Finite Element Method for The Heat equation

Andres Chaves(), Diego Montufar()

October 13, 2014

Abstract

The abstract text goes here.

1 Introduction

The finite-element method provides an alternative that is better suited for complex systems. In contrast to finite-difference techniques, the finite-element method divides the solution domain into simply shaped regions, or "elements". An approximate solution for the PDE can be developed for each of these elements. The total solution is then generated by linking together, or "assembling", the individual solutions taking care to ensure continuity at the interelement boundaries. Thus, the PDE is satisfied in a piecewise fashion.
Although the particulars will vary, the implementation of the finite-element approach usually follows a standard step-by-step procedure.
1. Discretization 2. Element Equations 3. Assembly

2 Problem Statement

The heat equation can be used to describe the temperature variation through a solid body. The equation derives from the principle of conservation of energy and and gives rise to a second order linear PDE of the form:

$$\rho C \frac{\partial T}{\partial t} = k \nabla^2 T \tag{1}$$

Where T is the temperature $[K]$,ρ is the mass density $[kgm^3]$, C is the specific heat capacity $[Jkg^{-1}K^{-1}]$, and k is the thermal conductivity $[Wm^{-1}K^{-1}]$. For this particular implementation the computational domain will describe cooling fins on top of a central processing unit and will be defined by an unstructured tetrahedral grid.

We assumed that the cooling fins are made from pure copper, having the properties $\rho = 8954kgm^3$, $C = 380Jkg^{-1}K^{-1}$, and $k = 386Wm^{-1}K^{-1}$.
In this particular case the CPU creates a constant heat flux Q_{cpu} , given by:

$$Q_{cpu} = k\nabla T \tag{2}$$

which defines a Neumann boundary condition on the base of the domain and $Q_{cpu} = 40,000Wm^{-2}$. For the remaining boundary of the cooling fins we assumed a convection boundary condition, given by:

$$Q_{air} = k\nabla T = h(T - T_{air}) \tag{3}$$

which defines a Robin boundary condition. Here we assumed that the convection coefficient $h = 100W/m^2K^{-1}$ and the ambient temperature $T_{air} = 300K$. Defining the initial temperature of the cooling fins as ambient, the main goal is to simulate the temperature rise in the domain $t \in [0, 100]s$ and determine the maximum steady state temperature of the cooling fins.

3 Governing Equations

In order to solve the Heat Equation using the Finite Element method, we are going to start by deriving the weak form of Equation (1). Applying this numerical method implies a series of steps until we get a global system of equations that must be assembled in order to compute the solution. So we are going to follow the next steps:

1. Derive the weak form of the Heat Equation by applying the Galerking Weighted residual method.

2. Express our field variable T and its derivatives in terms of the shape functions for a linear tetrahedral element.

3. Assemble the global system of equations.

3.1 Galerking Weighted residual method

Weighted residual methods assume that a solution can be approximated analytically or piecewise analytically. The idea of this method is that we can multiply the residual by a weighting function and force the integral of the weighted expression over the domain to vanish:

$$\int_{\Omega} W(x)r(x)d\Omega = 0 \tag{4}$$

We understand the residual as a continuous function of space. In the Galerking method the weight functions are chosen to be identical to the trial functions where the trial function is our assumed approximate solution for the PDE which takes the form:

$$\phi(x) = \sum_{n=0}^{N} a_n(x)p_n(x), \quad \text{Where} \quad W_n(x) = p_n(x) \tag{5}$$

Applying equation (4) to the Heat Equation (1) we have:

$$r = \rho C\frac{\partial T}{\partial t} - k\nabla^2 T$$

$$\int_{\Omega} W(\rho C\frac{\partial T}{\partial t} - k\nabla^2 T)d\Omega = 0$$

This is an appropriate time to include the convection term into our equation and expanding terms we get:

$$\int_{\Omega} W\rho C\frac{\partial T}{\partial t}d\Omega - \int_{\Omega} Wk\nabla^2 T d\Omega + \int_{\Omega} WQ_{air}d\Omega = 0 \tag{6}$$

We can modify the diffusive second order term by applying the product rule for derivatives which is defined by:

$$\begin{aligned}\nabla \cdot (W\nabla T) &= \nabla W \cdot \nabla T + W\nabla \cdot \nabla T \\ &= \nabla W \cdot \nabla T + W\nabla^2 T\end{aligned}$$

In our second term of Equation (6): Applying the product rule, rearranging, and integrating over the domain we get:

$$\int_{\Omega} Wk\nabla^2 T d\Omega = \int_{\Omega} \nabla \cdot (Wk\nabla T)d\Omega - \int_{\Omega} \nabla W \cdot k\nabla T d\Omega \tag{7}$$

in order to eliminate second order terms of the Equation (7), another important concept we are going to use is the theorem of divergence which can be expressed as follows:

$$\int_{\Omega} \nabla^2 2T d\Omega = \int_{\Gamma} \nabla T \cdot d\Gamma$$

Now, we can make use of the Divergence Theorem and apply it to the second order term of Equation (7) as follows:

$$\int_{\Omega} kW\nabla^2 T d\Omega = \int_{\Gamma} kW\nabla T \cdot d\mathbf{\Gamma} - \int_{\Omega} k\nabla W \cdot \nabla T d\Omega \tag{8}$$

Substitute Equation (8) in (6) we get:

$$\int_{\Omega} \rho CW\frac{\partial T}{\partial t}d\Omega - \int_{\Gamma_{Base}} kW\nabla T \cdot d\mathbf{\Gamma} - \int_{\Omega} k\nabla W \cdot \nabla T d\Omega + \int_{\Gamma_{Fins}} WQ_{air}d\Gamma = 0 \tag{9}$$

The important point here is that applying the Galerkin method of weighted residuals this way means that for every element we get a 'sub' system of equations local to each element and in terms of the local nodal values $1...N_n$, that must be assembled into a global system of equations with global indices $1...N_p$
Then we can see the second term in the equation corresponds to the value of Q_{cpu} as stated in the model problem. In contrast to other methods, the Neumann boundary conditions are automatically incorporated into the integral form of the PDE. Rearranging terms we get the Weak form of the Heat Equation:

$$\int_{\Omega} \rho CW\frac{\partial T}{\partial t}d\Omega + \int_{\Omega} k\nabla W \cdot \nabla T d\Omega = \int_{\Gamma_{Base}} kW\nabla T \cdot d\mathbf{\Gamma} - \int_{\Gamma_{Fins}} Wh(T - T_{air})d\Gamma \tag{10}$$

The overall global system of equations to solve our generic scalar transport equation is then given by:

$$\sum_{n=1}^{N_p}\int_{\Omega} (W\rho C\frac{\partial T}{\partial t} + \nabla W \cdot k\nabla T)d\Omega = \sum_{n=1}^{N_p}\int_{\Gamma_{Base}} Wk\nabla T \cdot d\mathbf{\Gamma} - \sum_{n=1}^{N_p}\int_{\Gamma_{Fins}} Wh(T - T_{air})d\Gamma \tag{11}$$

3.2 Shape functions

One of the fundamental steps in a finite element analysis is the discretization of a continuous body containing infinite number of points in the surface into a discrete model with a limited number of points (or nodes) in the surface. The shape of the body between these nodes its approximated by functions. These functions are known as shape functions, and allow us to relate the coordinates of every point of a finite element with the positions of its nodes.

Thinking of T most generally as being a continuous function of space and time, we will write the assumed solution in the form:

$$T(\mathbf{x},t) = \sum_{n=1}^{N_n} \eta_n(\mathbf{x}) T_n(t) \tag{12}$$

Where η_n are known as the *shape functions*, which are functions of spatial location only, while the nodal values of T are functions of time only.
If we now substitute in our assumed form of the solution from Equation (10), and consider the domain of integration to be the domain of an element itself, the weighted residual expression can be rewritten using Einstein summation notation as:

$$\int_{\Omega_e} \eta_p \eta_q \rho C \frac{\partial T}{\partial t} d\Omega + \int_{\Omega_e} k \nabla \eta_p \cdot \nabla \eta_q T_q d\Omega = \int_{\Gamma_e} k \eta_p \nabla T \cdot d\mathbf{\Gamma} - \int_{\Gamma_e} h \eta_p (\eta_q T_q - T_e) d\Gamma \tag{13}$$

Then the overall System of Equations to solve the Heat Equation is the given by:

$$\sum_{e=1}^{N_e} \int_{\Omega_e} \left(\eta_p \eta_q \rho C \frac{\partial T_q}{\partial t} d\Omega + k \nabla \eta_p \cdot \nabla \eta_q T_q \right) d\Omega = \sum_{e=1}^{N_e} \int_{\Gamma_e} k \eta_p \nabla T \cdot d\mathbf{\Gamma} - \sum_{e=1}^{N_e} \int_{\Gamma_e} \eta_p h (\eta_q T_q - T_{air}) d\Gamma \tag{14}$$

We can then rewrite the system of equations in the form:

$$M\dot{T} = KT + s \tag{15}$$

Where M represents the Mass matrix, K the Stiffness Matrix, both have size of N_pxN_p and finally the s term that is the load vector with size N_px1. For the Heat Equation we can recognize each term of Equation(15) in the Equation (13) as follows:

$$M^e_{pq} = \int_{\Omega_e} \eta_p \eta_q \rho C d\Omega \tag{16}$$

$$K^e_{pq} = -\int_{\Omega_e} k \nabla \eta_p \cdot \nabla \eta_q d\Omega - h \int_{\Gamma_e} \eta_p \eta_q d\Gamma \tag{17}$$

$$S^e_p = h T_{air} \int_{\Gamma_e} \eta_p d\Gamma + k \int_{\Gamma_e} \eta_p \nabla T \cdot d\mathbf{\Gamma} \tag{18}$$

For this case we are going to use tetrahedral elements, which means we are going to have 4 nodes in a 3D element as shown in Figure(). In order to derive the corresponding shape functions we begin by assuming a trial solution which takes the form:

$$T(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z \tag{19}$$

Where a_0,a_1,a_2 and a_3 are scalar coefficients. This is essentially the 3D equivalent of the trial solution to a triangular element. If we apply this trial solution in every node of the element we'll get:

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{bmatrix} \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \\ \mathbf{p_4} \end{bmatrix} \mathbf{a} = C\mathbf{a} \tag{20}$$

Where $p_n = \{1\, x_n\, y_n\, z_n\}$, C will be a square 4x4 matrix and we can solve for the unknown parameters by computing $\mathbf{a} = C^{-1}\mathbf{T}$. Doing so we can find our shape functions by substituting these coefficients back into our trial solution, So we can factor out the nodal values and rewrite the solution in the form:

$$T(x, y, z) = \sum_{n=1}^{N_n} \eta_n(x, y, z) T_n = \eta_1 T_1 + \eta_2 T_2 + \eta_3 T_3 + \eta_4 T_4 \tag{21}$$

Where the shape functions for the linear tetrahedron are:

$$
\begin{aligned}
\eta_1(x,y,z) &= \frac{1}{6V_e}\Big(x_2\left(y_3z_4 - y_4z_3\right) + x_3\left(y_4z_2 - y_2z_4\right) + x_4\left(y_2z_3 - y_3z_2\right) \\
&+ \left((y_4 - y_2)(z_3 - z_2) - (y_3 - y_2)(z_4 - z_2)\right)x \\
&+ \left((x_3 - x_2)(z_4 - z_2) - (x_4 - x_2)(z_3 - z_2)\right)y \\
&+ \left((x_4 - x_2)(y_3 - y_2) - (x_3 - x_2)(y_4 - y_2)\right)z\Big) \\
\eta_2(x,y,z) &= \frac{1}{6V_e}\Big(x_1\left(y_4z_3 - y_3z_4\right) + x_3\left(y_1z_4 - y_4z_1\right) + x_4\left(y_3z_1 - y_1z_3\right) \\
&+ \left((y_3 - y_1)(z_4 - z_3) - (y_3 - y_4)(z_1 - z_3)\right)x \\
&+ \left((x_4 - x_3)(z_3 - z_1) - (x_1 - x_3)(z_3 - z_4)\right)y \\
&+ \left((x_3 - x_1)(y_4 - y_3) - (x_3 - x_4)(y_1 - y_3)\right)z\Big) \\
\eta_3(x,y,z) &= \frac{1}{6V_e}\Big(x_1\left(y_2z_4 - y_4z_2\right) + x_2\left(y_4z_1 - y_1z_4\right) + x_4\left(y_1z_2 - y_2z_1\right) \\
&+ \left((y_2 - y_4)(z_1 - z_4) - (y_1 - y_4)(z_2 - z_4)\right)x \\
&+ \left((x_1 - x_4)(z_2 - z_4) - (x_2 - x_4)(z_1 - z_4)\right)y \\
&+ \left((x_2 - x_4)(y_1 - y_4) - (x_1 - x_4)(y_2 - y_4)\right)z\Big) \\
\eta_4(x,y,z) &= \frac{1}{6V_e}\Big(x_1\left(y_3z_2 - y_2z_3\right) + x_2\left(y_1z_3 - y_3z_1\right) + x_3\left(y_2z_1 - y_1z_2\right) \\
&+ \left((y_1 - y_3)(z_2 - z_1) - (y_1 - y_2)(z_3 - z_1)\right)x \\
&+ \left((x_2 - x_1)(z_1 - z_3) - (x_3 - x_1)(z_1 - z_2)\right)y \\
&+ \left((x_1 - x_3)(y_2 - y_1) - (x_1 - x_2)(y_3 - y_1)\right)z\Big)
\end{aligned}
$$

Where V_e is the volume of the element and is defined in terms of its nodal coordinates as:

$$
6V_e = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \tag{22}
$$

And we could express the derivatives of the shape functions as:

$$
\begin{aligned}
\frac{\partial \eta_1(x,y,z)}{\partial x} &= \frac{1}{6V_e}\left((y_4-y_2)(z_3-z_2)-(y_3-y_2)(z_4-z_2)\right)\\
\frac{\partial \eta_1(x,y,z)}{\partial y} &= \frac{1}{6V_e}\left((x_3-x_2)(z_4-z_2)-(x_4-x_2)(z_3-z_2)\right)\\
\frac{\partial \eta_1(x,y,z)}{\partial z} &= \frac{1}{6V_e}\left((x_4-x_2)(y_3-y_2)-(x_3-x_2)(y_4-y_2)\right)\\
\frac{\partial \eta_2(x,y,z)}{\partial x} &= \frac{1}{6V_e}\left((y_3-y_1)(z_4-z_3)-(y_3-y_4)(z_1-z_3)\right)\\
\frac{\partial \eta_2(x,y,z)}{\partial y} &= \frac{1}{6V_e}\left((x_4-x_3)(z_3-z_1)-(x_1-x_3)(z_3-z_4)\right)\\
\frac{\partial \eta_2(x,y,z)}{\partial z} &= \frac{1}{6V_e}\left((x_3-x_1)(y_4-y_3)-(x_3-x_4)(y_1-y_3)\right)\\
\frac{\partial \eta_3(x,y,z)}{\partial x} &= \frac{1}{6V_e}\left((y_2-y_4)(z_1-z_4)-(y_1-y_4)(z_2-z_4)\right)\\
\frac{\partial \eta_3(x,y,z)}{\partial y} &= \frac{1}{6V_e}\left((x_1-x_4)(z_2-z_4)-(x_2-x_4)(z_1-z_4)\right)\\
\frac{\partial \eta_3(x,y,z)}{\partial z} &= \frac{1}{6V_e}\left((x_2-x_4)(y_1-y_4)-(x_1-x_4)(y_2-y_4)\right)\\
\frac{\partial \eta_4(x,y,z)}{\partial x} &= \frac{1}{6V_e}\left((y_1-y_3)(z_2-z_1)-(y_1-y_2)(z_3-z_1)\right)\\
\frac{\partial \eta_4(x,y,z)}{\partial y} &= \frac{1}{6V_e}\left((x_2-x_1)(z_1-z_3)-(x_3-x_1)(z_1-z_2)\right)\\
\frac{\partial \eta_4(x,y,z)}{\partial z} &= \frac{1}{6V_e}\left((x_1-x_3)(y_2-y_1)-(x_1-x_2)(y_3-y_1)\right)
\end{aligned}
$$

Furthermore, we have the integration formulae defined for the linear tetrahedron:

$$\int_{\Omega_e} \eta_p^a \eta_q^b \eta_r^c \eta_s^d d\Omega = \frac{a!b!c!d!6\Omega_e}{(a+b+c+d+3)!} \tag{23}$$

and:

$$\int_{\Gamma_e} \eta_p^a \eta_q^b \eta_r^c d\Omega = \frac{a!b!c!2\Gamma_e}{(a+b+c+2)!} \tag{24}$$

Where for the 3D case Ω_e and Γ_e represent the volume and face area of a tetrahedron respectively.
Let's first consider the integration of the shape functions as required in the mass matrix from Equation (16):

$$M^e_{p,q} = \rho C \int_{\Omega_e} \eta_p \eta_q d\Omega$$

Remembering that p and q are in the range of 1 to 4 for the linear tetrahedral element, what we end up with is a local or 'sub' matrix M_e, which for our linear tetrahedral element will be a 4 x 4 matrix. In order to evaluate each term in the matrix we simply input the values of p and q to the integration formula in Equation (23). For the case where p and q are equal (i.e. for elements on the main diagonal) we get:

$$\begin{aligned} M^e_{p,p} &= \rho C \int_{\Omega_e} \eta_p \eta_p d\Omega \\ &= \rho C \int_{\Omega_e} \eta_p^2 \eta_q^0 \eta_r^0 \eta_s^0 d\Omega = \frac{\rho C 2!0!0!0!6\Omega_e}{(2+0+0+0+3)!} \\ &= \frac{2\rho C \Omega_e}{20} \end{aligned}$$

For the case where p and q are not equal (i.e. for elements off the main diagonal) we get:

$$\begin{aligned} M^e_{p,q} &= \rho C \int_{\Omega_e} \eta_p \eta_q d\Omega \\ &= \rho C \int_{\Omega_e} \eta_p^1 \eta_q^1 \eta_r^0 \eta_s^0 d\Omega = \frac{\rho C 1!1!0!0!6\Omega_e}{(1+1+0+0+3)!} \\ &= \frac{\rho C \Omega_e}{20} \end{aligned}$$

So we can write a single expression for the any element in our local mass matrix as:

$$M^e_{p,q} = \frac{\rho C (1 + \delta_{pq})}{20}$$

which is simple enough that we could write out the whole thing as:

$$M^e = \frac{\rho C \Omega_e}{20} \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \tag{25}$$

Considering now the contribution of the diffusive term to the stiffness matrix, and writing shape function derivative terms in the form of a column vector with modulo 3 notation for compactness we have:

$$-k\int_{\Omega_e}\nabla\eta_p\cdot\nabla\eta_q d\Omega =$$

$$= -k\int_{\Omega_e}\frac{1}{6\Omega_e}\begin{Bmatrix}(y_{p+3}-y_{p+1})(z_{p+2}-z_{p+1})-(y_{p+2}-y_{p+1})(z_{p+3}-z_{p+1})\\(x_{p+2}-x_{p+1})(z_{p+3}-z_{p+1})-(x_{p+3}-x_{p+1})(z_{p+2}-z_{p+1})\\(x_{p+3}-x_{p+1})(y_{p+2}-y_{p+1})-(x_{p+2}-x_{p+1})(y_{p+3}-y_{p+1})\\\cdots\end{Bmatrix}$$

$$.\frac{1}{6\Omega_e}\begin{Bmatrix}(y_{q+3}-y_{q+1})(z_{q+2}-z_{q+1})-(y_{q+2}-y_{q+1})(z_{q+3}-z_{q+1})\\(x_{q+2}-x_{q+1})(z_{q+3}-z_{q+1})-(x_{q+3}-x_{q+1})(z_{q+2}-z_{q+1})\\(x_{q+3}-x_{q+1})(y_{q+2}-y_{q+1})-(x_{q+2}-x_{q+1})(y_{q+3}-y_{q+1})\\\cdots\end{Bmatrix}$$

Performing the Integration we get:

$$-k\int_{\Omega_e}\nabla\eta_p\cdot\nabla\eta_q d\Omega =$$

$$-\frac{k}{36\Omega_e^2}\begin{Bmatrix}(y_{p+3}-y_{p+1})(z_{p+2}-z_{p+1})-(y_{p+2}-y_{p+1})(z_{p+3}-z_{p+1})\\(x_{p+2}-x_{p+1})(z_{p+3}-z_{p+1})-(x_{p+3}-x_{p+1})(z_{p+2}-z_{p+1})\\(x_{p+3}-x_{p+1})(y_{p+2}-y_{p+1})-(x_{p+2}-x_{p+1})(y_{p+3}-y_{p+1})\\\cdots\end{Bmatrix}$$

$$.\begin{Bmatrix}(y_{q+3}-y_{q+1})(z_{q+2}-z_{q+1})-(y_{q+2}-y_{q+1})(z_{q+3}-z_{q+1})\\(x_{q+2}-x_{q+1})(z_{q+3}-z_{q+1})-(x_{q+3}-x_{q+1})(z_{q+2}-z_{q+1})\\(x_{q+3}-x_{q+1})(y_{q+2}-y_{q+1})-(x_{q+2}-x_{q+1})(y_{q+3}-y_{q+1})\\\cdots\end{Bmatrix}\Omega_e \tag{26}$$

The important point to note is that the dot product between these two vectors produce a scalar value. Now we obtain the second part of the contribution to the stiffness matrix again first evaluating the integral for the main diagonal where p=q:

$$
\begin{aligned}
-h \int_{\Gamma_e} \eta_p \eta_p d\Gamma &= -h \int_{\Gamma_e} \eta_p^2 \eta_q^0 \eta_r^0 d\Gamma \\
&= -\frac{h2!0!0!2\Gamma_e}{(2+0+0+2)!} \\
&= -\frac{2h\Gamma_e}{12}
\end{aligned}
$$

Then we do the same for the off diagonal terms where p $\neq$ q:

$$
\begin{aligned}
-h \int_{\Gamma_e} \eta_p \eta_q d\Gamma &= -h \int_{\Gamma_e} \eta_p^1 \eta_q^1 \eta_r^0 d\Gamma \\
&= -\frac{h1!1!0!2\Gamma_e}{(1+1+0+2)!} \\
&= -\frac{h\Gamma_e}{12}
\end{aligned}
$$

So we can write a single expression for the any element in the contribution term for our local stiffness matrix as:

$$
K_{p,q}^e = -\frac{h(1+\delta_{pq})}{12}
$$

which is simple enough that we could write out the whole thing as:

$$
K^e = -\frac{h\Gamma_e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{27}
$$

Considering the contribution of the Neumann boundary condition term to the load vector, we have to perform the integral:

$$
\mathbf{s}_p^f = k \int_{\Gamma_e} \eta_p \nabla T \cdot d\mathbf{\Gamma}
$$

An important point to note is that we will only perform this integral over the boundary of the element (which in our case translates into a face of a triangle) if that element happens to be on a Neumann boundary of our computational domain. In this case p will be in the range 1 to 3. Because our Neumann boundary conditions are applied

normal to the boundary, the gradient can be specified as a scalar (the implied direction being normal to the boundary face) and so the dot product will drop out of the integral. We will furthermore assume that T does not vary over the boundary of the element. Making use of the second integration formula (24) we then have:

$$\begin{aligned}
\mathbf{s}_p^f &= k \int_{\Gamma_e} \eta_p \nabla T \cdot d\mathbf{\Gamma} \\
&= k\nabla T \int_{\Gamma_e} \eta_p \cdot d\mathbf{\Gamma} \\
&= k\nabla T \frac{h1!0!0!2\Gamma_e}{(1+0+0+2)!} \\
&= \frac{k\nabla T \Gamma_e}{3}
\end{aligned}$$

So we can write the contribution to our load vector as:

$$\mathbf{s}_p^f = \frac{k\nabla T \Gamma_e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{28}$$

And finally we obtain the first part of our load vector as follows:

$$\begin{aligned}
\mathbf{s}_p^e &= hT_{air} \int_{\Gamma_e} \eta_p \cdot d\mathbf{\Gamma} \\
&= hT_{air} \int_{\Gamma_e} \eta_p \cdot d\mathbf{\Gamma} \\
&= hT_{air} \frac{h1!0!0!2\Gamma_e}{(1+0+0+2)!} \\
&= \frac{hT_{air}\Gamma_e}{3}
\end{aligned}$$

So we can write the contribution to our load vector as:

$$\mathbf{s}_p^e = \frac{hT_{air}\Gamma_e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{29}$$

4 Implementation

Write your Implementation here.

4.1 Matlab program

Write your Implementation here.

4.2 Hybrid openMP and MPI program

Write your Implementation here.

5 Discussion

Write your Discussion here.

6 Results

Write your Results here.

7 References

Write your References here.