Derivation:

Apply weak form derivation:

Expand the equation:

then using Einstein summation notation:

Re-arrange this equation to be:

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

then

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

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:

which will give us:

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 are defined as:

And

where is the volume of the tetrahedron volume and 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:

and for p and q are not equal:

so the M matrix of each element will be:

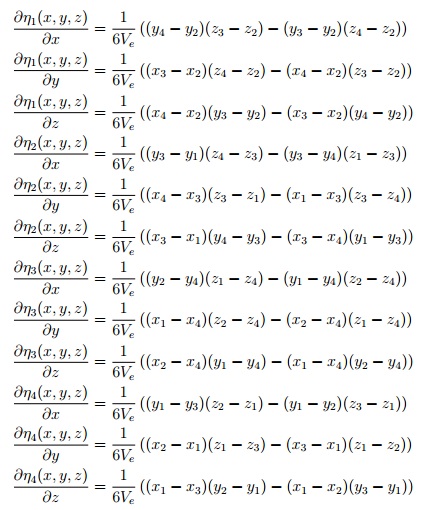
and is the volume of the tetrahedron element.

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:

and when p and q are not equal:

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

for the part of the K matrix, from the lecture notes in page 349, to calculate , it will follow the following derivations:



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 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)];

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

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. This 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 ANALYSIS:

After discretize the original equation by Finite Element Method, the ODE of the original equation will be solved by implicit Euler’s method. In the MATLAB analysis, the discretized format of the original equation will be in the form of:

M, K and s will be constructed in the CPU\_HEAT function by looping over all of the tetrahedron elements with analysis on the boundary conditions.

The first part of the MATLAB analysis is to write a function that can return M, K and s values. Initially, the volume of each tetrahedron element and surface area of each surface are calculated and stored in arrays. Then M and part of the K will be calculated by looping around all elements and based on the geometrical information of that particular element.

For the K matrix construction, refer to the derivation section, part is one part of the Robin condition of the CUP fin. With consideration on both the Neumann and Robin boundary faces, the s values and will be calculated by looping all of the faces of the sides with boundary conditions.

Implicit Euler’s method is applied in the time marching loop that analyzed the equation from 0s to 100s. By applying Implicit Euler’s method, the equation will be transformed into the form

which can be written in the form:

And

In the time marching loop T could be caudated from T values from previous time step by calculating T=b/A.

