Derivation:

$$\rho c \frac{dT}{dt} = k\nabla^2 \mathrm{T}$$

Apply weak form derivation:

$$\int_{\Omega} W(\,\rho c \frac{dT}{dt} - k\nabla^2 \mathrm{T}) d\Omega = 0$$

$$\int_{\Omega} W\,\rho c \frac{dT}{dt} d\Omega - \int_{\Omega} Wk\nabla^2 \mathrm{T} d\Omega = 0$$

Expand the equation:

$$\int_{\Omega} W\rho c \frac{dT}{dt} d\Omega - (\int_{\Omega} k\nabla \cdot (W\nabla T) d\Omega - \int_{\Omega} k\nabla W \cdot \nabla T\, d\Omega) = 0$$

Then using Einstein's techniques:

$$\int_{\Omega} W\rho c \frac{dT}{dt} d\Omega - (\int_{\Gamma} kW\nabla T \cdot d\Gamma - \int_{\Omega} k\nabla W \cdot \nabla T\, d\Omega) = 0$$

Re-arrange this equation to be:

$$\int_{\Omega} W\rho c \frac{dT}{dt} d\Omega + \int_{\Omega} k\nabla W \cdot \nabla T\, d\Omega = \int_{\Gamma} kW\nabla T \cdot d\Gamma$$

$$\int_{\Gamma} kW\nabla T \cdot d\Gamma = \int_{\Gamma} WQ_{cpu} \cdot d\Gamma - \int_{\Gamma} WQ_{air} \cdot d\Gamma$$

$$\int_{\Gamma} kW\nabla T \cdot d\Gamma = \int_{\Gamma} WQ_{cpu} \cdot d\Gamma - \int_{\Gamma} hW(T - T_{air}) \cdot d\Gamma$$

Re-arrange the equation with weight function in weak form:

$$\int_{\Omega} W\rho c \frac{dT}{dt} d\Omega + \int_{\Omega} k\nabla W \cdot \nabla T\, d\Omega = \int_{\Gamma} WQ_{cpu} \cdot d\Gamma - \int_{\Gamma} hW(T - T_{air}) \cdot d\Gamma$$

then

$$\int_{\Omega_e} \eta_p \eta_q \rho c \frac{dT_q}{dt} d\Omega + \int_{\Omega_e} k\nabla \eta_p \cdot \nabla \eta_q T_q\, d\Omega = \int_{\Gamma_e} \eta_p Q_{cpu} d\Gamma - \int_{\Gamma_e} h\eta_p(\eta_q T - T_{air}) d\Gamma$$

so while p and q are in the range 1 to 4 for our tetrahedron element, the overall system modeling equation will be:

$$\sum_{e=1}^{Ne} \int_{\Omega_e} \eta_p \eta_q \rho c \frac{dT_q}{dt} d\Omega + \sum_{e=1}^{Ne} \int_{\Omega_e} k \nabla \eta_p \cdot \nabla \eta_q T_q \, d\Omega$$
$$= \sum_{e=1}^{Ne} \int_{\Gamma_e} \eta_p Q_{cpu} d\Gamma - \sum_{e=1}^{Ne} \int_{\Gamma_e} h \eta_p (\eta_q T - T_{air}) d\Gamma$$

From the giving information, Ne will be the total number of elements and Qcpu and Tair are both constant.

re-arrange the equation in the form of:

$$M \frac{dT_q}{dt} = K T_q + s$$

which will give us:

$$M = \sum_{e=1}^{Ne} \int_{\Omega_e} \eta_p \eta_q \rho c d\Omega$$
$$K = -\sum_{e=1}^{Ne} \int_{\Omega_e} k \nabla \eta_p \cdot \nabla \eta_q \, d\Omega - \sum_{e=1}^{Ne} \int_{\Gamma_e} h \eta_p \eta_q d\Gamma$$
$$s = \sum_{e=1}^{Ne} \int_{\Gamma_e} \eta_p Q_{cpu} d\Gamma + \sum_{e=1}^{Ne} \int_{\Gamma_e} h \eta_p T_{air} d\Gamma$$

These derivation above are the discretization part of our partial differential equations, the next step is determine the key values that will be put into the formula. For calculating the integration of the shape function of M and K matrices, it requires to use the integration formulae for the integration of the shape function.

The integration formulae is defined as:

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

And

$$\int_{\Gamma_e} \eta_p^a \eta_q^b \eta_r^c \eta_s^d d\Gamma = \frac{a!\, b!\, c!\, 2\Gamma_e}{(a+b+c+2)!}$$

Where Ω_e is the volume of the tetrahedron volume and Γ_e is the tetrahedron face area of the element.

M is a 4 by 4 matrix because p and q are in the range from 1 to 4, and for the case when p and q are equal:

$$\int_{\Omega_e} \eta_p^2\, \eta_q^0 = \frac{2!\,6\Omega_e}{(2+3)!} = \frac{1}{10}\Omega_e$$

and for p and q are not equal:

$$\int_{\Omega_e} \eta_p^1\, \eta_q^1 = \frac{1!\,1!\,6\Omega_e}{(2+3)!} = \frac{1}{20}\Omega_e$$

so the M matrix of each element will be:

$$M = \frac{\rho c \Omega_e}{20}\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

and Ω_e is the volume of the tetrahedron element.

When calculating the one robin condition part of the K matrix, we will use the integration formulae for the surface area, and as we know the each surface area is a triangular shape so there will be three points to form a face, in this case the p and q values are in the range from 1 to 3, when p and q values are equal:

$$\int_{\Gamma_e} \eta_p^2 \eta_q^0 d\Gamma = \frac{2!\,2\Gamma_e}{(2+2)!} = \frac{1}{6}\Gamma_e$$

and when p and q are not equal:

$$\int_{\Gamma_e} \eta_p^1 \eta_q^1 d\Gamma = \frac{1!\,1!\,2\Gamma_e}{(1+1+2)!} = \frac{1}{12}\Gamma_e$$

and the matrix of the shape function will be the 3 by 3 matrix, so this part of the matrix will be:

$$\frac{\Gamma_e}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

for the $-\sum_{e=1}^{Ne} \int_{\Omega_e} k \nabla\eta_p \cdot \nabla\eta_q \, d\Omega$part of the K matrix, from the lecture notes in page 349, to calculate $\nabla\eta_p \cdot \nabla\eta_q$, it will follow the following derivations:

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

to simplify the calculation in coding, we will also make a G matrix that will perform all of the elements calculation of each point of the element, the G matrix for the tetrahedron element is given by:

```
  G=
[((y(4)-y(2))*(z(3)-z(2))-(y(3)-y(2))*(z(4)-z(2))),
((y(3)-y(1))*(z(4)-z(3))-(y(3)-y(4))*(z(1)-z(3))),
((y(2)-y(4))*(z(1)-z(4))-(y(1)-y(4))*(z(2)-z(4))),
((y(1)-y(3))*(z(2)-z(1))-(y(1)-y(2))*(z(3)-z(1)));

((x(3)-x(2))*(z(4)-z(2))-(x(4)-x(2))*(z(3)-z(2))),
((x(4)-x(3))*(z(3)-z(1))-(x(1)-x(3))*(z(3)-z(4))),
((x(1)-x(4))*(z(2)-z(4))-(x(2)-x(4))*(z(1)-z(4))),
((x(2)-x(1))*(z(1)-z(3))-(x(3)-x(1))*(z(1)-z(2)));

 ((x(4)-x(2))*(y(3)-y(2))-(x(3)-x(2))*(y(4)-y(2))),
((x(3)-x(1))*(y(4)-y(3))-(x(3)-x(4))*(y(1)-y(3))),
```

```
((x(2)-x(4))*(y(1)-y(4))-(x(1)-x(4))*(y(2)-y(4))),
((x(1)-x(3))*(y(2)-y(1))-(x(1)-x(2))*(y(3)-y(1)))];
```

and in the loop p and q will be represent two points in the element, the calculation for each element of this part $\sum_{e=1}^{Ne} \int_{\Omega_e} \nabla\eta_p \cdot \nabla\eta_q \, d\Omega$ will be:

looping all 4 points in the element for p from 1 to 4 and q from 1 to 4

```
Gp              = [G(1,p),  G(2,p),G(3,p)];
Gq              = [G(1,q),  G(2,q),G(3,q)];
```

$$\int_{\Omega_e} \nabla\eta_p \cdot \nabla\eta_q d\Omega = (\mathrm{Gp} \cdot \mathrm{Gq})/(36\Omega)$$

for calculating the weight function of s where only including the shape integration formulae:

$$\int_{\Gamma_e} \eta_p d\Gamma = \frac{1!\,2\Gamma_e}{(1+2)!} = \frac{1}{3}\Gamma_e$$

The final part of the derivation before coding is to calculate the volume of tetrahedron element and face areas of each face in the element. Which is :

```
% Area of a triangular element with coordinates
% (x1, y1, z1), (x2, y2, z2), (x3, y3, z3):
Gamma = sqrt(((y2-y1)*(z3-z1) - (z2-z1)*(y3-y1))^2 ...
          + ((z2-z1)*(x3-x1) - (x2-x1)*(z3-z1))^2 ...
          + ((x2-x1)*(y3-y1) - (y2-y1)*(x3-x1))^2)/2;
```

and

```
% Volume of a tetrahedral element with coordinates
% (x1, y1, z1), (x2, y2, z2), (x3, y3, z3), (x4, y4, z4):
Omega = abs( x1*y2*z3 - x1*y3*z2 - x2*y1*z3 ...
          + x2*y3*z1 + x3*y1*z2 - x3*y2*z1 ...
          - x1*y2*z4 + x1*y4*z2 + x2*y1*z4 ...
          - x2*y4*z1 - x4*y1*z2 + x4*y2*z1 ...
          + x1*y3*z4 - x1*y4*z3 - x3*y1*z4 ...
          + x3*y4*z1 + x4*y1*z3 - x4*y3*z1 ...
          - x2*y3*z4 + x2*y4*z3 + x3*y2*z4 ...
          - x3*y4*z2 - x4*y2*z3 + x4*y3*z2 ) /6;
```

MATLAB CODE ANALYSIS: