16.90 Project 2

Spring 2014

1 Fluid Part

1.1

We begin by writing the energy conservation equation in conservation form.

$$c_p \dot{m} \frac{dT_{t,cool}}{dx} = 2\vec{q} \cdot \vec{n} \tag{1}$$

$$\int c_p \dot{m} \frac{dT_{t,cool}}{dx} dx = \int 2\vec{q} \cdot \vec{n} \; dx \tag{2}$$

$$c_p \dot{m} T_{t,cool}(x) = \int 2\vec{q} \cdot \vec{n} \; dx + c \tag{3}$$

The integration constant c imposes the boundary conditions at $x = 0$. This form relates the flux through the control volume to the source. For this situation, the flux function is

$$F(x) = c_p \dot{m} T_{t,cool}(x) \tag{4}$$

The source for this problem is the heat transfer through the turbine blade metal to the cooling flow. This can be expressed as $\boxed{S(x) = 2\vec{q} \cdot \vec{n}}$. Integrating the source function of some control volume i gives the heat that is added to that control volume. This can be expressed as $\boxed{Q_i = \int_{x_L}^{x_R} 2\vec{q} \cdot \vec{n} \; dx}$ where x_L and x_R are the left and right hand boundaries of the control volume, respectively.

1.2

We examine the domain $x \in [0, \; 1cm] = [0, \; 0.01][m]$. The upwind flux is defined as

$$F_{i+\frac{1}{2}} = F(T + i) \tag{5}$$

We discretize the domain into N control volumes.

1	2	...	$i-1$	i	$i+1$	...

The boundaries of the ith control volume are given by $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ which correspond to the left and right boundaries, respectively. In general, this system can be written as

$$F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} 2(1-100x)dx \tag{6}$$

$$F(T_i) - F(T_{i-1}) = 2x - 100x^2 \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = S_i \tag{7}$$

Here, $F_{i+\frac{1}{2}}$ is the flux through the right hand boundary of control volume i, while $F_{i-\frac{1}{2}}$ is the flux through the left hand boundary of the ith control volume. Since we, are using an upwind flux, $F_{i+\frac{1}{2}} = F(T_i)$ which means that the flux at the right hand boundary is equal to the flux as a function of the average value of the cell, which in this problem is the temperature of the ith cell. This means that $F_{i-\frac{1}{2}}$ is equal to the flux as a function of T_{i-1}, the temperature of the $i-1$ control volume. Thus, we can write the discretization in the following form, noting that $F(T_i) = c_p\dot{m}T_i$.

$$c_p\dot{m}(T_i - T_{i-1}) = 2x - 100x^2 \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}$$

$$T_i - T_{i-1} = \frac{2x - 100x^2}{c_p\dot{m}} \Bigg|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = Q_i$$

$$\boxed{T_i - T_{i-1} = \frac{1}{c_p\dot{m}}[2x_{i+\frac{1}{2}} - 100x_{i+\frac{1}{2}}^2 - 2x_{i-\frac{1}{2}} + 100x_{i-\frac{1}{2}}^2] = Q_i} \tag{8}$$

The above equation denotes our discretization of the original equation $c_p\dot{m}\frac{dT}{dx} = 2\vec{q}\cdot\vec{n}$.

1.3

To solve this equation, we first build a system of N linear equations, where N is equal to the number of control volumes. We use the following discretization, which is the same as in part 2, equation (8), of this problem.

In code we compute Q_i by computing

$$q(x) = (2x - 100x^2)\frac{1e6}{c_p\dot{m}}$$

$$Q_i = q(x_{i+\frac{1}{2}}) - q(x_{i-\frac{1}{2}})$$

The 1×10^6 term is to convert to watts, since the $\vec{q}\cdot\vec{n}$ function as given was in megawatts. We set the boundaries of the control volumes as being equal to the domain from 0 to 0.01 discretized by $dx = \frac{1}{N}$. Thus, the values of $x_{i+\frac{1}{2}}$ and $x_{i-\frac{1}{2}}$ are multiples of dx.

We next write equation (8) in matrix form.

$$\begin{bmatrix} 1 & 0 & \cdots & & & \\ -1 & 1 & 0 & \cdots & & \\ 0 & -1 & 1 & 0 & \cdots & \\ & \ddots & \ddots & \ddots & \ddots & \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{bmatrix}$$

This can be summarized as

$$AT = Q \tag{9}$$

To enforce the boundary condition, we note that over control volume 1

$$\begin{aligned} T_1 - T_0 &= \frac{1\times 10^6}{c_p \dot{m}} \left[2x - 100x^2\right]_0^{dx} \\ T_1 &= T_0 + \frac{1\times 10^6}{c_p \dot{m}} \left[2(dx) - 100(dx)^2\right]_0^{dx} \\ Q_1 &= \frac{1\times 10^6}{c_p \dot{m}} \left[2(dx) - 100(dx)^2\right] \end{aligned}$$

where $T_0 = 800$K, the temperature of the air at the left most boundary, which is the inlet to the interior of the blade.

We now solve this system of equation using MATLAB®'s '\' command.

$$T = A \backslash Q$$

This results in a set of N temperatures, which in turn correspond to the values of t for each of the N control volumes.

We now compute the analytic solution.

$$\begin{aligned} c_p \dot{m} \frac{dT_{t,cool}}{dx} &= 2\vec{q}\cdot\vec{n} \\ c_p \dot{m} \frac{dT_{t,cool}}{dx} &= 2(1 - 100x)(1\times 10^6) \\ \int c_p \dot{m} \frac{dT_{t,cool}}{dx} dx &= \int (1-100x)(2\times 10^6) dx \\ c_p \dot{m} T(x) &= 1\times 10^6 (2x - 100x^2) + c \\ T(x) &= \frac{1\times 10^6}{c_p \dot{m}} (2x - 100x^2) + c \end{aligned}$$

We use the boundary condition $T(0) = 800$K to determine c. We also note that $\dot{m} = 0.05$ kg/m and $c_p = 1000$ J/kg-K.

$$\begin{aligned} &800 = \frac{1\times 10^6}{(0.05)(1000)} (2(0) - 100(0)) + c \\ &\Rightarrow c = 800\text{K} \end{aligned}$$

Thus, $T_{analytic}(x) = \dfrac{1 \times 10^6}{c_p \dot{m}}(2x - 100x^2) + 800$ for any general c_p and $\dot{m}$.

$$T_{analytic}(x) = 2 \times 10^4(2x - 100x^2) + 800 \tag{10}$$

We can now compute the analytic solution at any point x.

In order to compute the error of the discretization, we note that each T_i temperature in the ith control volume is an average value over the domain of the control volume. Thus, we wish to compare the value of the temperature in the ith control volume with the average of the analytic solution over the same control volume. To do this, we recall that $\bar{f} = \dfrac{1}{\Delta x}\int_x^{x+\Delta x} f(x)dx$. Thus,

$$\begin{aligned}
\bar{T}(x) &= \frac{1}{\Delta x}\int_x^{x+\Delta x} T(x)dx \\
&= \frac{1}{\Delta x}\int_x^{x+\Delta x} 2 \times 10^4(2x - 100x^2) + 800dx \\
&= \frac{1}{\Delta x}\left[2 \times 10^4\left(x^2 - \frac{100}{3}x^3\right) + 800x\right]_x^{x+\Delta x}
\end{aligned}$$

In MATLAB we compute,

$$T_{av}(x) = \frac{1 \times 10^6}{c_p \dot{m}}\left(x^2 - \frac{100}{3}x^3\right) + 800 \tag{11}$$

We then compute

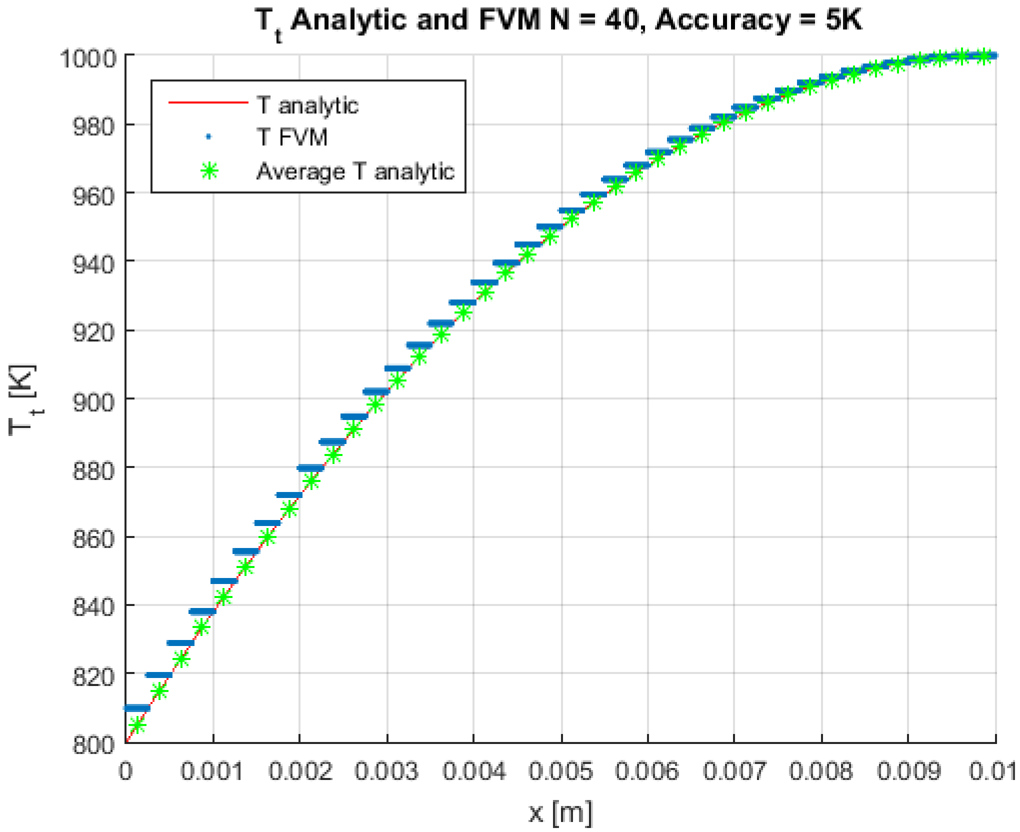
$$\bar{T}(x) = \frac{1}{\Delta x}\left(T_{av}(x + \Delta x) - T_{av}(x)\right)$$

This gives us the average value of $T_{analytic}$ over any domain $[x,\ x + \Delta x]$. For each control volume, these values are the left and right boundaries, which were also used to compute the values of Q_i.

We next compute the maximum error between the discretization and the analytic solution by computing $\max(|\bar{T} - T|)$ where $\bar{T}$ is a vector of the average values of the analytic solution computed over the control volumes, and T is the vector of the values of T_i, the temperature in each control volume.

It could be said that equation (11) is unnecessary since the average value of the analytic solution in each cell is the same as the value at the midpoint. However, the formulation of equation (11) eliminates any doubt about guessing the value of the average temperature using the analytic solution.

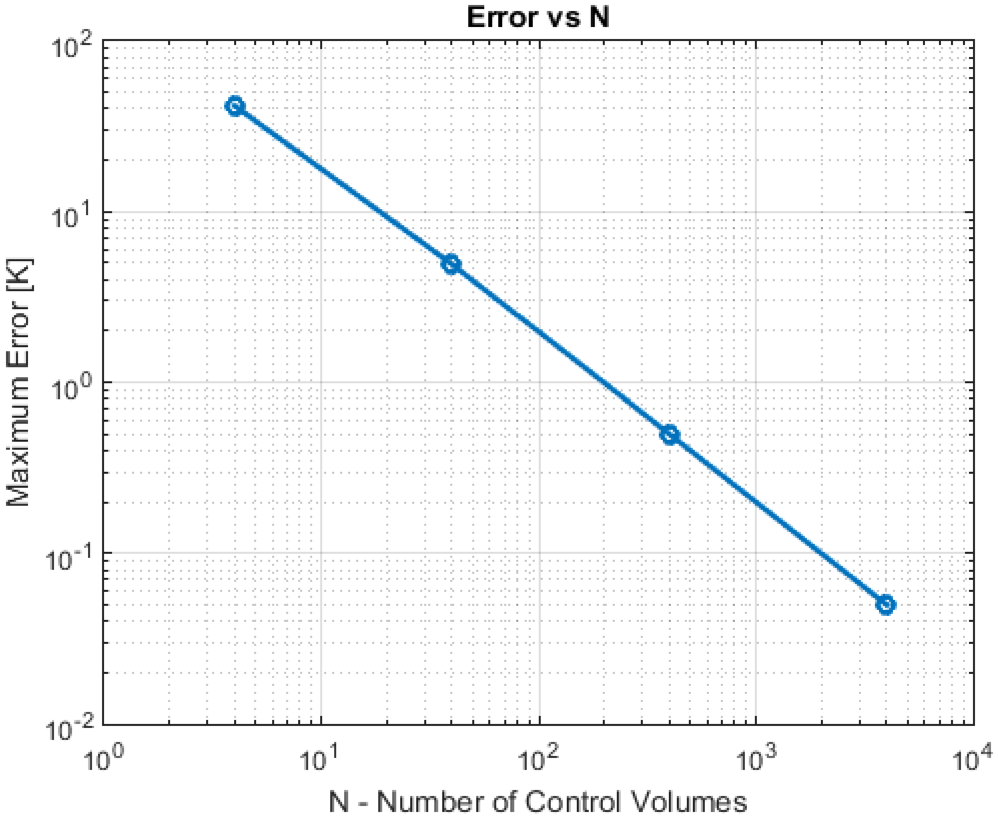
We next set the acceptable tolerance to 5 Kelvin and run the solver for values of N until the desired tolerance is met. The results are plotted below.



The analysis shows that for $\boxed{\text{40 control volumes}}$, the maximum difference between the values of the N control volumes and the average value of the analytic function over these N control volumes is 5 Kelvin or less.

There are a couple of interesting things to note about the results of this analysis. First, we note that the value of T_R for each control volume is equal to the analytic solution at that boundary. This makes sense due to our method for computing T_i. Since equation (8) integrates the source over the control volume and adds it to the value of T_L, we expect that given the first $T_2 = T_0$, which is the boundary condition, that we will always return the value of the analytic solution at all $x_R = x_{i+\frac{1}{2}}$. This results in our approximation overshooting the analytic solution at almost all points.

To examine the order of convergence, we compute the maximum error over a vector of Ns. Here we choose $N = [0.1\ 1\ 10\ 100]$. We compute the maximum error for each value of n and plot them versus N on a loglog plot.



The results show that the $\boxed{\text{order of convergence is } \mathcal{O}(N^1)}$. As N is increased by a factor of 10, the error is subsequently reduced by the same factor. This makes sense because we know that $T_i = T_{analytic}(x_{Ri})$. As the distance between x_L and x_R decreases, so too does the difference between $T(x_R)$ and $T(x_L)$ by the same amount, since $T(x)$ is quadratic. Consecutive values of $T(x)$ converge with the order of x of the derivative of T. Since $\dfrac{dT(x)}{dx} = \mathcal{O}$, we expect the convergence between points to be of the same order. This means that since values of T_i and $T(x_{i+\frac{1}{2}})$ are equal, we also expect the error between our discretization and the analytic solution to converge at the same rate. The numerical results presented in the above plot confirm this for us.

1.4

If the control volumes was non-uniform, the finite volume method would be able to approximate the analytical solution. This can be done by using the derivative of the function to intelligently change the size of the control volumes. For regions where the derivative is large, the grid spacing could be reduced, resulting in a better approximation and thus reducing the overall error. By the same logic, the grid spacing for regions where the temperature is relatively constant can be increased. The implementation of the discretized governing equations would be the same. Care would have to be taken, however, to make sure that the grid spacing did not hide key elements to the solution, such as small oscillations or discontinuities that would be hidden by the non-uniform grid.

If the heat flux was a piecewise-linear function, we most likely see a reduction in the error between the analytic and FVM solutions. Due to our definition of the flux function, it is reasonable to assume that if the heat flux function was piecewise linear, each control volume could be constructed along the breakpoints in the heat flux function. Thus, each control volume could be constructed to contain linear values between $T(x_{i-\frac{1}{2}})$ and $T(x_{i+\frac{1}{2}})$. This would result in the control volumes approximating the analytic solution in a much more accurate way than when the heat flux is quadratic. This is because the midpoint for each control volume and the analytic solution would be the same, or at least very close, thereby greatly reducing the error for a given N control volumes. Care would have to be taken in the construction of the grid for the full benefits of this to be realized. However, with careful construction of the domain, the solution has the potential to be very accurate.

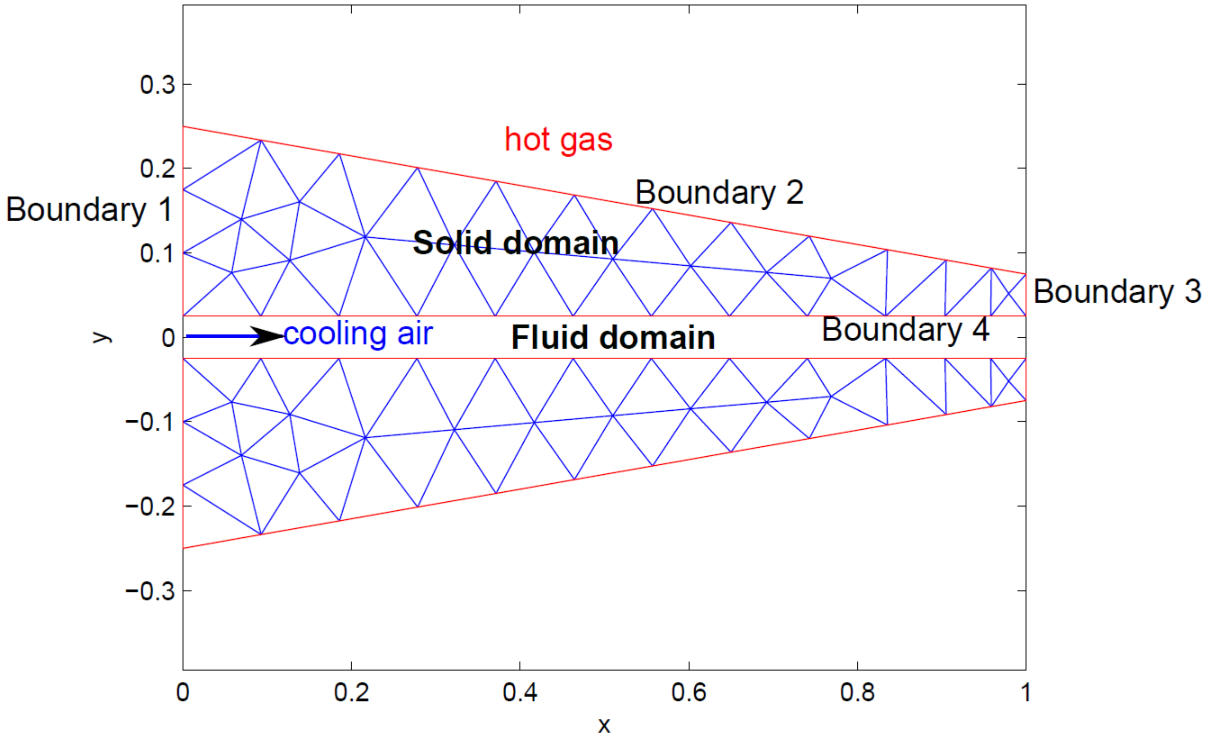
2 Thermal Part

2.1

From the problem statement we begin with

$$\vec{q} = -k\nabla T \tag{12}$$

$$\vec{q}\cdot\vec{n} = h(T - T_t) \tag{13}$$



We now derive the boundary conditions for the blade.

Boundary 1 is adiabatic. Thus, on boundary 1 we have $\vec{q} \cdot \vec{n} = 0$ Boundary 2, 3, and 4 are governed by equation (13). Over the surface we have the following conservation:

$$\int_{\delta\Omega} \vec{q} \cdot \vec{n} \; ds = \int_{\delta\Omega} h(T - T_t) \; ds = \int_{\delta\Omega} -k\nabla T \cdot \vec{n} \; ds$$

Thus, for boundaries 2 and 3, which are on the hot side, we have

$$\int_{\delta\Omega_{2,3}} h(T - T_t) \; ds + \int_{\delta\Omega_{2,3}} k\nabla T \cdot \vec{n} \; ds = 0$$
$$\Rightarrow h(T - T_{t,hot}) + k\nabla t \cdot \vec{n} = 0$$
$$\Rightarrow k\nabla T \cdot \vec{n} + hT = hT_{t,hot}$$

Similarly, for boundary 4 we have the same general equation, but instead substitute in $T_{t,cool}$ for $T_{t,hot}$, the exterior temperature.

$$\nabla T \cdot \vec{n} + hT = hT_{t,cool}$$

Thus, the boundary conditions are given by

$$\begin{aligned} 1 &: \; -k\nabla T = 0 \\ 2 &: k\nabla T \cdot \vec{n_2} + hT = hT_{t,hot} \\ 3 &: k\nabla T \cdot \vec{n_3} + hT = hT_{t,hot} \\ 4 &: k\nabla T \cdot \vec{n_4} + hT = hT_{t,cool} \end{aligned}$$

There are three types of boundary conditions that have been discussed in class thus far.

$$\begin{aligned} \text{Dirichlet} &\rightarrow \tilde{T} \text{ known on the boundary} \\ \text{Neumann} &\rightarrow \tilde{T_x} \text{ known on the boundary} \\ \text{Robin} &\rightarrow \alpha_0\tilde{T} + \alpha_1\tilde{T_x} \text{ known on the boundary} \end{aligned}$$

By examining the boundary conditions derived above, it can be seen that on boundary 1, since $\nabla T = 0$ is known, we have a Neumann boundary condition. On the other three boundaries the equation is of the form $k\nabla T \cdot \vec{n} + hT = hT_t$. This is a Robin boundary condition with $\alpha_0 = h$ and $\alpha_1 = k$.

Thus, the boundary conditions are as follows:

Boundary	Equation	Type
1	$\nabla T = 0$	Neumann
2	$k\nabla T \cdot \vec{n_2} + hT = hT_{t,hot}$	Robin
3	$k\nabla T \cdot \vec{n_3} + hT = hT_{t,hot}$	Robin
4	$k\nabla T \cdot \vec{n_4} + hT = hT_{t,cool}$	Robin

2.2

We now wish to find the residuals in their weak form. We begin with equation (1) from the prompt.

$$0 = \nabla \cdot (k\nabla T) \tag{14}$$

For the equation to be conservative, the following must be true:

$$0 = \int_{\Omega} \nabla \cdot (k\nabla T)\ dA \tag{15}$$

We now find the weighted residual.

$$R = \int_{\Omega} w\nabla \cdot (k\nabla T)\ dA \tag{16}$$

Since we do not know $\nabla^2 T$ and we wish for the order of the derivative on the weight function and the approximation to be equal, we use integration by parts to separate the residual.

$$R = \int_{\Omega} w\nabla \cdot (k\nabla T)\ dA = \int_{\delta\Omega} wk\nabla T \cdot \vec{n}\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA$$

We have thus obtained the weak form of the residual.

$$\boxed{R = \int_{\delta\Omega} wk\nabla T \cdot \vec{n}\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA} \tag{17}$$

Applying the boundary conditions, we find the bilinear form. This has looks the same as the residual due to the nature of the boundary conditions. The bilinear form is made up of all the terms that depend on w and T.

$$\boxed{R = \int_{\delta\Omega} wk\nabla T \cdot \vec{n}\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA} \tag{18}$$

Note that without the boundary conditions, the heat is totally conserved, which means that all heat flow depends on the temperature of the element, with no additional source terms.

We not add the boundary conditions previously derived.

$$
\begin{aligned}
R &= \int_{\delta\Omega} wk\nabla T \cdot \vec{n}\ dx - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA \\
&= \int_{\delta\Omega_1} w(-q_1 \cdot \vec{n_1})\ ds + \int_{\delta\Omega_2} w(-q_2 \cdot \vec{n_2})\ ds + \int_{\delta\Omega_3} w(-q_3 \cdot \vec{n_3})\ ds \\
&+ \int_{\delta\Omega_4} w(-q_4 \cdot \vec{n_4})\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA \\
&= \int_{\delta\Omega_1} w(0)\ ds + \int_{\delta\Omega_2} wh(T_{t_{hot}} - T)\ ds + \int_{\delta\Omega_3} wh(T_{t_{hot}} - T)\ ds \\
&+ \int_{\delta\Omega_4} wh(T_{t_{cool}} - T)\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA \\
R &= \int_{\delta\Omega_2} whT_{t_{hot}}\ ds - \int_{\delta\Omega_2} whT\ ds + \int_{\delta\Omega_3} whT_{t_{hot}}\ ds - \int_{\delta\Omega_3} whT\ ds \\
&+ \int_{\delta\Omega_4} whT_{t_{cool}}\ ds - \int_{\delta\Omega_4} whT\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA
\end{aligned}
$$

Reorganizing this we get

$$
\begin{aligned}
R = \int_{\delta\Omega_2} whT_{t_{hot}}\ ds + \int_{\delta\Omega_3} whT_{t_{hot}}\ ds + \int_{\delta\Omega_4} whT_{t_{cool}}\ ds \\
- \int_{\delta\Omega_2} whT\ ds - \int_{\delta\Omega_3} whT\ ds - \int_{\delta\Omega_4} whT\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA \quad (19)
\end{aligned}
$$

This is the fully expanded form of the weighted residual. This has changed the bilinear form and the linear functionals. By adding in the boundary conditions, we have added some source terms. We now have the following:

$$
\boxed{
\begin{array}{l}
\text{Bilinear Form: } - \int_{\delta\Omega_2} whT\ ds - \int_{\delta\Omega_3} whT\ ds - \int_{\delta\Omega_4} whT\ ds - \int_{\Omega} \nabla w \cdot (k\nabla T)\ dA \\
\text{Linear Functional: } \int_{\delta\Omega_2} whT_{t_{hot}}\ ds + \int_{\delta\Omega_3} whT_{t_{hot}}\ ds + \int_{\delta\Omega_4} whT_{t_{cool}}\ ds
\end{array}
}
$$

We now implement the above equations into a MATLAB function. The skeleton code provided asks for four lines of inputs as follows:

$$M_{edge} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\nabla\phi(\xi) = \begin{bmatrix} \quad \end{bmatrix}$$

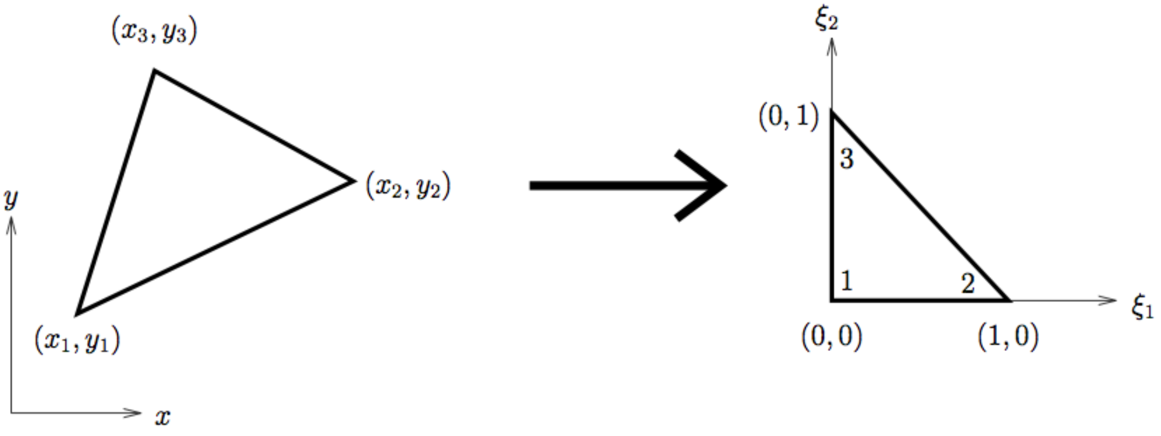
$$K(I, I) = K(I, I) + ...$$

$$F(I) = F(I) + ds * ...$$

In order to compute these lines, we note that for the above residual we have made a Galerkin approximation. This means that

$$w_j = \phi_j$$
$$\tilde{T} = \sum_{i=1}^{N} a_i \phi_i$$

We also note that we are using the following mapping for the reference element.



$$\begin{aligned}
\phi_1(\xi_1,\ \xi_2) &= 1 - \xi_1 - \xi_2 \\
\phi_2(\xi_1,\ \xi_2) &= \xi_1 \\
\phi_3(\xi_1,\ \xi_2) &= \xi_2
\end{aligned}$$

Thus the gradient is given by

$$\nabla\phi_i(\xi_1,\ \xi_2) = \begin{bmatrix} \dfrac{\partial\phi_1}{\partial\xi_1} & \dfrac{\partial\phi_1}{\partial\xi_2} \\ \\ \dfrac{\partial\phi_2}{\partial\xi_1} & \dfrac{\partial\phi_2}{\partial\xi_2} \\ \\ \dfrac{\partial\phi_3}{\partial\xi_1} & \dfrac{\partial\phi_3}{\partial\xi_2} \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

To compute K we note that this statement corresponds to

$$\int_{\Omega} \nabla w \cdot (k\nabla T)\ dA$$

We now transform into the reference frame.

$$\begin{aligned} K_{i,j} &= \int_\Omega \nabla\phi(x,y) \cdot^{-} k \sum_{i=1}^{N} (\nabla\phi_i)\ dxdy \\ &= \int_\Omega \nabla\phi(\xi_1,\ \xi_2) \cdot k\nabla\phi_i(\xi_1,\ \xi_2) A_{x,y}\ d\xi_1 d\xi_2 \end{aligned}$$

Since $\nabla\phi(\xi_1,\ \xi_2)$ is a constant we have $K_{i,j} = k\nabla\phi_j \cdot \nabla\phi_i A_{x,y}$ where A is the area of the element and has already been computed by the skeleton code. The dot product $\nabla\phi_j \cdot \nabla\phi_i$ is a dot product. In matrix algebra this means that we need to compute $\nabla\phi\nabla\phi'$. This will yield a 3×3 matrix.

$$\nabla\phi\nabla\phi' = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

In code we simply state that

$$K(I,I) = K(I,I) + k\nabla\phi\nabla\phi' A$$

We now move onto the boundary elements. We start with the constants $\int_{\delta\Omega} \phi_j(x,y) hT_t\ ds$ where we have substituted in ϕ_j for w_j. We again transform into the reference coordinates:

$$\begin{aligned} \int_{\delta\Omega} \phi_j(x,y) hT_t\ ds &= hT_t \int \phi_j(\xi_1,\ \xi_2) |ds_{x,y}|\ ds_\xi \\ &= hT_t |ds_{x,y}| \int \phi_j(\xi_1,\ \xi_2)\ ds_\xi \end{aligned}$$

Here $|ds_{x,y}|$ is the length of the edge and has already been computed by the code by finding the length based on the x, y coordinates of the nodes.

We thus need to compute $\int \phi_j(\xi_1,\ \xi_2)\ ds$ over all of the possible edges. Beginning with edge 1, we compute the integral.

$$\begin{aligned} \int_0^1 \phi_1\ d\xi_1 &= \int_0^1 1 - \xi_1\ d\xi_1 \\ &= \left(\xi_1 - \frac{\xi_1^2}{2}\right|_0^1 \\ &= 1 - 1/2 \\ &= 1/2 \end{aligned}$$

We note that the integral over edge 2 has the same form as that over edge 1, so $\int_0^1 \phi_1\ d\xi_2 = 1/2$. Additionally, $\int_0^1 \phi_2\ d\xi_1 = \int_0^1 \phi_3\ d\xi_2 = 1/2$. Over edge 3 we note that the area under the curve is 1/2 for both ϕ_2 and ϕ_3.

The results of the integration show that $\int hT_t\phi \ ds = 0.5HT_t|ds|$ for boundaries 2, 3, and 4 of the blade. Thus, in the code

$$F(I) = F(I) + 0.5Th|ds|$$

This leaves the M_{edge} matrix. This matrix is used to compute the addition of the Robin boundary to the k matrix. This contribution comes from the term in the bilinear form $\int wHT \ ds = \int \phi hT \ ds$.

We again compute this in the reference triangle after making the approximation that

$$\tilde{T} = \sum_{i=1}^{N} a_i\phi_i$$

Thus, we have

$$\begin{aligned} \int w_j hT \ ds &= \int \phi_j h \sum a_i\phi_i \ ds \\ K_{i,j} &= \int \phi_j h\phi_i \ ds_{x,y} \\ &= \int h\phi_j\phi_i |ds_{x,y}| ds_\xi \end{aligned}$$

We now compute all of the possible $\phi_i\phi_j$ combination. We begin with edge 1.

$$\begin{aligned} \int_0^1 \phi_1\phi_1 \ d\xi_1 &= \int_0^1 (1-\xi_1)(1-\xi_1)d\xi_1 \\ &= \int_0^1 (1-2\xi_i+\xi_1^2)d\xi_1 \\ &= (\xi_1 - \xi_1^2 + \frac{\xi_1^3}{3}|_0^1 \\ &= 1 - 1 - 1/3 \\ &= 1/3 \end{aligned}$$

$$\begin{aligned} \int_0^1 \phi_1\phi_2 \ d\xi_1 &= \int_0^1 (1-\xi_1)(\xi_1)d\xi_1 \\ &= \int_0^1 (\xi_1 - \xi_1^2)d\xi_1 \\ &= (\frac{\xi_1^2}{2} - \frac{\xi_1^3}{3}|_0^1 \\ &= \frac{1}{2} - \frac{1}{3} \\ &= 1/6 \end{aligned}$$

$$\begin{aligned}\int_0^1 \phi_2\phi_2 \; d\xi_1 &= \int_0^1 (\xi_1)(\xi_1)d\xi_1 \\ &= \int_0^1 \xi_1^2 \; d\xi_1 \\ &= (\frac{\xi_1^3}{3}|_0^1 \\ &= 1/3\end{aligned}$$

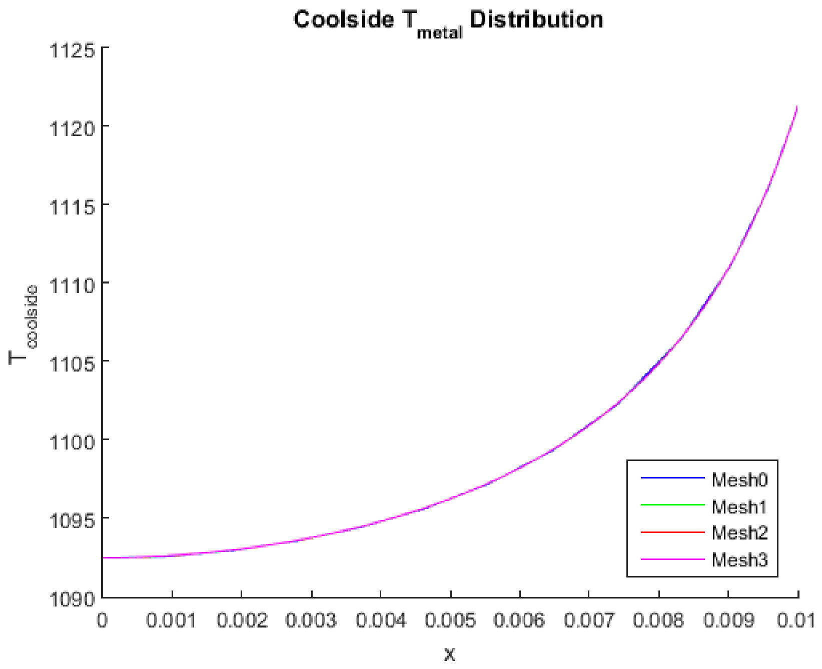
Over edges two and three of the element, we use the results computed over edge 1.

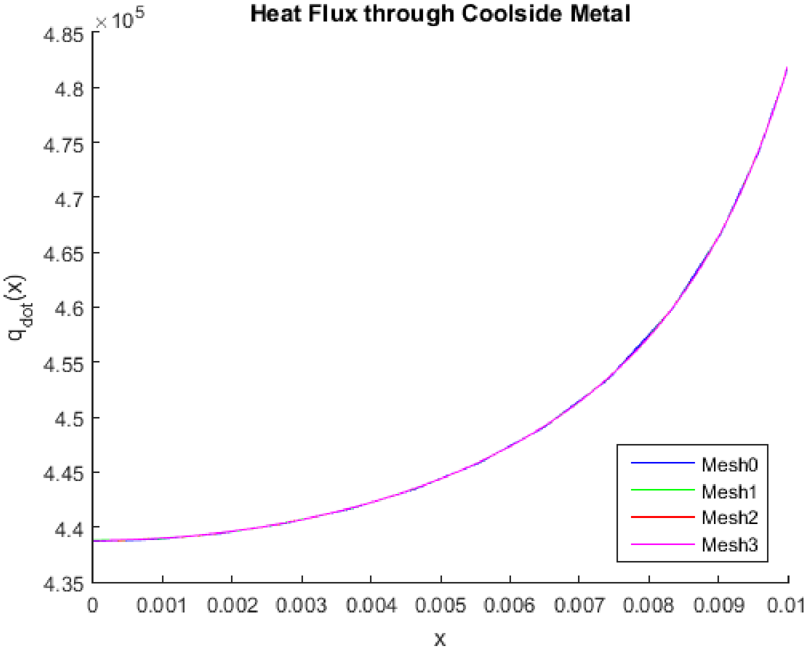
$$\begin{aligned}\int_0^1 \phi_1\phi_1 \; d\xi_2 &= 1/3 \\ \int_0^1 \phi_1\phi_3 \; d\xi_2 &= 1/6 \\ \int_0^1 \phi_3\phi_3 \; d\xi_2 &= 1/3 \\ \int_0^1 \phi_2\phi_3 \; d\xi_1 &= 1/6 \\ \int_0^1 \phi_2\phi_2 \; d\xi_1 &= 1/3 \\ \int_0^1 \phi_3\phi_3 \; d\xi_2 &= 1/3\end{aligned}$$

Thus, for all edges,

$$M_{edge} = \begin{bmatrix} \phi_i\phi_i & \phi_i\phi_j \\ \phi_j\phi_i & \phi_j\phi_j \end{bmatrix} = \begin{bmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{bmatrix}$$

We have now computed all of the required inputs to the FEM code and we are ready to run the simulation. We set $T_{t_{hot}} = 1400$K and $T_{t_{cool}} = 800$K along the entire length of boundary four. To determine $\dot{q}(x)$ we note that $\vec{q}\cdot\vec{n} = h(T - T_t)$. Thus, we can compute $\dot{q}(x)$ with $\dot{q}(x) = h(T_{metal} - T_{t_{cool}})$ where $h = 1500$ W/m^{2}K and $T_{t_{cool}} = 800$K. The temperature distribution of the coolside metal and the heat flux through the coolside metal are plotted below for all four meshes.

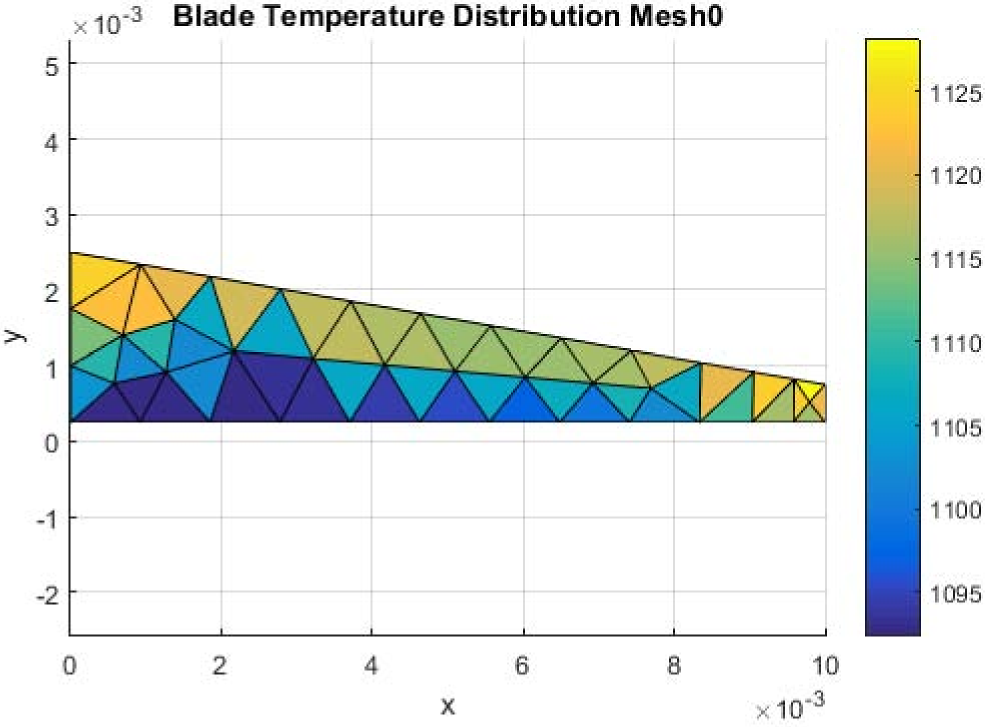




We see from the plots that both T_{metal} and $\dot{q}(x)$ exponentially increase as x increases. This makes sense because as x increases, the blade gets thinner, resulting in the metal having less space in which to diffuse the heat. This results in the tip getting hotter than the base, which increases the metal temperature and thus the heat flow into the cooling gas.

We see from the results that the mesh resolution smooths out the curve as the number of elements increases. However, the trend is the same and the error is obviously negligible from the results. Thus, the answer barely changes with increasing mesh resolution.

The temperature distribution from Mesh 0 is plotted below. It is interesting to note that the low resolution causes the large spread in temperature values. However, as noted previously, along the boundary this low resolution barely changes the resulting temperature along boundary four.



3 Thermal Fluid Coupling

3.1 Problem 8

The first task of the coupling is to solve the system with the $\dot{q}$ from part 2. We recall from part 1 of this project the implementation of the finite volume scheme using upwinding flux. From the FEM script we obtain the temperatures at the nodes of the mesh that are on the coolside boundary. These $N+1$ nodal values can be used along with the $T_{coolside}$ distribution that was assumed in the FEM calculation to obtain $\dot{q}(x)$ at the nodes. We next recall from part 1 that

$$T_i - T_{i-1} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \dot{q}(x)dx$$

Since we have $\dot{q}(x)$, we numerically integrate using a spacial integration scheme. For this particular situation we choose a trapezoidal integration scheme. This choice was made for two reasons. First, we know from prior work that the trapezoidal integration scheme is accurate to $\mathcal{O}(\Delta x^2)$. Second, we know that the values of the mesh elements are piecewise linear, as this is the key approximation made by the finite element method. The trapezoidal integration similarly linearly interpolates the solution by integrating the trapezoidal area under the curve. Thus, we use the following process.

$$dx_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$$

Also, note that $q = 2q$ since there are two sides of the blade contributing to the heating of the cooling air. The trapezoidal integration is given by:

$$Q_i = \frac{\Delta x(q_{i+\frac{1}{2}} q_{i-\frac{1}{2}})}{2}$$

We now solve the $N \times N$ system

$$AT = F$$

$$A = \begin{bmatrix} 1 & 0 & \cdots & & & \\ -1 & 1 & 0 & \cdots & & \\ 0 & -1 & 1 & 0 & \cdots & \\ & \ddots & \ddots & \ddots & \ddots & \end{bmatrix}$$

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}$$

$$F = \frac{1}{c_p \dot{m}} \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{bmatrix}$$

Here T_i are the values of the temperature in the N control volumes. The values of Q_i are computed using trapezoidal integration as noted above. We now solve the system

$$T = A \setminus F$$

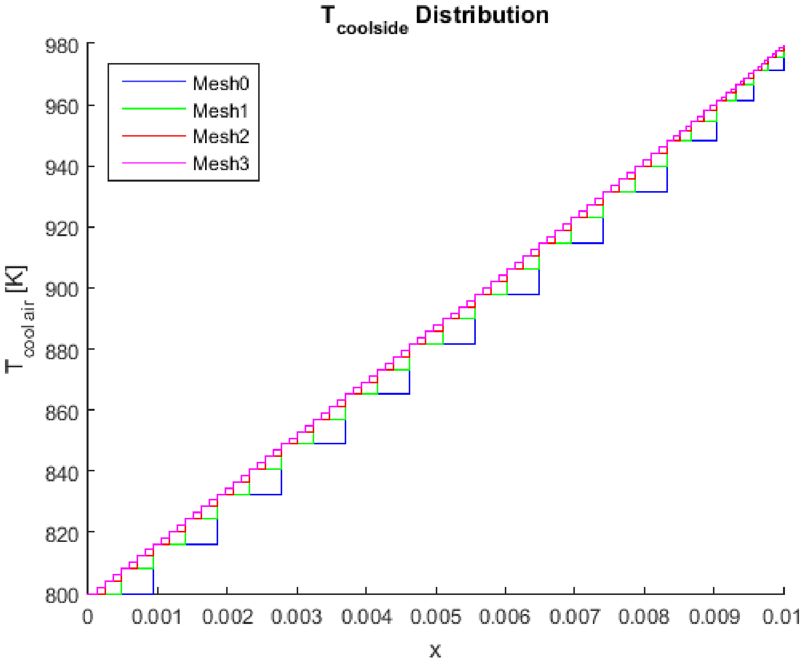
This returns N temperatures corresponding to the N control volumes.

Back in the main script, we now determine the new $\dot{q}(x)$ distribution. Here we must make the upwinding flux approximation. This states that $F_{i+\frac{1}{2}} F_i$. Thus, $T_{i+\frac{1}{2}} T_i$. This means that when computing the $\dot{q}(x)$, we take the vector of $N+1$ nodes from the FEM function and subtract

$$T_{cool} = \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}$$

Where $T_0 = 800\text{K}$ and $T_1 ... T_N$ are the values of the N control volumes from the finite volume method.

The temperature of the coolside for all four meshes are plotted below.

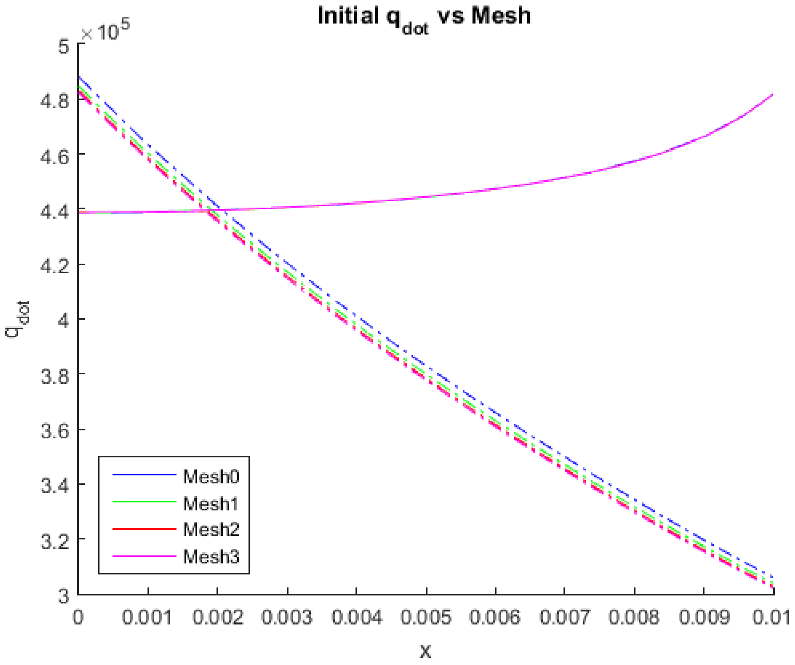


The values of T at the nodes can be seen on the left hand side of each step in the stair function. Due to plotting issues, to see the temperature of the control volumes, we must shift each stair one control volume to the left. The nodal values are correct, however. We see that the temperature distribution appears to be linear.

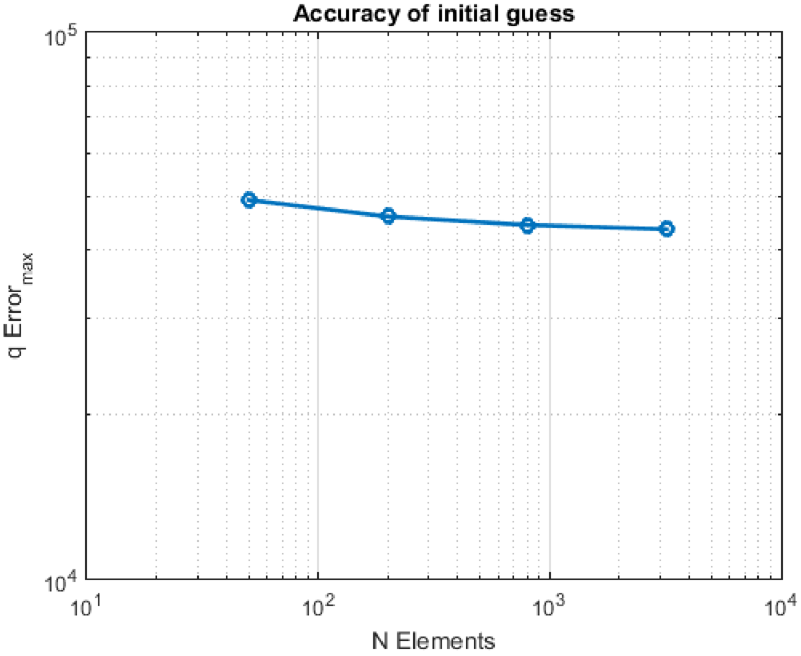
As a side note, for the next part Mesh 3, the highest resolution mesh, was used to compute the distribution. This was done to minimize error. As notes in task 7, the choice of mesh does not actually matter much since the error is for all practical purposes negligible. Thus, the choice of mesh only really affect computation time. Since there was plenty of time for this project, Mesh 3 was used.

3.2

We now compute the new $\dot{q}(x)$ by putting our N temperatures back into the FEM code. The results are plotted below. The solid lines are the initial $\dot{q}(x)$ computed by assuming that T_0 was constant along the entire coolside boundary. The dashed lines were computed by putting the N values of T back into the FEM code, finding the new metal temperature, and then recomputing $\dot{q}(x) = h(T_{metal} - T_{t_{cool}})$. It can be clearly seen that there is a large difference in the results from the two runs. This indicates that assuming T_0 is constant is not a good approximation.



We next compute the absolute error of the difference between $\dot{q}_1(x)$ and $\dot{q}_0(x)$. A plot of this error is below.



The plot shows that the maximum error is almost 16000 W/m^2 between the two trials. However, it is interesting to note that the error barely changes with increases in the mesh resolution. This agrees with the trends previously notices in tasks 7 and 8 where the different meshes hardly changed the temperature distributions.

The large difference between the $\dot{q}(x)$ computed using T_0 = constant and $T_{coolside}$ as output by the FVM code are striking. The large error shows that assuming a constant T_0 is very unrealistic. Encouragingly, the results from the FVM generated temperature distribution appear to be much more accurate, and thus our choice of linking these two models appears to be justified.

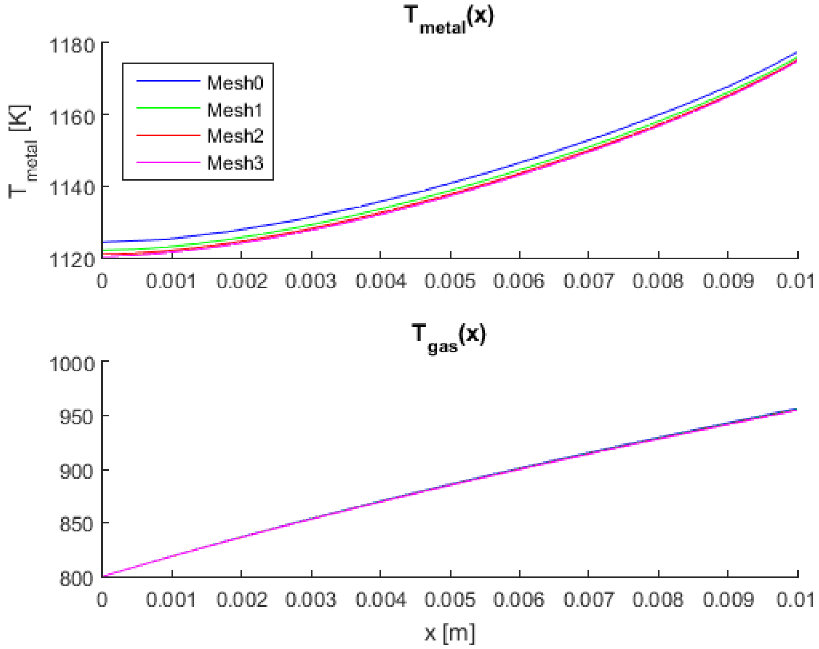
3.3

We now force the code to converge the $\dot{q}$ distributions. We first compute $\dot{q}(x)$ given a constant temperature of the collside gas. We then sue these in the finite volume code to get the new coolside temperature distribution. This new distribution is fed back into the FEM code, and $\dot{q}(x)$ is recomputed. This process is repeated until the error is 1×10^-3W/m^2. The error is defined by

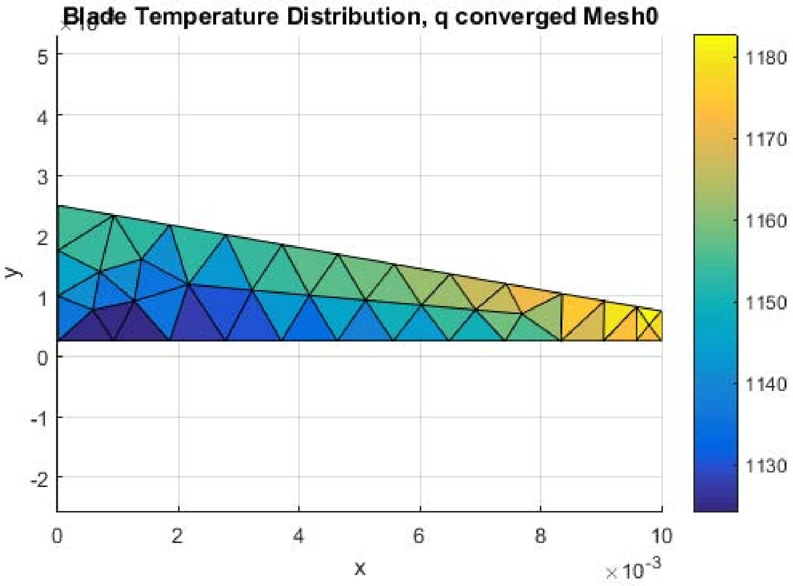
$$error = \max(|q_i - q_{i-1}|)$$

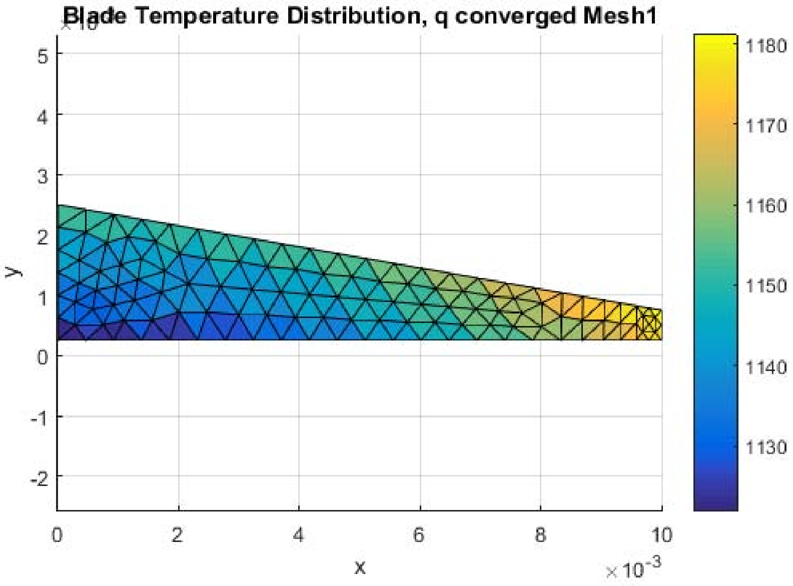
Note that both q are vectors of dimension $N + 1$.

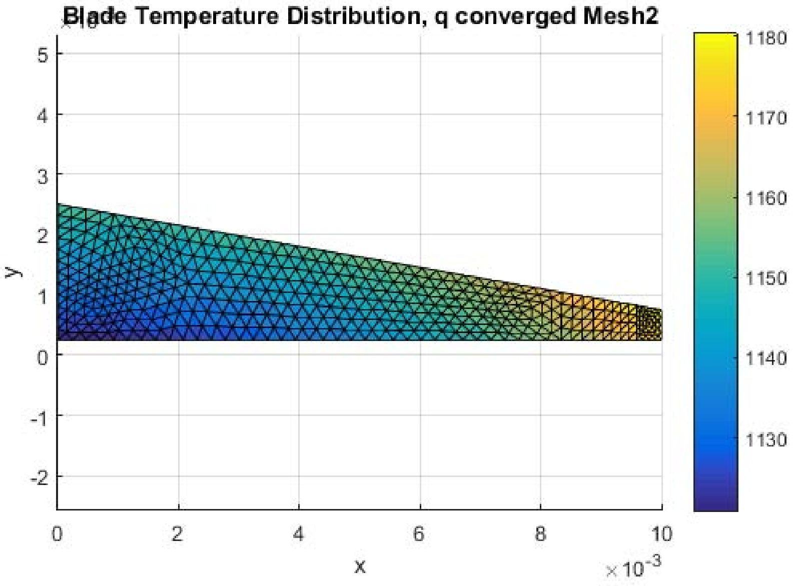
The metal and gas temperature distributions for the converged models are plotted below. We see again that T_{metal} increases exponentially, while T_{gas} appears to increase relatively linearly over the mesh.

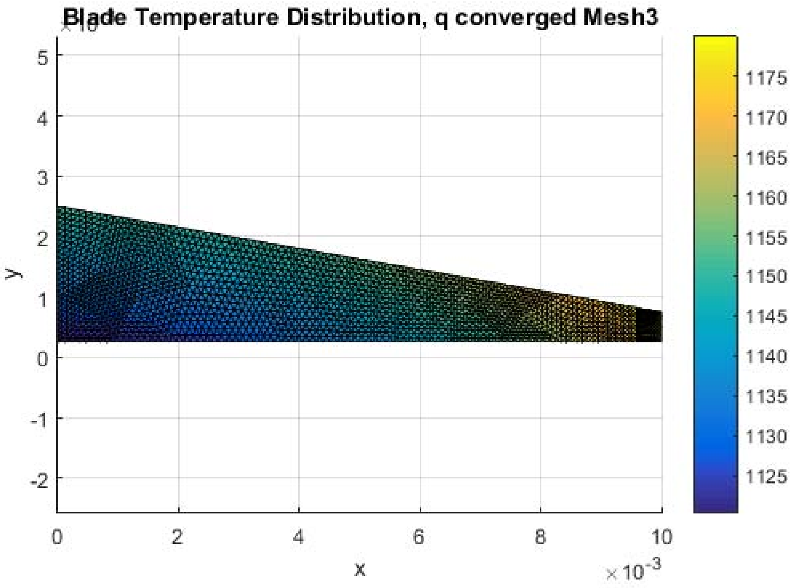


The blade temperature distribution plots are shown below.



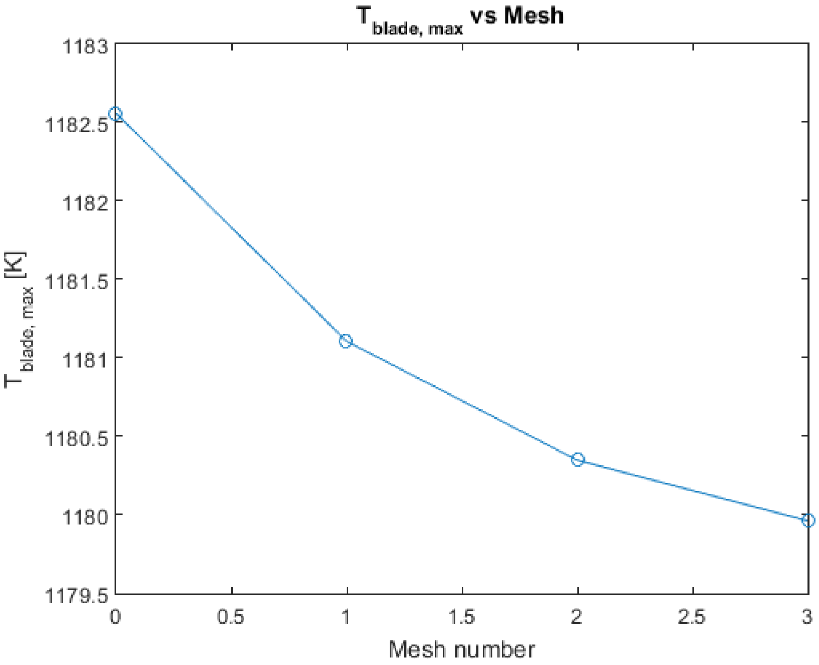

Blade Temperature Distribution, q converged Mesh1
×10⁻³
y
5
4
3
2
1
0
-1
-2
x
0
2
4
6
8
10
×10⁻³
1180
1170
1160
1150
1140
1130


Blade Temperature Distribution, q converged Mesh2
×10⁻³
y
5
4
3
2
1
0
-1
-2
x
0
2
4
6
8
10
×10⁻³
1180
1170
1160
1150
1140
1130



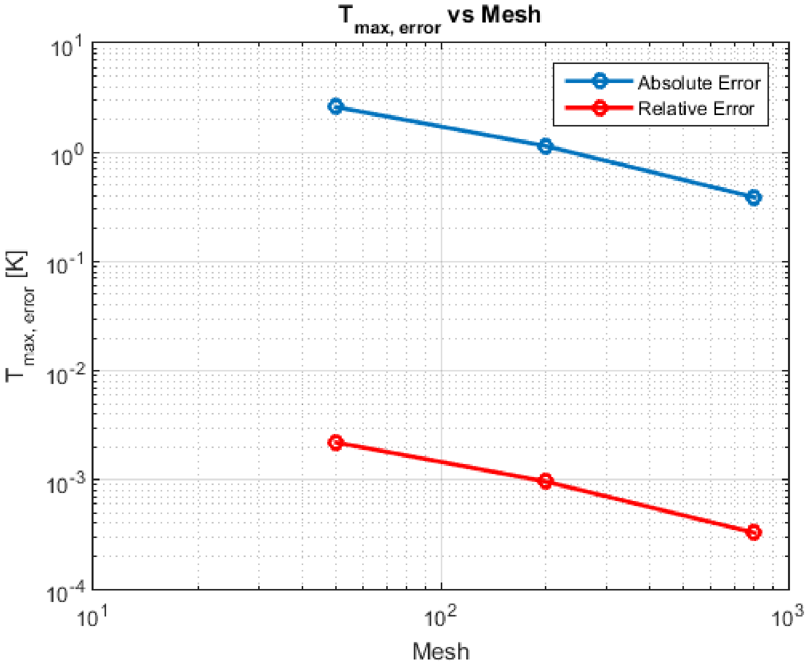
The temperature distributions show the same general trends no matter the resolution of the mesh. Increases in resolution result in the temperature fronts being more clearly defined, but the distribution is the same.

The maximum blade temperatures for all four meshes are plotted below.



This plot shows that as the resolution increases, the maximum temperature falls. However, by looking more carefully at the maximum temperature plot, we see that the temperature only drops by 2.5K from the coarsest to the finest mesh. This is 0.2% of the 1180K value of T_{max} in the most resolved mesh.

The accuracy plot, below, corroborates these results.



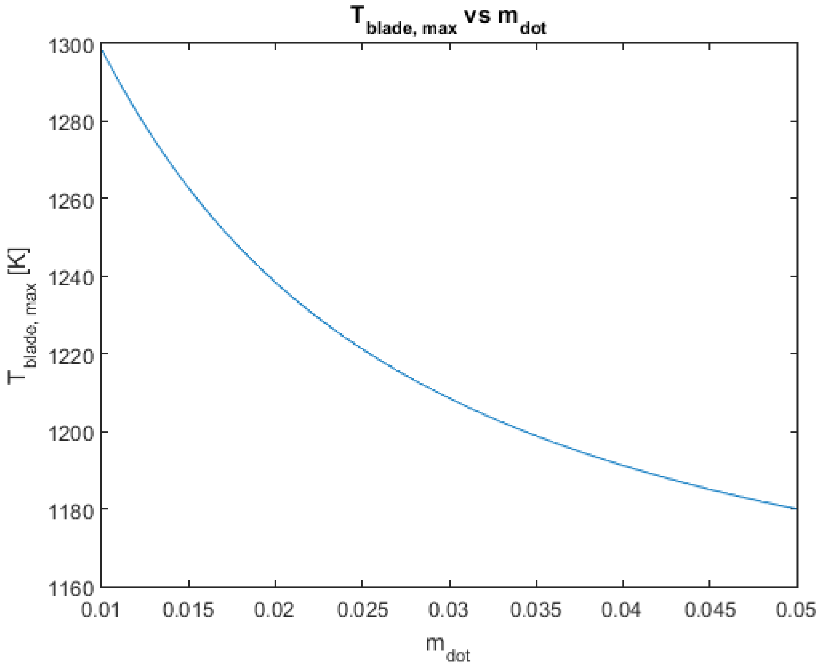
The error is approximately of $\mathcal{O}(N_e^1)$. These errors were computed by comparing the results of the three coarser meshes to the finest mesh, Mesh 3. The absolute error is $T_{max} - T_{max,3}$ and the relative error is defined by

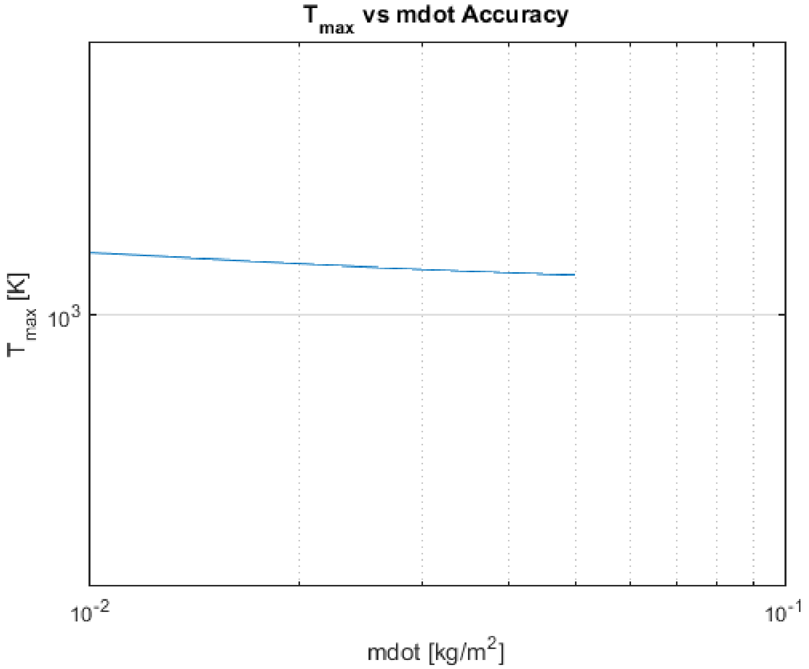
$$error_{rel} = \frac{T_{max} - T_{max,3}}{T_{max,3}} \times 100$$

We therefore see that the maximum error is on the order of $10^{-3}\%$ or approximately $8 \times 10^{-4}\%$. This is an absolutely negligible error. Thus, the choice of mesh does not matter if all that is desired is the maximum temperature in the blade. This is an important point. The results show that the choice of mesh does not significantly impact the maximum interior blade temperature. This is the power of the FEM method. Even for low resolution meshes, the scheme is accurate enough to correctly predict the behavior of the system, for this scenario.

3.4

We now examine the effect of changing the mass flow rate of the cooling air over boundary 4 on the maximum blade temperature. The results are plotted below.





The results show that as $\dot{m}$ increases, the maximum temperature in the blade decreases exponentially. The order of accuracy appears to be of order $\mathcal{O}(\dot{m}^1)$. This is tricky to see since for an increase in $\dot{m}$ of 0.04 kg/m the temperature only decreases by about 120K, or about 10%. We can compute the accuracy by hand by comparing the maximum and minimum mass flow rates to the maximum blade temperature for each flow rate.

$$\frac{1300}{1180} \rightarrow \frac{0.01}{0.05}$$
$$1.10 \rightarrow 0.2$$

Thus, taking the log base 10 of each value, 10^{-1} is compared to $10^{0.7}$, which is approximately a 10^{-1} decrease in the temperature for approximately 10^{1} increase in the mass flow rate of the cool air. For a 5-fold increase in mass flow, we get a 10% reduction in T_{max}. This means that the order of accuracy of T_{max} with respect to $\dot{m}$ is approximately $\mathcal{O}(\dot{m}^1)$. Thus, the maximum temperature decreases with increasing $\dot{m}$ by about $\mathcal{O}(\dot{m}^1)$. This is expected since increased cooling air allows for more heat to be transferred into the cooling flow from the blade, thereby reducing the maximum blade temperature.

16.90 Computational Methods in Aerospace Engineering
Spring 2014