Proposing a Numerical Solution for the 3D Heat Conduction Equation

Mansour al Qubeissi
Researcher in Thermo-Fluid Mechanics Research Centre (TFMRC)
Eng. Dept., University of Sussex
Brighton, UK BN1 9QT
ma292@sussex.ac.uk

Abstract— **The current paper presents a numerical technique in solving the 3D heat conduction equation. The Finite Volume method is used in the discretisation scheme. Gauss's theorem has also been employed for solving the integral parts of the general heat conduction equation in solving problems of steady and unsteady states. The proposed technique is applicable to unstructured (tetrahedral) elements for dealing with domains of complex geometries. The validation cases of the developed, FORTRAN based, heat conduction code in 1D, 2D and 3D representations have been reviewed with a grid independence check. Comparisons to the available exact solution and a commercial software solver are attached to the manuscript.**

Keywords-Heat Transfer; Heat Conduction code, Finite Volume Method; Gauss's theorem; Computational Fluid Dynamics.

I. Introduction

The heat conduction problems are very common industrial cases. HC code is developed to solve the heat conduction equation. The governing equation is solved for a control volume, shown in "Fig. 1", with no internal heat generation as:

$$\rho C\,\frac{\partial T}{\partial t} + \vec{\nabla}\cdot\vec{q} = 0 \qquad (1)$$

In (1), T is the temperature value given in (K) and $\rho\, C$ represent density and heat capacity with units $(\mathrm{kg\,m^{-3}})$ & $(\mathrm{W\,kg^{-1}K^{-1}})$ respectively. $\vec{q}$ is the heat flux in three components, as a vector value, which is given by Fourier's law as:

$$\vec{q} = -k\,\vec{\nabla}T \qquad (1.1)$$

The numerical solution of (1) is further described in the following section.

II. Finite Volume Method

Equation (1) can be discretised, over a control volume, as shown in "Fig. 1", by the application of the variational method [1] and [2]:

$$\rho C\int_V \frac{\partial T}{\partial t}\,dV = \int_V \vec{\nabla}\cdot\left(k\,\vec{\nabla}T\right)dV;\ k = Const.$$

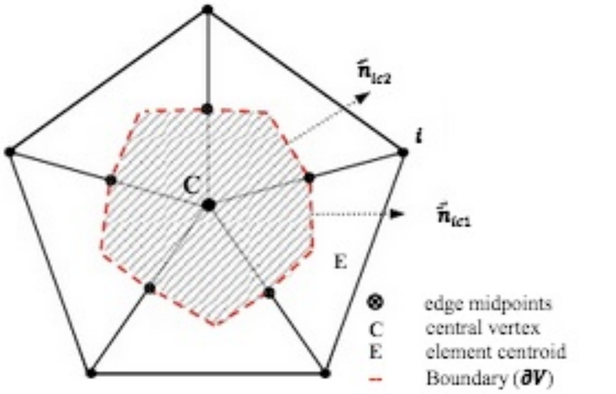

Figure 1. The 2D view of a vertex centered control volume (the shaded area)

$$\int_V \frac{\partial T}{\partial t}\,dV = \alpha\int_V \bar{\nabla}\cdot\bar{\nabla}T\,dV \qquad (2)$$

Where the diffusivity $\alpha = \frac{k}{\rho C}$ (with units m^2s^{-1}). The LHS of (2) takes the form of:

$$\int_V \frac{\partial T}{\partial t}\,dV = \frac{\partial T}{\partial t}\,V \qquad (2.1)$$

This term can be solved using the finite difference (forward) explicit scheme:

$$\frac{\partial T}{\partial t} \cong \frac{T^{t+\Delta t} - T^{t}}{\Delta t}$$

It is worth mentioning that in the case of a steady state, $\frac{\partial T}{\partial t}$ refers to the residuals (change of temperature) over the time steps Δt [2]. This means in a steady state at $t\sim\infty, \frac{\partial T}{\partial t} \to 0$ [3]. The RHS of (2) can be solved using Gauss's (divergence) theorem [3] and [4 as:

$$\alpha\int_V \bar{\nabla}\cdot\bar{\nabla}T\,dV = \alpha\oint_{\partial V}\bar{\nabla}T\cdot\bar{n}\,\mathrm{d}S \qquad (2.2)$$

Where $\bar{n}$ is the outward normal vector. By substituting the terms (2.1) and (2.2) into (2), the heat conduction equation takes the form of:

978-0-7695-4730-5/12 $26.00 © 2012 IEEE
DOI 10.1109/AMS.2012.10

$$\frac{T^{t+\Delta t}-T^{t}}{\Delta t} = \frac{\alpha}{V}\oint_{\partial V}\vec{\nabla}T\cdot d\vec{S} \tag{3}$$

Where the weight at the boundary of the control volume for each edge is $d\vec{S} = \vec{n}\,\mathrm{dS}$. $\vec{\nabla}T$ is the average temperature gradient at the boundary of the control volume ∂V located at the midpoint, as shown in figure (1), i.e.

$$\vec{\nabla}T = \frac{\vec{\nabla}T_i + \vec{\nabla}T_c}{2} \tag{3.1}$$

Equation (3.1) gives the temperature gradient at the boundary of the control volume, by considering linear distribution of gradient along the edge. Similarly, $\bar{T} = \frac{T_i + T_c}{2}$ [3] and [5]. The gradient temperature components are also calculated using Gauss's (divergence) theorem at the control volume boundary points (midpoints) as:

$$\vec{\nabla}T_c = \frac{1}{V_c}\oint_{\partial V}\bar{T}\,d\vec{S} \tag{4}$$

Or in a matrix format, (4) can be re-written as:

$$\vec{\nabla}T_c = \frac{1}{V_c}\sum_{i=1}^{N}\bar{T}\,\Delta\vec{S}_i$$

Equation (3) can be re-written in a matrix format as:

$$\frac{T^{t+\Delta t}-T^{t}}{\Delta t} = \frac{\alpha}{V}\sum_{i=1}^{N}\left(\vec{\nabla}T\cdot\Delta\vec{S}\right)_i$$

Or further simplified to:

$$T^{t+\Delta t} = T^{t} + \beta\sum_{i=1}^{N}\left(\vec{\nabla}T\cdot\Delta\vec{S}\right)_i \tag{5}$$

Where $\beta = \frac{\Delta t\,\alpha}{V}$, i is index of neighbor nodes in that cell (around node c) and N is number of neighbor nodes around that cell central node. Low values of β within this range will keep a bound to the amplified errors between initial and updated values. The value of β is the control parameter of the equation. Von Neumann condition of stability leads to a restriction over the values of stability factors [3] as:

$$0 < \alpha\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}\right)\Delta t \le \frac{1}{2}$$

Or simplified to:

$$0 < \alpha\left(\frac{A}{V}\right)^2\Delta t \le \frac{1}{2}$$

Where, A represents the boundary area of the control volume V , which is a scalar value given by $A = \sqrt{\Delta x^2\Delta y^2 + \Delta x^2\Delta z^2 + \Delta y^2\Delta z^2}$ and V is the control volume given by $V = \Delta x\Delta y\Delta z$, with the units m^2 and m^3 respectively. The range of time steps can be expressed by:

$$0 < \Delta t \le \frac{1}{2\alpha}\left(\frac{V}{A}\right)^2 \tag{5.1}$$

Tests over the value Δt were carried out within the range given in (5.1). The test was applied on a simple geometry, given in appendix B, with the following grid densities: $\sim 390, \sim 13000, \sim 91000$ and $\sim 500{,}000$ nodes. With finer grids Δt tends to be smaller to a minimum of $\sim 0.5\ s$. Δt becomes bigger with coarser meshes to $\sim 6\ \mathrm{s}$. Higher values of Δt in the tests resulted in unstable solutions. Hence, it is essensial to assume a fixed small value of Δt in order to avoid developing unstable solutions with coarse grids by assuming $\Delta t = 0.5\ s$ regardless to the grid sizes. Also, in (5) the temperature is updated from the given initial guess and boundary conditions. Hence, it is necessary to choose the appropriate initial guess. This can be set within the given range of boundary conditions. Equation (5) is applied at each cell central node (indexed c in figure 1) for solving 2D and 3D problems.

III. Boundary conditions

There are two major types of boundary conditions applied, either given boundary temperature (Dirichlet) or given heat flux (Neumann) boundary conditions. There can be a case of a mix of these two types of boundary conditions. Further details about these types are given in the following sections.

A. Dirichlet boundary condition

When the temperature is imposed at the boundary, a Dirichlet boundary condition is applied. Hence the temperature values at the boundary nodes will be replaced by the prescribed values of temperature $\bar{T}$ over that boundary, i.e.

$$T = \bar{T}$$

However, the temperature gradient is still needed in (3.1) and thereafter (5) in order to find the temperatures at the adjacent nodes.

B. Neumann boundary conditions

In this type of boundary conditions, the normal temperature gradient (as a function of heat flux) is imposed at the boundary. The heat conduction flux is given by Fourier's law as: $\vec{q}_c = -k\,\vec{\nabla}T$. At the boundary, the normal heat flux from both sides is balanced as:

$$\vec{q}_c\cdot\vec{n} = q_b$$

Where, $\vec{n}$ is the normal vector components at that boundary node, $\vec{q}_c$ is the heat flux components that conducted from internal nodes towards the centre of the boundary cell and q_b is the prescribed heat flux from outside the domain normal to the boundary, which can be convection ($q_b = h\,\Delta T_{fb}$), fixed heat flux ($q_b = const.$), or adiabatic ($q_b = 0$). Therefore,

$$k\,\vec{\nabla} T \cdot \vec{n} + q_{\ b} = 0 \tag{6}$$

The boundary temperature gradient has two components normal $\vec{F}_n$ and tangential $\vec{F}_t$. Equation (6) is subject to the normal boundary fluxes only. The tangential portion of the flux is automatically set to zero [5]. It is therefore important to calculate the tangential temperature gradient. The tangential gradient is unknown but it can be decomposed from the tangent gradient of neighbor nodes [2]. At the boundary, the temperature gradient can be decomposed as:

$$\vec{\nabla} T = \vec{F}_n + \vec{F}_t \tag{7}$$

To overcome the complexity of determining the tangential vectors, the tangential temperature gradient is decomposed from the estimated gradient as,

$$\vec{F}_t = \vec{\nabla} T_e + (\vec{\nabla} T_e \cdot \vec{n})\,\vec{n} \tag{7.1}$$

Where: $\vec{\nabla} T_e$ represents the estimated value of $\vec{\nabla} T$, which is defined as the rate of heat exchange (divided by k) per unit volume of the surface [2]. It is important to distinguish between $\vec{n}$, given in (6) and (7.1), which is the average inward normal vector at the boundary.

$$\vec{\nabla} T_e \cong \vec{\nabla} T$$

The estimated gradient $\vec{\nabla} T_e$ is calculated from the assumed control volume at the boundary by creating a mirror cell to close that control volume, as shown in "Fig. 2". When the weights and the volume are doubled for that control volume (4) will be called again, with adding the boundary temperature distribution effect, for computing $\vec{\nabla} T_e$ as:

$$\vec{\nabla} T_e = \frac{1}{V_b}\left(\int_{\partial V} \bar{T}\, d\vec{S} + \int_{A} \bar{T}\,\vec{n}\,\Delta a \right) \tag{7.2}$$

Where, Δa is the boundary area of that node within the connected boundary elements and $\mathbf{A}$ refers to the total surface area of that boundary node. The gradient in (7.2) takes into account tangential derivatives accurately, but not the normal ones. However, with Neumann boundary type it is only important to get the right tangential component because the normal one is already specified by the imposed normal heat flux, in (6), as:

$$\vec{F}_n = -\frac{q_b}{k}\,\vec{n} \tag{7.3}$$

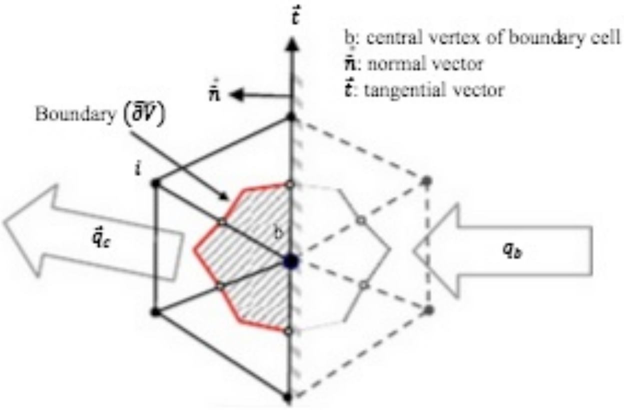

Figure 2. A 2D view of a boundary cell showing the normal and tangential vectors

By substituting (7.1) and (7.3) into (7), the (corrected) temperature gradient at the surface is:

$$\vec{\nabla} T = -\frac{q_b}{k}\,\vec{n} + \vec{\nabla} T_e + (\vec{\nabla} T_e \cdot \vec{n})\,\vec{n} \tag{8}$$

In order to find the temperature at the boundary, the heat flux term is added to (5) for closing the boundary control volume. Equation (5) at the boundary will become:

$$T^{t+\Delta t} = T^t + \beta \sum_{\imath=1}^{N} (\vec{\nabla} T \cdot \Delta \vec{S})_\imath + \sum_{j=1}^{K} (q_b\,\Delta \mathrm{a})_j \tag{9}$$

Where j is index of boundary elements and K is the number of boundary elements of that control volume. $\mathbf{\Delta a}$ is the boundary area of each boundary element. The temperature gradient $\vec{\nabla} T$, given in (8), can be substituted into (9) in order to find the temperature at the boundary. Although "Fig. 1"and "Fig. 2" of the control volume show 2D views for simplifications, the discretised formats, (5) and (9), of the heat conduction equation are straightforwardly applicable to 3D problems.

IV. VALIDATION CASES

The solution, derived in previous sections, has been validated against the available exact solution for 1D and 2D cases. The mesh generated for this case is unstructured tetrahedral element type. In 3D case, the validation was carried out against ANSYS (commercial Computation Fluid Dynamics solver). A grid independence check was carried out for a wide range of node numbers of: **~390, 13000 & 91000** nodes, as shown in "Fig. 3".

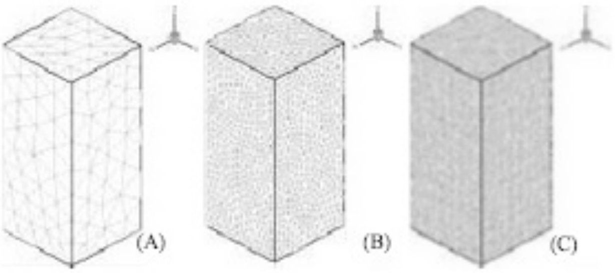

Figure 3. 3D view of the tetrahedral meshes used in testing HC code for the 3D case, (A) **~390** nodes (B) **~13000** nodes (C) **~91,000** nodes

A. 1D case

In 1D case, the temperature distribution (heat conduction) equation can be derived to:

$$T = C_1\, z + C_2$$

Where temperature varies only in z-direction. C_1 **&** C_2 are constants. A fixed temperature (Dirichlet) boundary condition is specified at both ends of the domain as,
At $z = 0, T = 400\ K$ and at $z = 1, T = 300\ K$. The other sides of the domain were assumed perfectly adiabatic. The analytical (exact) solution of this case becomes:

$$T = -100\ (z) + 400$$

Where $0 \leq z \leq 1$. The given solution of this case compared to the analytical solution and ANSYS is shown in appendix (A).

B. 2D case

In 2D case, geometry of a plate, with dimensions$1 \times 1 \times 0.2\ m^3$, was used for validation. The following types of boundary conditions have been prescribed: at x=0 and x=1, T = 0; at y=0, T = 0; at y=0.5, $T = \sin(\pi x/L)$ and at z=0 and z=1, adiabatic surfaces $(q = 0)$. This case has been introduced for the availability of the exact solution [6] and [7], as:

$$T(x,y) = \frac{\sinh(\pi\, y/L)}{\sinh(\pi\, H/L)}\ \sin(\pi\, x/L)$$

Where, L is the length of that cross section $(L = 1\ m)$ and H is the hight $(H = 1\ m)$. An excellent agreement with the analytical solution is approved by HC code, as illustrated in diagrams of appendix (B).

C. 3D case

The validation is made, in 3D, for the geometry of a rectangular stainless steel rod, of the dimensions $0.5 \times 0.5 \times 1\ m^3$, which is used for 1D case, with thermal conductivity$k = 15.1\ W\ m^{-1}\ K^{-1}$, as shown in figure 3. The boundary conditions applied on this case are: At x= 0.25 m, y= 0.25 m & z= 0, T = 300 K; At x= 0, y= 0 & z= 1 m, fixed heat flux $(q = 800\ W\ m^{-2})$. The commercial engineering software (ANSYS) is used for comparison with HC due to the difficulty of providing analytical solution for such a 3D case. The results of HC code have shown good agreement with ANSYS solver, as illustrated in diagrams of appendix (C).

V. Conclusions

The finite volume vertex centered scheme has been applied in discretising the heat conduction equation to examine the temperature profile of any domain. The integral format of the equation is solved using Gauss's theorem. The FORTRAN based heat conduction (HC) code is been developed and validated in using unstructured hybrid grids with edge-based calculations. The results, obtained for 1D, 2D and 3D cases, have been assessed with the exact and numerical (commercial) solutions. The predicted temperature profiles given by HC code are in excellent agreement with the available standard solutions. While the code is computationally efficient, further work improvement in the code is possible by imposing radiation boundary condition in comparison to the effect of other types of boundary conditions. Also, a conjugate heat transfer type of boundary conditions is under consideration.

References

[1] Younes, A., Ackerer, P. and Chavent, G. (2004) From Mixed finite elements to finite volumes for elliptic PDEs in two and Three dimensions. Int. J. Numer. Meth. Eng. 59;365-388.

[2] Wegian, F.M., Yazdi, S.R. (2008) Symmetric Boundary Condition Technique in NASIR Galerkin Finite Volume Solver for 3D Temperature Field. J. Jordan Civil Eng. Vol. 2, 4.

[3] Hirsch, C. (1988) Numerical Computation of Internal and External Flows (Vol. 1: Fundamentals of Numerical Discretization). Wiley series in Num. Meth. in Eng; 1.

[4] Chung, T.J. (2002) Computational Fluid Dynamics. Cambridge Univ. Press, UK.

[5] Lyra, P.R.M., de Lima R. de C.F., Guimaraes, de Carvalho, D.K.E. (2004) An edge-based unstructured Finite Volume Procedure for the Numerical Analysis of Heat Conduction Applications. J. Brazillian Soc. Of Mech. Sci. & Eng. 26, No 2.

[6] Karlekar, B.V. and Desmond, R.M. (1982) Heat Transfer. 2nd Ed., Wes Publishing Co.

[7] White, F. M. (1984) Heat Transfer. Addison-Wesley Pub. Co.

APPENDIX (A): 1D CASE

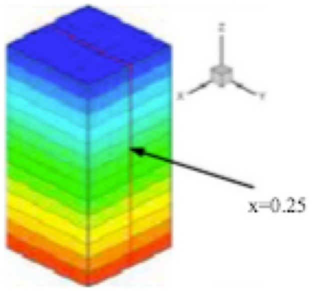

Figure 4. 3D view of the geometry showing the 1D temperature profile

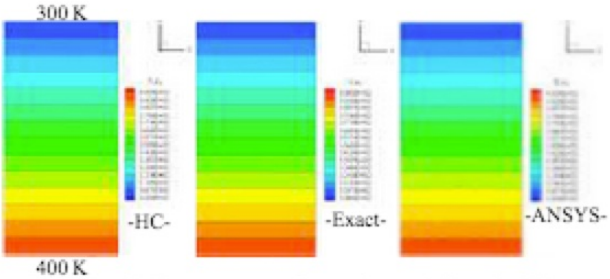

Figure 5. 1D Temperature profile, at plane x=0.25 m, for HC vs. ANSYS & Analytical solutions

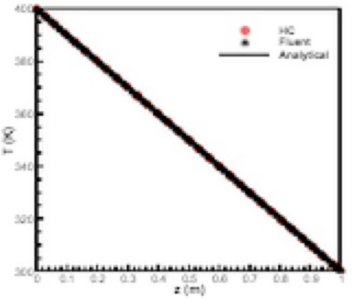

Figure 6. Comparison of HC results vs. ANSYS & Analytical solutions for 1D case

APPENDIX (B): 2D CASE

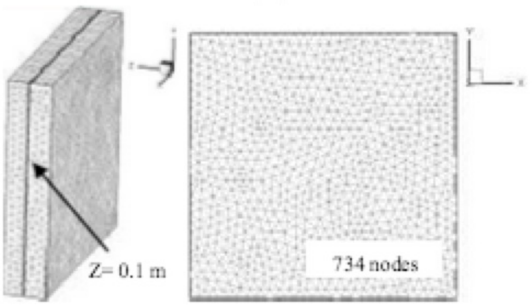

Figure 7. 3D and 2D (at z=0.1) views of the tetrahedral mesh used for solving the 2D case

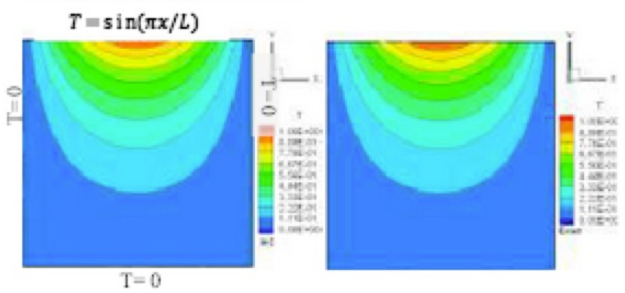

Figure 8. 2D temperature profile at the section z=0.1 for a HC with a mesh density of **~730** nodes (at section z=0.1) in a comparison with the exact solution

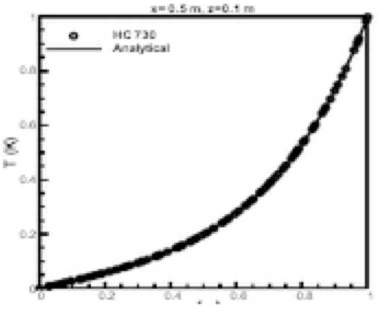

Figure 9. Temperature distribution in y-direction at z=0.1 and x=0.5 of HC (with **~740** nodes) vs. the analytical solution

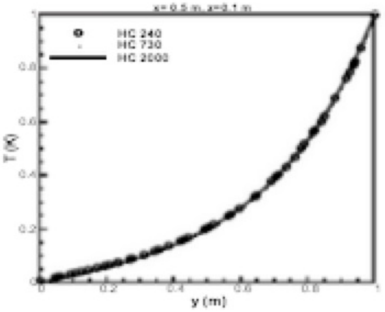

Figure 10. Grid independence check for the 2D case, over a range of node numbers from **~240 to ~2,000** nodes at the section z=0.1, in y-direction

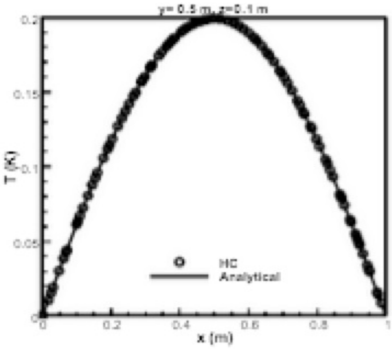

Figure 11. Temperature distribution in x-direction at z=0.1 and y=0.5 of HC (with **~740** nodes) vs. the analytical solution

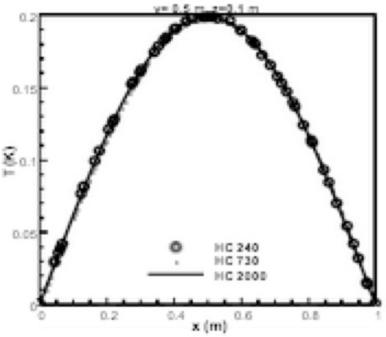

Figure 12. Grid independence check for the 2D case, over a range of node numbers from **~240 to ~2,000** nodes, at section z=0.1, in x-direction

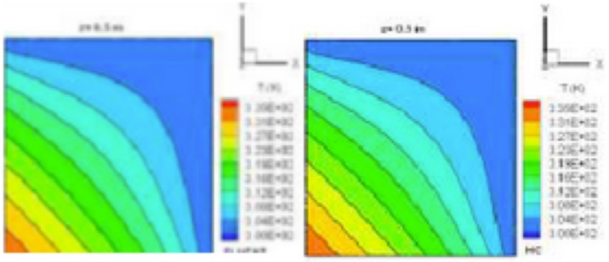

Figure 15. HC and ANSYS color bands of temperature profile at section z=0.5

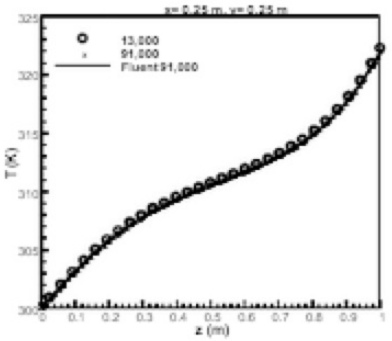

Figure 16. Temperature distribution in z-direction of the 3D case for HC (with two grid densities) vs. ANSYS, at x=0.25 m, y= 0.25 m

APPENDIX (C): 3D CASE

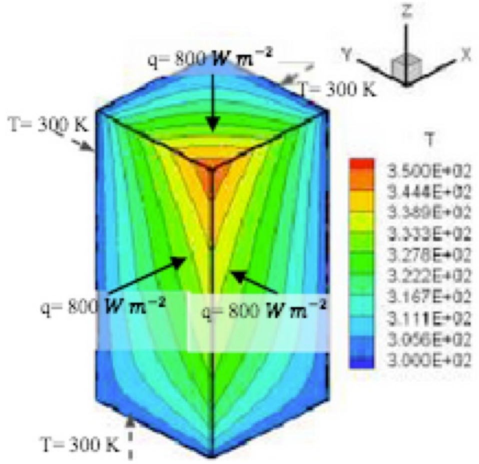

Figure 13. The 3D color bands of HC temperature profile for a mesh of **~13500** nodes

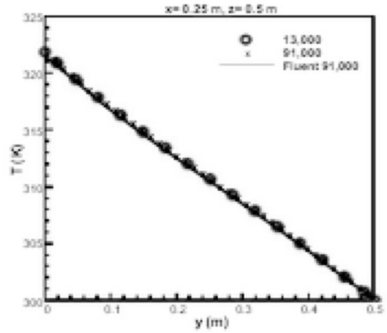

Figure 17. Temperature distribution in y-direction of the 3D case for HC (with two grid densities) vs. ANSYS at x=0.25 m, z= 0.5 m

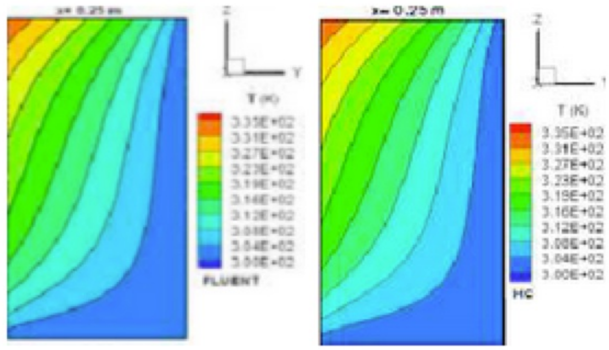

Figure 14. HC and ANSYS color bands of temperature profile at section x=0.25

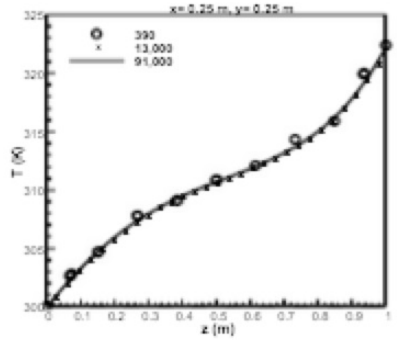

Figure 18. Grid independence check for the 3D case, over a range of node numbers from **~390 to ~91,000** nodes in z-direction