

2.2. Heat flux field from MATLAB solution for QUAD4



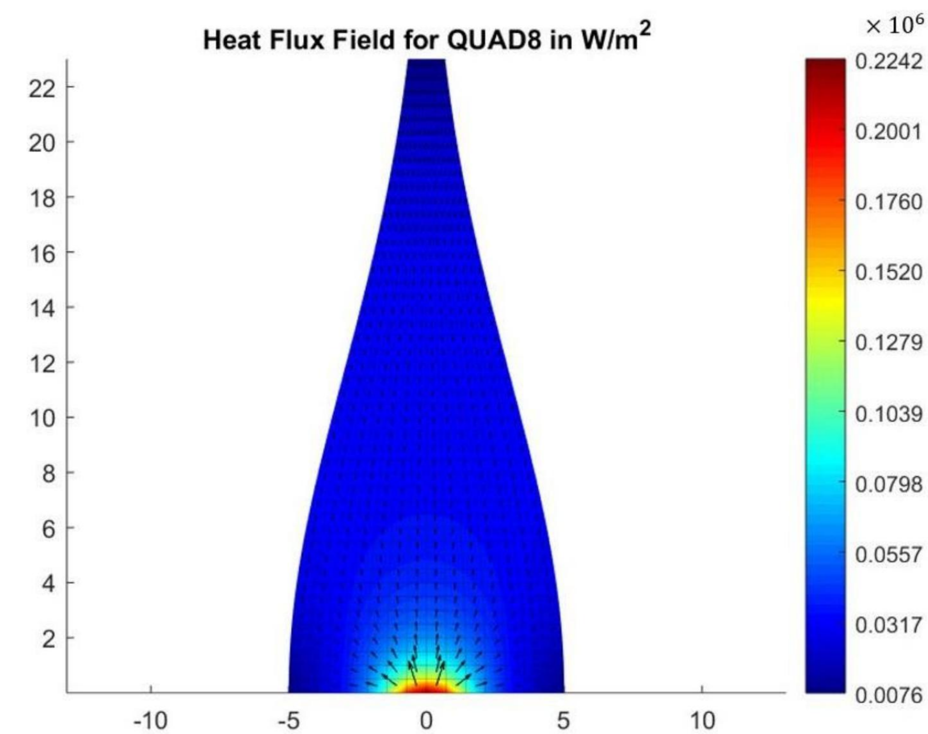
Maximum Heat Flux = 201039.588 W/m^2

2.3. Temperature field from MATLAB solution for QUAD8



Maximum Temperature = 95.882ºC

2.4. Heat flux field from MATLAB solution for QUAD8



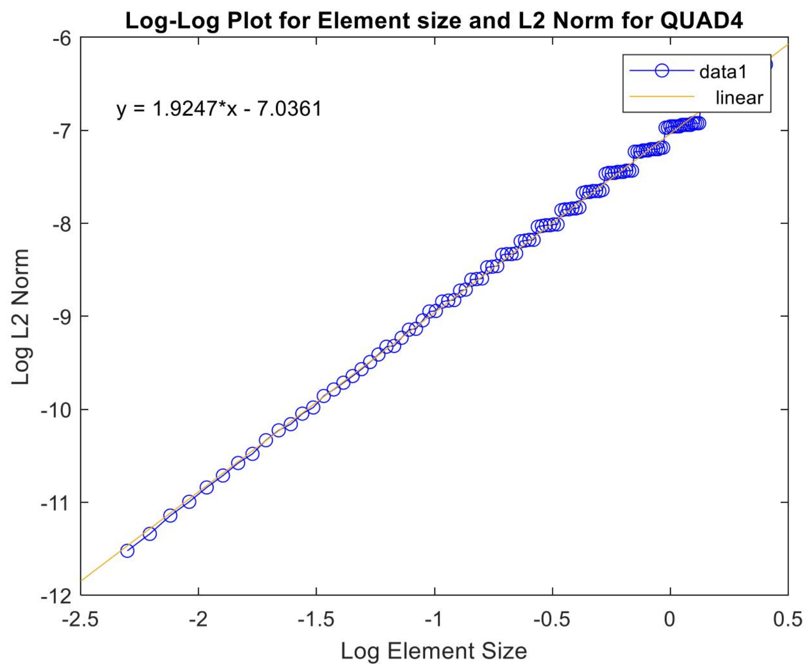
Maximum Heat Flux = 0.2242×10^6 W/m^2

2 Representative solution **15 / 15**

- ✓ **- 0 pts** **Both temperature and flux appear accurate**
- **- 0 pts** temperature not localized in center of fin (probable unit conversion error)
- **- 5 pts** temperature field incorrect
- **- 5 pts** heat flux incorrect
- **- 15 pts** plots copied from another group
- **- 2 pts** missing temperature and/or heat flux units
- **- 3 pts** Flux projection for QUAD8 incorrect

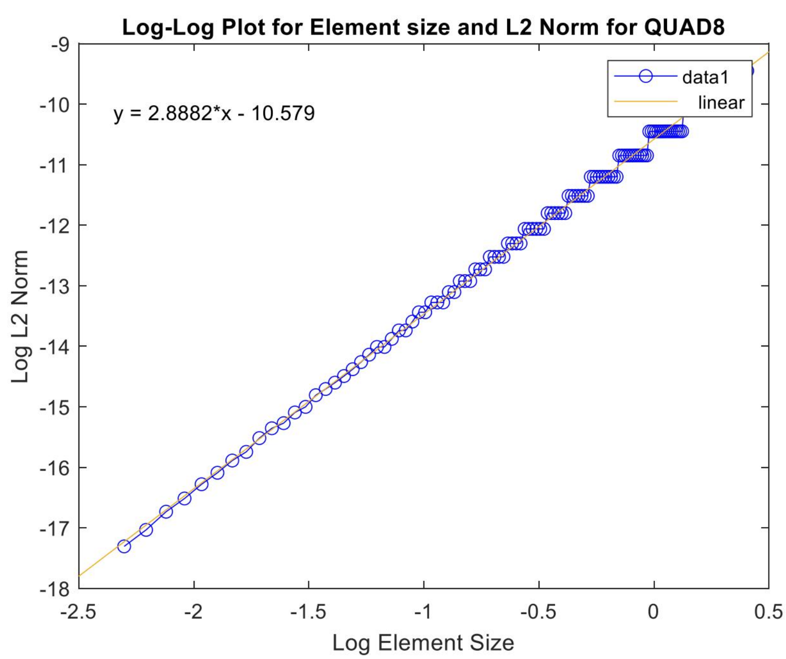
3. Convergence Test

3.1. Convergence Test for QUAD4



Rate of Convergence = 1.9247

3.2. Convergence Test for QUAD8



Rate of Convergence = 2.8882

3 Convergence test (MAE 598 only) **30 / 30**

✓ **- 0 pts** **Correct**

- 5 pts report does not state what the manufactured solution is

- 10 pts convergence of Q4 elements is incorrect

- 5 pts convergence of Q8 elements is incorrect

- 10 pts missing computed rates of convergence

- 30 pts all group members enrolled in MAE 404

4. Mesh Refinement Test

4.1. Mesh Refinement Test for QUAD4

Linear Quad4 Element						
Element size h (mm)	T_{max}	Q_{max}	Error in Temperature (%)	Error in Heat Flux (%)	p	c
2	104.549	0.1362853	0.369132715	-0.068503301	For Temperature	
1	94.392	0.1487334	0.313777	-0.040276229	0.234400	0.313777
0.5	95.921	0.2010396	0.266722514	-0.02368024		
0.25	95.881	0.2107971	0.226724391	-0.013922698	For Heat Flux	
0.125	95.847	0.216534	0.192724449	-0.008185792	0.766245	-0.040276
0.0625	95.867	0.2207656	0.163823192	-0.004812802		

We can see that the error approaches zero with increase in mesh refinement

4.2. Mesh Refinement Test for QUAD8

Quadratic Quad8 Element						
Element size h (mm)	T_{max}	Q_{max}	Error in Temperature (%)	Error in Heat Flux (%)	p	c
2	95.232	0.1831	0.800333413	-0.0064	For Temperature	
1	95.778	0.1971	0.114333345	-0.0032	2.807355	0.114333
0.5	95.882	0.2242	0.016333335	-0.0016		
0.25	95.868	0.224	0.002333334	-0.0008	For Heat Flux	
0.125	95.866	0.2244	0.000333333	-0.0004	1.000000	-0.003200
0.0625	95.866	0.2249	4.76191E-05	-0.0002		

We can see that the error approaches zero with increase in mesh refinement

4 Mesh refinement test 20 / 20

✓ **- 0 pts** **Correct**

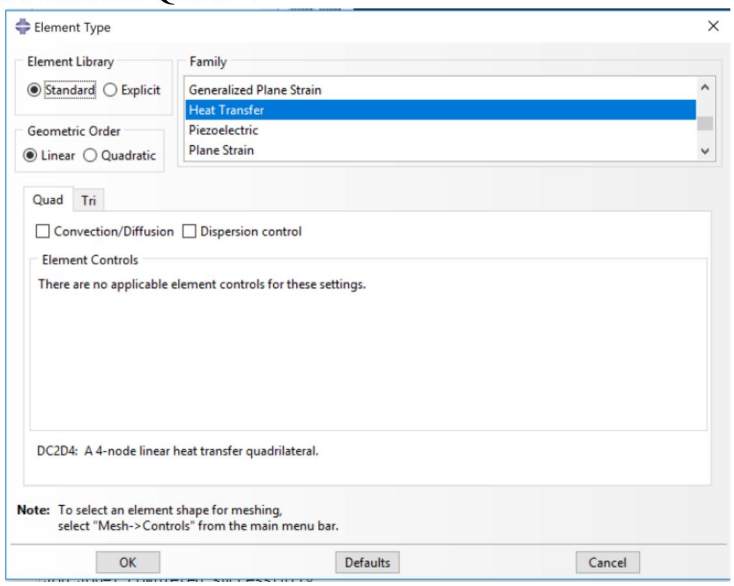
- 5 pts missing estimate of error for each mesh

- 5 pts does not show convergence in Tmax and/or qmax

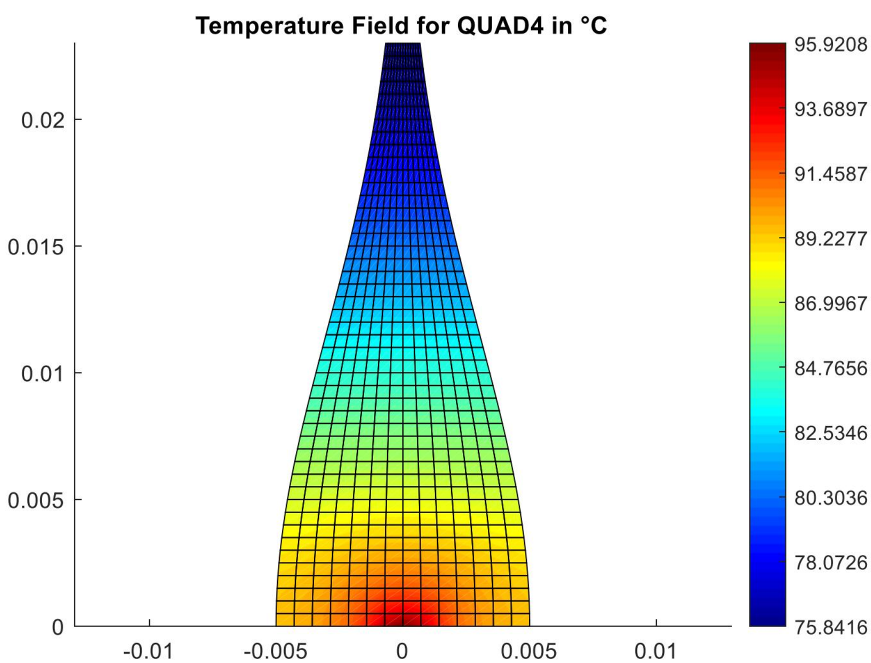
5. Comparison with ABAQUS

5.1. <u>For QUAD4</u>

5.1.1 Temperature field for QUAD4

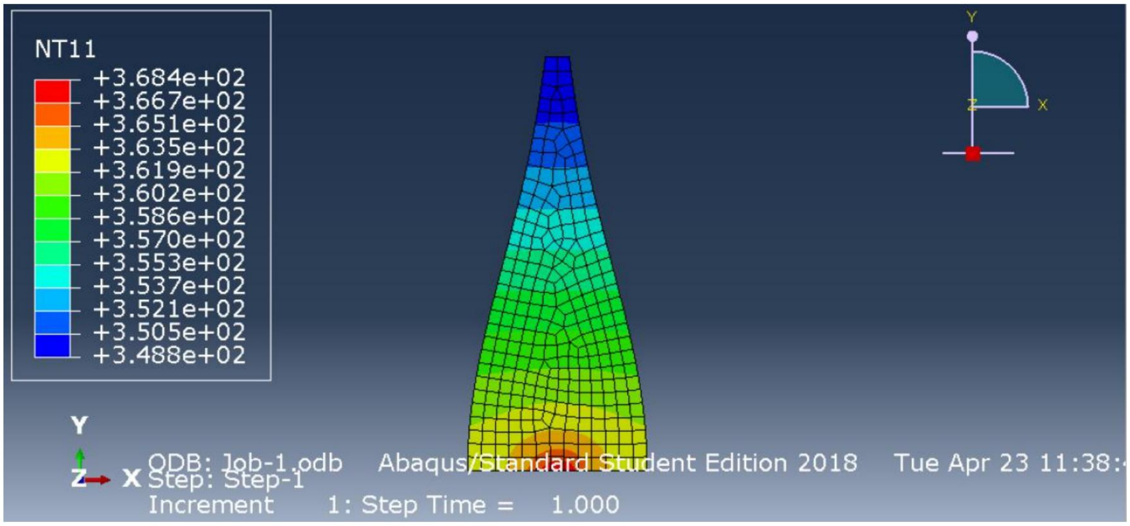


Element type



Temperature Field for QUAD4 from MATLAB (ºC)

Maximum Temperature = 95.921ºC

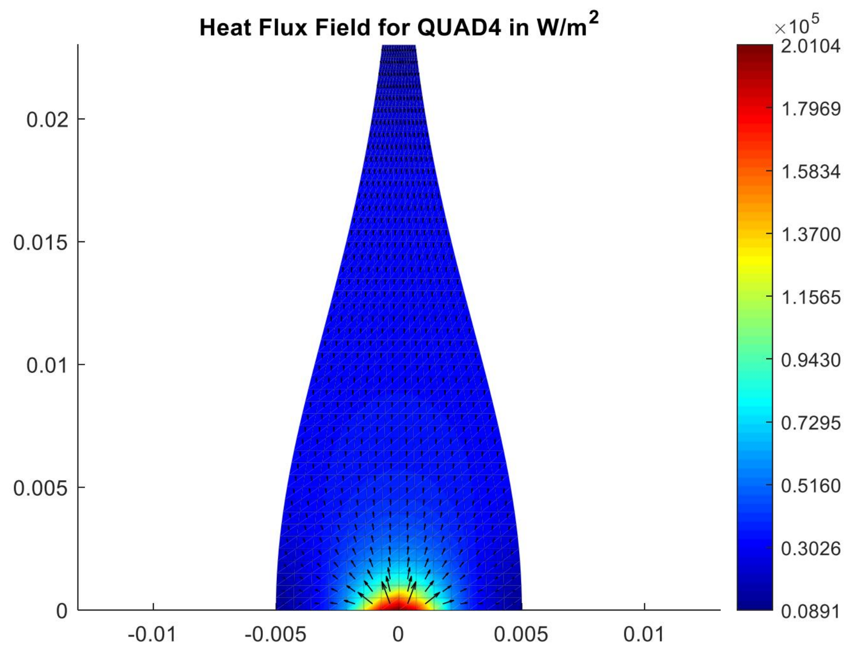


Temperature Field for QUAD4 from ABAQUS (Note: The units in Abaqus are in Kelvin)

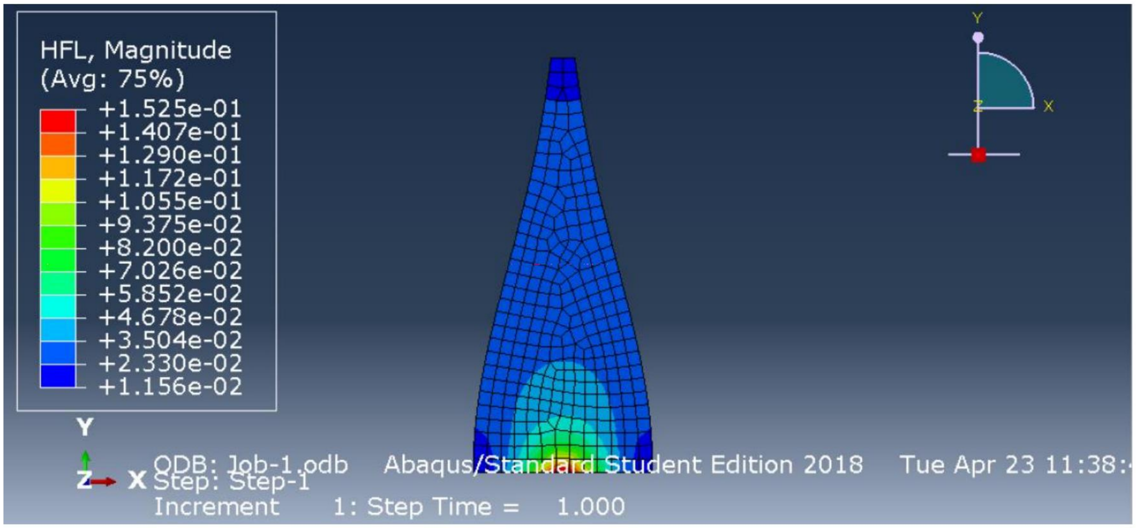
Maximum Temperature = 368.4 K = 368.4-273= 95.4

Difference in values = 0.00543%

5.1.2 Heat Flux Field for QUAD4



Heat Flux Field for QUAD4 from MATLAB
Maximum Heat Flux = 0.201039588 W/mm^2

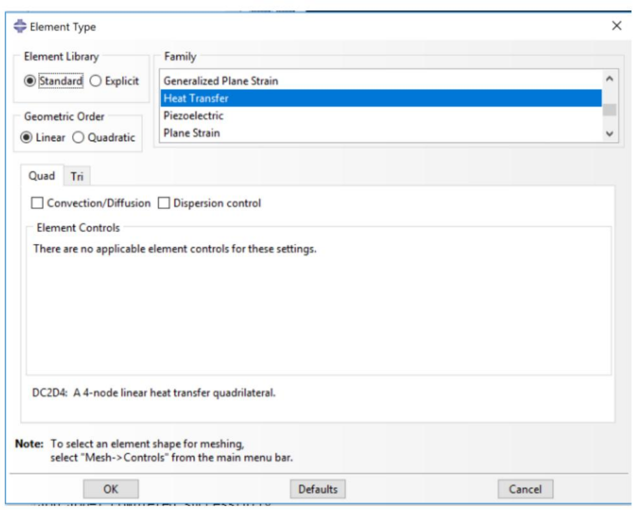


Heat Flux Field for QUAD4 from ABAQUS
Maximum Heat Flux = 0.1525 W/mm^2

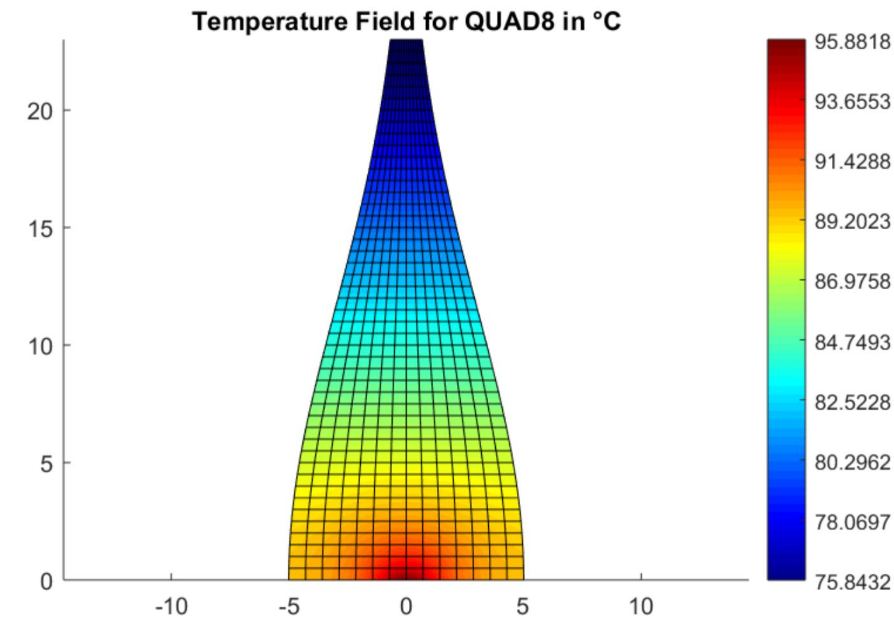
This value would be closer to the MATLAB output for a finer mesh.

5.2. For QUAD8

5.2.1 Temperature Field for QUAD8

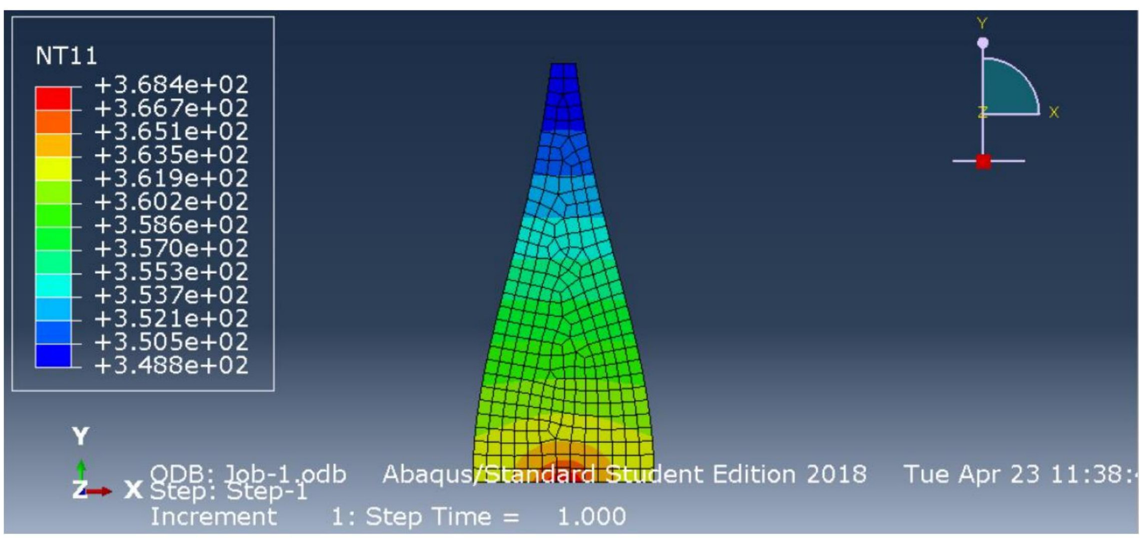


Element type



Temperature Field for QUAD8 from MATLAB (ºC)

Maximum Temperature = 95.882ºC

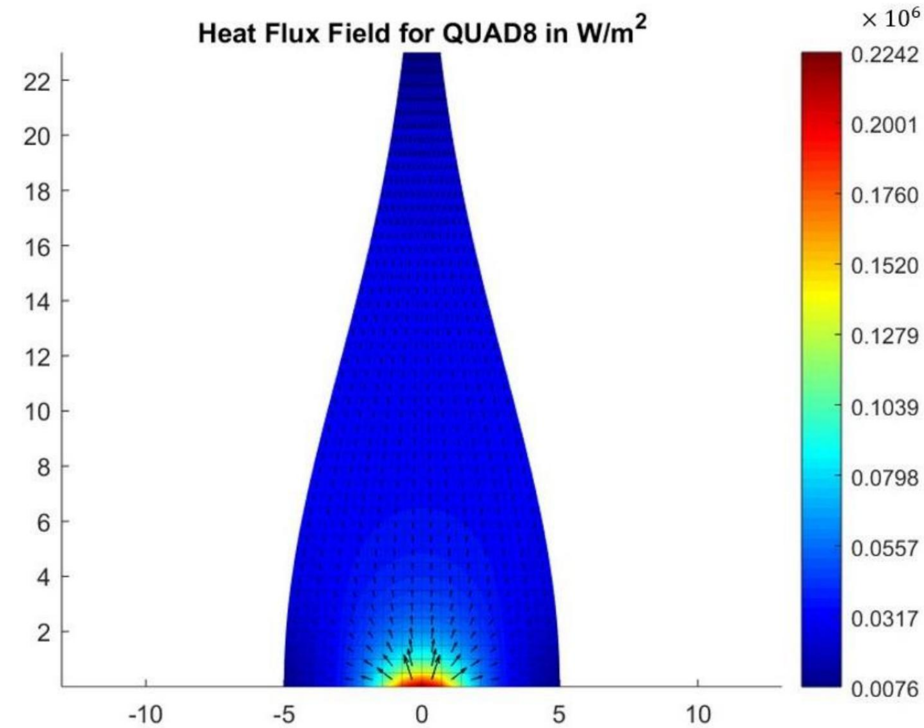


Temperature Field for QUAD8 from ABAQUS (Note: The units in Abaqus are in Kelvin)

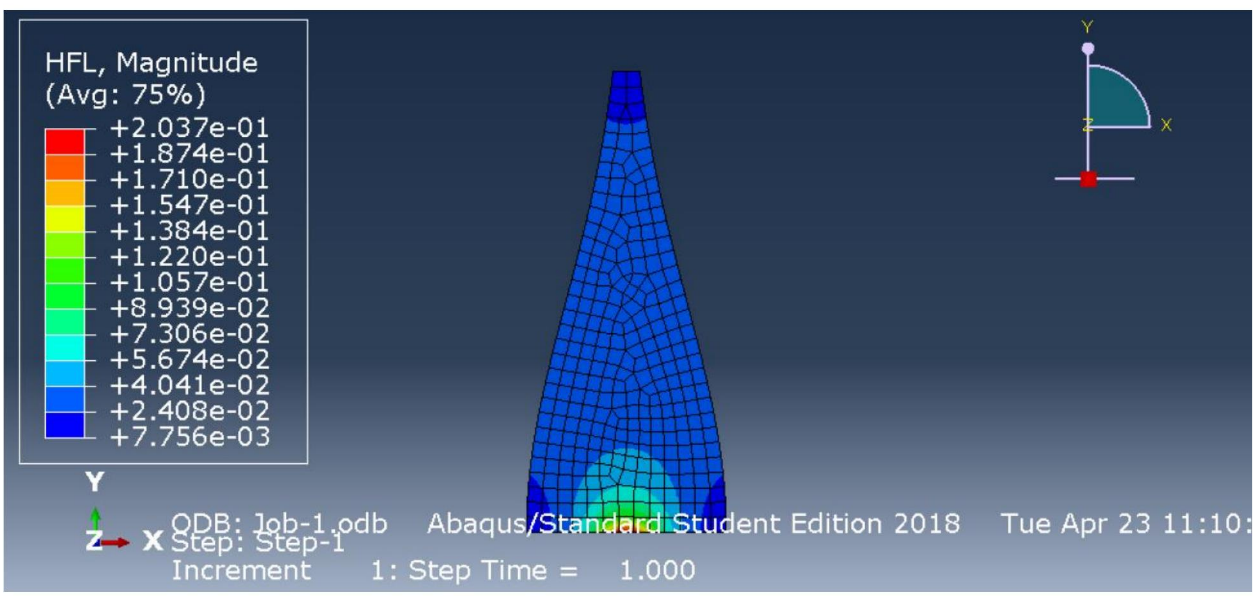
Maximum Temperature = 368.4 K = 368.4-273= 95.4ºC

Difference in values = 0.00502%

5.2.2 Heat flux Field for QUAD8



Heat Flux Field for QUAD8 from MATLAB
Maximum Heat flux in MATLAB = 0.2242 W/mm^2



Heat Flux Field for QUAD8 from ABAQUS
Maximum Heat flux in Abaqus = 0.2037 W/mm^2

This value would be closer to the MATLAB output for a finer mesh.

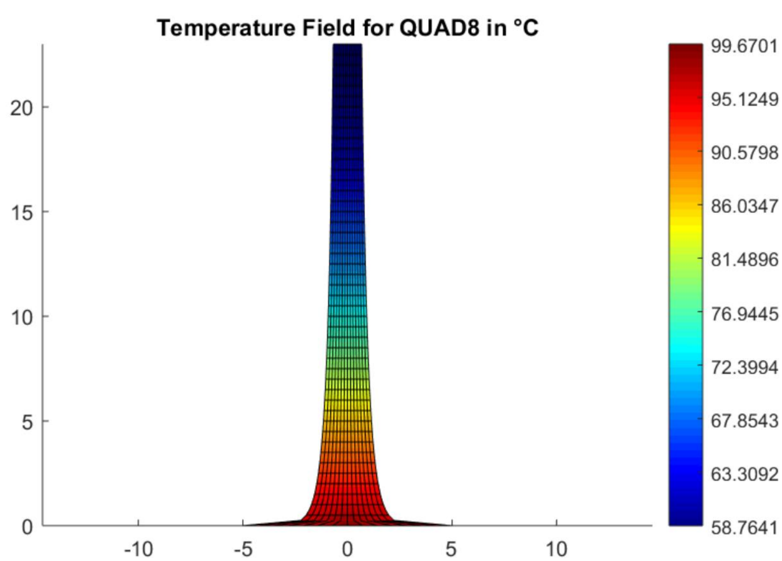
5 Comparison to commercial FEM code 20 / 20

- ✓ **- 0 pts** Correct
- **- 5 pts** missing comparison of heat flux
- **- 10 pts** abaqus heat flux not prescribed correctly
- **- 3 pts** big discrepancy in heat flux
- **- 20 pts** missing
- **- 5 pts** not showing matlab solution in comparison

6. **Determine the optimal design parameters (l_y and β) so that the least amount of material is used to keep the temperature of the fin below 100ºC the ambient temperature**

Lx	Ly	Beta	Area (mm^2)	Temperature (ºC)
10	23	2	137.5731	95.882
10	23	3	158.8682	94.783
10	18	3	124.331	113.968
10	20	3	138.146	105.198
10	23	1	99.436	98.284
10	23	0.5	68.3093	99.166
10	23	0.3	53.1869	99.257
10	22.6	0.3	52.2619	100.106
10	23	0.2	45.413	99.67
10	23	0.1	37.8988	101.033

Observation and Results:
When Ly = 23 for Beta = 3; Temperature increases to 113.968ºC and area also increases to 158.86mm^2
When Ly = 23 for Beta = 1; Temperature decreases to 98.284ºC and area decreases to 99.436mm^2
Now, Beta being reduced further to 0.3 and Ly reduced to 22.6mm area obtained is 52.26mm^2 but temperature rises to 100.106ºC
Therefore, the most optimum design will be for:
Ly = 23, Beta = 0.2 for which Area = 45.413 mm^2, Maximum Temperature = 99.67ºC



6 Optimal design parameters **15 / 15**

- ✓ **- 0 pts** **Correct**
- **- 2 pts** did not explore sensitivity of beta parameter
- **- 3 pts** insufficient description of procedure used
- **- 15 pts** missing

7. MATLAB Code

7.1. MATLAB code for Temperature Field and Heat Flux Field for QUAD4

```
function [] = svmathe1_project2_QUAD4()
    lx = 0.01;
    ly = 0.023;
    h = 0.0005;
    beta = 2;
    etype = 'QUAD4';
    q0 = 225000;
    mesh = fin_mesh(lx, ly, h, beta, etype);
    qpts = [[-1 1 1 -1; -1 -1 1 1]/sqrt(3); 1 1 1 1];
    k = 45*eye(2);
    hc = 100*eye(2);
    Ecb = [mesh.bottom_nodes(1:end-1); mesh.bottom_nodes(2:end)];
    Ect = [mesh.top_nodes(1:end-1); mesh.top_nodes(2:end)];
    Ecl = [mesh.left_nodes(1:end-1); mesh.left_nodes(2:end)];
    Ecr = [mesh.right_nodes(1:end-1); mesh.right_nodes(2:end)];
    Ecc = [Ecl Ect Ecr];
    V = spalloc(length(mesh.x), length(mesh.x), 9*length(mesh.x));
    f = zeros(length(mesh.x), 1);
    for c = mesh.conn
        xe = mesh.x(:,c);
        Ke = zeros(length(c));
        for q = qpts
            [~, dNdp] = shape4(q);
            J = xe * dNdp;
            G = dNdp/J;
            Ke = Ke + G * k * G' * det(J)*q(end);
        end
        V(c,c) = V(c,c) + Ke;
    end
    V1 = spalloc(length(mesh.x), length(mesh.x), 9*length(mesh.x));
    for c = Ecc
        xe = mesh.x(:,c);
        Me = zeros(2);
        for q = [[1 -1]/sqrt(3); 1 1]
            [N, dNdp] = shape2(q(1,:));
            J = xe*dNdp;
            Me = Me + N*N'*hc*norm(J)*q(2);
        end
        V1(c,c) = V1(c,c) + Me;
    end
    Bx = [mesh.x(1,mesh.bottom_nodes(1:end))];
    for c = Ecb
        xe = Bx(:, c);
        le = norm(xe(:,2)-xe(:,1));
        for q = [[1 -1]/sqrt(3); 1 1]
            N = 0.5*[1-q(1); 1+q(1)];
            x = xe*N;
```

```
            qbar = q0*exp(-((x*1000)^2));
            f(c) = f(c) + (N*qbar)*(le/2)*q(end);
        end
    end
    d = (V+V1)\f;
    M = spalloc(length(mesh.x), length(mesh.x), 9*(length(mesh.x)));
    y = zeros(length(mesh.x),2);
    for c = mesh.conn
        xe = mesh.x(:,c);
        de = d(c)';
        ae = zeros(length(c));
        for q = qpts
            [N,dNdp] = shape4(q);
            J = xe*dNdp;
            G = dNdp/J;
            qb = -de*G*k;
            ae = ae+N*N'*det(J)*q(end);
            y(c,:) = y(c,:)+N*qb*det(J)*q(end);
        end
        M(c,c) = M(c,c)+ae;
    end
        heatflux = M\y;
        figure()
    p.faces=mesh.conn';
    p.vertices=mesh.x';
    p.facecolor='interp';
    p.facevertexcdata = d;
    title('Temperature Field for QUAD4 in °C');
    patch(p);
    colorbar;
    colormap(jet);
    he=colorbar;
    t=get(he,'Limits');
    set(he,'Ticks',linspace(t(1),t(2),10))
    axis equal
    figure()
    p.facevertexcdata = sqrt(heatflux(:,2).^2+heatflux(:,1).^2);
    title('Heat Flux Field for QUAD4 in W/m^2');
    mesh.plot_nodal(vecnorm(heatflux')');
    axis equal
    colorbar;
    colormap(jet);
    he=colorbar;
    t=get(he,'Limits');
    set(he,'Ticks',linspace(t(1),t(2),10))
    maxT = max(d);
    fprintf("Maximum Temperature is %.3f\n",maxT);
    maxflux = max(vecnorm(heatflux'))
    fprintf("Maximum Heat flux is %.3f\n",maxflux)
        qq = [];
        xx = [];
```

```
    for c = mesh.conn
        xe = mesh.x(:,c);
        de = d(c)';
        [N,dNdp] = shape4([0;0]);
        J = xe * dNdp;
        G = dNdp / J;
        qq(end+1, :) = -de * G * k;
        xx(end+1, :) = xe * N;
    end
    hold on;
    quiver(xx(:,1), xx(:,2), qq(:,1), qq(:,2), 1.0, 'color', 'k');

end
function [N, dNdp] = shape2(p)
     N = [0.5*(1-p), 0.5*(1+p)]';
     dNdp = [-1/2, 1/2]';
end

function [N, dNdp] = shape4(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*[-(1-p(2)), -(1-p(1));...
                  (1-p(2)), -(1+p(1));...
                  (1+p(2)),  (1+p(1));...
                 -(1+p(2)),   (1-p(1))];
end
```

7.2. MATLAB code for Convergence Test for QUAD4

```
function []= svmathe1_project2_QUAD4_Convergence()
    h1=[0.1:0.01:1.5]
         Aq=[];
         for j=1:length(h1)
        lx=0.01;
        ly=0.023;
        k=45;
        h=h1(j)/1000;
        beta=2;
        mesh=fin_mesh(lx, ly, h, beta, 'QUAD4');
        [qpts]=svmathe1_hw9(3);
        V = spalloc(mesh.num_nodes, mesh.num_nodes, 9*mesh.num_nodes);
        f = zeros(mesh.num_nodes, 1);
        for c = mesh.conn
            xe = mesh.x(:,c);
            Ke = zeros(4);
            for q = qpts
                [N, dNdp] = shape(q);
                J = xe * dNdp;
```

```
                x=xe(1,:)*N;
                y=xe(2,:)*N;
                G = dNdp/J;
                Ke = Ke + G*k*G' * det(J)*q(end);
                smanu=(566100000*sin(10000*x^2 + (1000000*y^2)/529))/529 +
18000000000*x^2*cos(10000*x^2 + (1000000*y^2)/529) +
(18000000000000*y^2*cos(10000*x^2 + (1000000*y^2)/529))/279841;
                f(c) = f(c) + N * smanu * det(J)*q(end);
            end
            V(c,c) = V(c,c) + Ke;
            end
            Ecb = [mesh.bottom_nodes(1:end-1); mesh.bottom_nodes(2:end)];
            Ect = [mesh.top_nodes(1:end-1); mesh.top_nodes(2:end)];
            Ecl = [mesh.left_nodes(1:end-1); mesh.left_nodes(2:end)];
            Ecr = [mesh.right_nodes(1:end-1); mesh.right_nodes(2:end)];
            Ecc = [Ecl Ect Ecr];
            qpts_e=quadrature(3);
            for c = Ecb
                xe = mesh.x(:, c);
                for q=qpts_e
                    [N, dNdp] = shape2(q);
                    J = xe * dNdp;
                    x=xe(1,:)*N;
                    y=xe(2,:)*N;
                    n=[0,-1];
                    qmanu=[900000*x*sin(10000*x^2 +
(1000000*y^2)/529),(90000000*y*sin(10000*x^2 + (1000000*y^2)/529))/529];
                    qbar=dot((qmanu),n);
                    f(c) = f(c) + N * qbar * norm(J)*q(end);
                end
            end
            for c = Ecc
                xe = mesh.x(:, c);
                V(c,:)=0;
                V(c,c)=eye(length(c));
                x=xe(1,:);
                y=xe(2,:);
                Tmanu=cos(10000*x.^2 + (1000000*y.^2)/529);
                f(c) = (Tmanu);
            end
            T=V\f
            wt=zeros();
            wb=zeros();
            for c=mesh.conn
                xe=mesh.x(:,c);
                de=T(c)';
            for q=qpts
                [N,dNdp]=shape(q);
                J=xe*dNdp;
                x=xe(1,:)*N;
                y=xe(2,:)*N;
```

```
                Tmanu=cos(10000*x^2 + (1000000*y^2)/529);
                aex=(Tmanu);
                ah=de*N;
                wt=wt+(aex-ah)^2*det(J)*q(end);
                wb=wb+(aex)^2*det(J)*q(end);
                a1=sqrt(wt/wb);
            end
        end
        Aq(end+1)=(a1)
    end
    plot(log(h1),log(Aq),'b-o')
    title('Log-Log Plot for Element size and L2 Norm for QUAD4');
    ylabel('Log L2 Norm');
    xlabel('Log Element Size');
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*[-(1-p(2)), -(1-p(1));
                  (1-p(2)), -(1+p(1));
                  (1+p(2)),  (1+p(1));
                 -(1+p(2)),  (1-p(1))];
end
function[N, dNdp]=shape2(p)
    N=0.5*[1-p(1);p(1)+1];
    dNdp=[-0.5;0.5];
end
function [qpts_e] = quadrature(n)
    u = 1:n-1;
    u = u./sqrt(4*u.^2 - 1);
    A = zeros(n);
    A(2:n+1:n*(n-1)) = u;
    A(n+1:n+1:n^2-1) = u;
    [v, x] = eig(A);
    [x, k] = sort(diag(x));
    qpts_e = [x'; 2*v(1,k).^2];
end
```

7.3. MATLAB code for Temperature Field and Heat Flux Field for QUAD8

```
function [] = svmathe1_project_2_QUAD8()
    lx = 10;
    ly = 23;
    beta = 2;
    h = 0.5;
    k = 45*10^-3*eye(2);
    mesh = fin_mesh(lx, ly, h, beta, 'QUAD8');
```

```
    qpts = svmathe1_hw9(3);
    Egq = [mesh.bottom_nodes(1:2:end-2);mesh.bottom_nodes(2:2:end-
1);mesh.bottom_nodes(3:2:end)];
    V = spalloc(length(mesh.x),length(mesh.x),9*length(mesh.x));
    for c = mesh.conn
        xe = mesh.x(:,c);
        Ke = zeros(length(c));
        size(Ke)
        for q = qpts
            [N,dNdp] = shape_q8(q);
            J = xe*dNdp;
            G = dNdp/J;
            Ke = Ke + G*k*G'*det(J)*q(3);
        end
        V(c,c) = V(c,c) + Ke;
    end
    right = flip(mesh.right_nodes);
    Egh = [mesh.left_nodes(1:end-1),mesh.top_nodes(1:end-1),right];
    Eghc(1,:) = Egh(1:2:end-2);
    Eghc(2,:) = Egh(2:2:end-1);
    Eghc(3,:) = Egh(3:2:end);
    V1 = spalloc(length(mesh.x),length(mesh.x),9*length(mesh.x));
    for c = Eghc
        xe = mesh.x(:,c);
        ce = zeros(length(c));
        for q = quadrature(3)
            [N,dNdp] = svmathe1_hw5(q(1),3);
            dx = xe(:,3) - xe(:,1);
            J = norm(dx)/2;
            ce = ce + N*N'*(100*10^-6)*J*q(2);
        end
        V1(c,c) = V1(c,c) + ce;
    end
    f = zeros(mesh.num_nodes, 1);
    Bx = [mesh.x(1,mesh.bottom_nodes(1:end))];
    for c = Egq
        xe = Bx(:,c);
        le = norm(xe(:,3) - xe(:,1));
        for q = quadrature(3)
            [N,dNdp] = svmathe1_hw5(q(1),3);
             x = xe*N;
            qbar = 225000*10^-6*exp(-(x)^2);
            f(c) = f(c) + N*qbar*(le/2)*q(2);
        end
    end
    d = (V+V1)\f;
         M = spalloc(length(mesh.x), length(mesh.x), 9*(length(mesh.x)));
         y = zeros(length(mesh.x),2);
         for c = mesh.conn
        xe = mesh.x(:,c);
        de = d(c)';
```

```
        Me = zeros(length(c));
        for q = qpts
            [N,dNdp] = shape_q8(q);
            J = xe*dNdp;
            G = dNdp/J;
            Qb = -de*G*k;
            Me = Me+N*N'*det(J)*q(end);
            y(c,:) = y(c,:)+N*Qb*det(J)*q(end);
        end
        M(c,c) = M(c,c)+Me;
    end
    heatflux = M\y;
         figure()
    p.faces=mesh.conn';
    p.vertices=mesh.x';
    if size(mesh.conn,1) == 8
        p.faces = p.faces(:,[1, 5, 2, 6, 3, 7, 4, 8]);
    end
    p.facecolor='interp';
    p.facevertexcdata = d;
    title('Temperature Field for QUAD8 in °C');
    patch(p);
    axis equal
    colorbar;
    colormap(jet);
    he=colorbar;
    t=get(he,'Limits');
    set(he,'Ticks',linspace(t(1),t(2),10))
    figure()
    title('Heat Flux Field for QUAD8 in W/m^2');
    mesh.plot_nodal(vecnorm(heatflux')');
    axis equal
    colorbar;
    colormap(jet);
    he=colorbar;
    t=get(he,'Limits');
    set(he,'Ticks',linspace(t(1),t(2),10))
    maxT = max(d);
    fprintf("Maximum Temperature is %.3f\n",maxT);
    maxflux = max(vecnorm(heatflux'))
    fprintf("Maximum flux is %.3f\n",maxflux)
         qq = [];
         xx = [];
         for c = mesh.conn
        xe = mesh.x(:,c);
        de = d(c)';
        [N,dNdp] = shape_q8([0;0]);
        J = xe * dNdp;
        G = dNdp / J;
        qq(end+1, :) = -de * G * k;
        xx(end+1, :) = xe * N;
```

```
    end
         hold on;
         quiver(xx(:,1), xx(:,2), qq(:,1), qq(:,2), 1.0, 'color', 'k');
end
function [N, dNdp] = shape_q8(p)
        N = [-0.25*(1-p(1))*(1-p(2))*(1+p(1)+p(2));
         0.25*(1+p(1))*(1-p(2))*(p(1)-p(2)-1);
         0.25*(1+p(1))*(1+p(2))*(p(1)+p(2)-1);
         -0.25*(1-p(1))*(1+p(2))*(p(1)-p(2)+1);
         0.5*(1-p(1)^2)*(1-p(2));
         0.5*(1-p(2)^2)*(1+p(1));
         0.5*(1-p(1)^2)*(1+p(2));
         0.5*(1-p(2)^2)*(1-p(1))];
     dNdp = [0.25*(1-p(2))*(2*p(1)+p(2)), -0.25*(-1+p(1))*(p(1)+2*p(2));
             0.25*(-1+p(2))*(-2*p(1)+p(2)), 0.25*(1+p(1))*(-p(1)+2*p(2));
             0.25*(1+p(2))*(2*p(1)+p(2)), 0.25*(1+p(1))*(p(1)+2*p(2));
             -0.25*(1+p(2))*(-2*p(1)+p(2)), -0.25*(-1+p(1))*(-p(1)+2*p(2));
             p(1)*(-1+p(2)), 0.5*(1+p(1))*(-1+p(1));
             -0.5*(1+p(2))*(-1+p(2)), -p(2)*(1+p(1));
             -p(1)*(1+p(2)), -0.5*(1+p(1))*(-1+p(1));
             0.5*(1+p(2))*(-1+p(2)), p(2)*(-1+p(1))];

end
function [qpts] = quadrature(n)
    u = 1:n-1;u = u./sqrt(4*u.^2 - 1);
    A = zeros(n);
    A(2:n+1:n*(n-1)) = u;
    A(n+1:n+1:n^2-1) = u;
    [v, x] = eig(A);
    [x, k] = sort(diag(x));
    qpts = [x'; 2*v(1,k).^2];
end
```

7.4. MATLAB Code for Convergence Test for QUAD8

```
function []= svmathe1_project2_QUAD8_Convergence()
         h1=[0.1:0.01:1.5];
         Aq=[];
    for j=1:length(h1)
        lx=0.01;
        ly=0.023;
        k=45;
        h=h1(j)/1000;
        beta=2;
        mesh=fin_mesh(lx, ly, h, beta, 'QUAD8');
        [qpts]=svmathe1_hw9(3);
        V = spalloc(mesh.num_nodes, mesh.num_nodes, 9*mesh.num_nodes);
        f = zeros(mesh.num_nodes, 1);
        for c = mesh.conn
```

```
            xe = mesh.x(:,c);
            Ke = zeros(length(c));
            for q = qpts
                [N, dNdp] = shape8(q);
                J = xe * dNdp;
                x=xe(1,:)*N;
                y=xe(2,:)*N;
                G = dNdp/J;
                Ke = Ke + G*k*G' * det(J)*q(end);
                smanu=(566100000*sin(10000*x^2 + (1000000*y^2)/529))/529 +
18000000000*x^2*cos(10000*x^2 + (1000000*y^2)/529) +
(180000000000000*y^2*cos(10000*x^2 + (1000000*y^2)/529))/279841;
                f(c) = f(c) + N * smanu * det(J)*q(end);
            end
            V(c,c) = V(c,c) + Ke;
        end
        Ecb = [mesh.bottom_nodes(1:2:end-2);mesh.bottom_nodes(2:2:end-
1);mesh.bottom_nodes(3:2:end)];
        Ect = [mesh.top_nodes(1:end-1); mesh.top_nodes(2:end)];
        Ecl = [mesh.left_nodes(1:end-1); mesh.left_nodes(2:end)];
        Ecr = [mesh.right_nodes(1:end-1); mesh.right_nodes(2:end)];
        Ecc = [Ecl Ect Ecr];
        qptse=quadrature(3);
        for c = Ecb
            xe = mesh.x(:, c)
            for q=qptse
                [N, dNdp] = svmathe1_hw5(q(1),3);;
                J = xe * dNdp;
                x=xe(1,:)*N;
                y=xe(2,:)*N;
                n=[0,-1];
                qmanu=[900000*x*sin(10000*x^2 +
(1000000*y^2)/529),(90000000*y*sin(10000*x^2 + (1000000*y^2)/529))/529];
                qbar=dot((qmanu),n);
                f(c) = f(c) + N * qbar * norm(J)*q(end);
            end
        end
        for c = Ecc
            xe = mesh.x(:, c);
            V(c,:)=0;
            V(c,c)=eye(length(c));
            x=xe(1,:);
            y=xe(2,:);
            Tmanu=cos(10000*x.^2 + (1000000*y.^2)/529);
            f(c) = (Tmanu);
        end
        T=V\f;
        wt=zeros();
        wb=zeros();
        for c=mesh.conn
            xe=mesh.x(:,c);
```

```
            de=T(c)';
            for q=qpts
                [N,dNdp]=shape8(q);
                J=xe*dNdp;
                x=xe(1,:)*N;
                y=xe(2,:)*N;
                Tmanu=cos(10000*x^2 + (1000000*y^2)/529);
                aex=(Tmanu);
                ah=de*N;
                wt=wt+(aex-ah)^2*det(J)*q(end);
                wb=wb+(aex)^2*det(J)*q(end);
                a1=sqrt(wt/wb);
            end
        end
        Aq(end+1)=(a1)
    end
        plot(log(h1),log(Aq),'b-o')
        title('Log-Log Plot for Element size and L2 Norm for QUAD8');
        ylabel('Log L2 Norm');
        xlabel('Log Element Size');
end

function[N, dNdp] = shape8(p)
N = [(-(1-p(1))*(1-p(2))*(1+p(1)+p(2))/4); (-(1+p(1))*(1-p(2))*(1-p(1)+p(2))/4);...
    (-(1+p(1))*(1+p(2))*(1-p(1)-p(2))/4); (-(1-p(1))*(1+p(2))*(1+p(1)-p(2))/4);...
    ((1-p(1))*(1+p(1))*(1-p(2))/2); ((1+p(1))*(1+p(2))*(1-p(2))/2);...
    ((1-p(1))*(1+p(1))*(1+p(2))/2); ((1-p(1))*(1+p(2))*(1-p(2))/2)];
dNdp = [- ((p(1) - 1)*(p(2) - 1))/4 - ((p(2) - 1)*(p(1) + p(2) + 1))/4, - ((p(1) -
1)*(p(2) - 1))/4 - ((p(1) - 1)*(p(1) + p(2) + 1))/4;...
    ((p(2) - 1)*(p(2) - p(1) + 1))/4 - ((p(1) + 1)*(p(2) - 1))/4, ((p(1) + 1)*(p(2) -
1))/4 + ((p(1) + 1)*(p(2) - p(1) + 1))/4;...
    ((p(1) + 1)*(p(2) + 1))/4 + ((p(2) + 1)*(p(1) + p(2) - 1))/4, ((p(1) + 1)*(p(2) +
1))/4 + ((p(1) + 1)*(p(1) + p(2) - 1))/4;...
    ((p(1) - 1)*(p(2) + 1))/4 + ((p(2) + 1)*(p(1) - p(2) + 1))/4, ((p(1) - 1)*(p(1) -
p(2) + 1))/4 - ((p(1) - 1)*(p(2) + 1))/4;...
    ((p(1) - 1)*(p(2) - 1))/2 + ((p(1) + 1)*(p(2) - 1))/2, ((p(1) - 1)*(p(1) +
1))/2;...
    -((p(2) - 1)*(p(2) + 1))/2, - ((p(1) + 1)*(p(2) - 1))/2 - ((p(1) + 1)*(p(2) +
1))/2;...
    - ((p(1) - 1)*(p(2) + 1))/2 - ((p(1) + 1)*(p(2) + 1))/2, -((p(1) - 1)*(p(1) +
1))/2;...
    ((p(2) - 1)*(p(2) + 1))/2, ((p(1) - 1)*(p(2) - 1))/2 + ((p(1) - 1)*(p(2) + 1))/2];
end

function [N, dNdp] = shape3(p)
    N = [0.5*p.*(p-1); 1-p.*p; 0.5*p.*(p+1)];
    dNdp = [p-0.5 -2*p p+0.5]';
end
function [qpts_e] = quadrature(n)
    u = 1:n-1;
    u = u./sqrt(4*u.^2 - 1);
```

```
    A = zeros(n);
    A(2:n+1:n*(n-1)) = u;
    A(n+1:n+1:n^2-1) = u;

    [v, x] = eig(A);
    [x, k] = sort(diag(x));
    qpts_e = [x'; 2*v(1,k).^2];
end
```

7 Appendix - code 5 / 5

✓ - **0 pts** **Correct**

- **5 pts** no appendix