Ninth International Conference on Computational Fluid Dynamics (ICCFD9), Istanbul, Turkey, July 11-15, 2016

ICCFD9-0113

Solution of the Two-Dimensional Steady State Heat Conduction using the Finite Volume Method

Rico Morasata[1], Kürşad Melih Güleren[2]

[1] Faculty of Aeronautics and Astronautics,University of Turkish Aeronautical Association,
Ankara, Turkey
[2] Faculty of Aeronautics and Astronautics, Anadolu University,
Eskişehir, Turkey
Corresponding author: kmguleren@anadolu.edu.tr

Abstract: This project aims to solve a two-dimensional steady-state diffusion problem by means of a home-made Finite Volume Method (FVM) code. This project solves the two-dimensional steady-state heat conduction equation over a plate whose bottom comprises different-sized fins in order to investigate the temperature distribution within a non-uniform rectangular domain. The plate is subject to constant temperatures at its edges. A code was developed to implement the FVM and obtain the temperature distribution over the domain. The code is structured according to the main CFD analysis steps: pre-processing, solution of equations, and post-processing. The set of equations were solved using the iterative Conjugate Gradient Method (CGM). Moreover, the problem was analyzed in ANSYS FLUENT® and the outputs of the two methods are compared.

Keywords: FVM, CFD, steady-state heat conduction, PDE, CGM, ANSYS FLUENT®.

1 Introduction

Heat transfer is an important problems in many disciplines including science, physics, and engineering. Various numerical techniques have been developed in order to solve and simulate real-world heat transfer problems, among which the FVM. In this project, the FVM is implemented to compute the temperature distribution over a plate, finned at its bottom edge, and subject to prescribed boundary temperatures. A home-made code was developed to implement the numerical technique. The system of equations arising from the discretization of the governing equation is solved iteratively by means of the CGM. The obtained temperature distribution is then compared with the output of a commercial code, ANSYS FLUENT®, and a comparative analysis of the two solutions will be presented.

2 Problem Statement

The governing transport equation for a two-dimensional steady-state diffusion problem is given by:

$$\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\Phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\Phi}{\partial y}\right)+S_\Phi=0 \tag{2.1}$$

where x, y are the space dimensions, Γ is the diffusion coefficient, Φ is the diffusive flux, and S_Φ is a source term [2]. For the special case of steady-state heat conduction without volumetric heat generation, and the diffusive flux is the temperature T, (2.1) has the following form:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)=0 \tag{2.2}$$

where k is the material thermal conductivity.
The geometry of the problem is shown in the following figure:

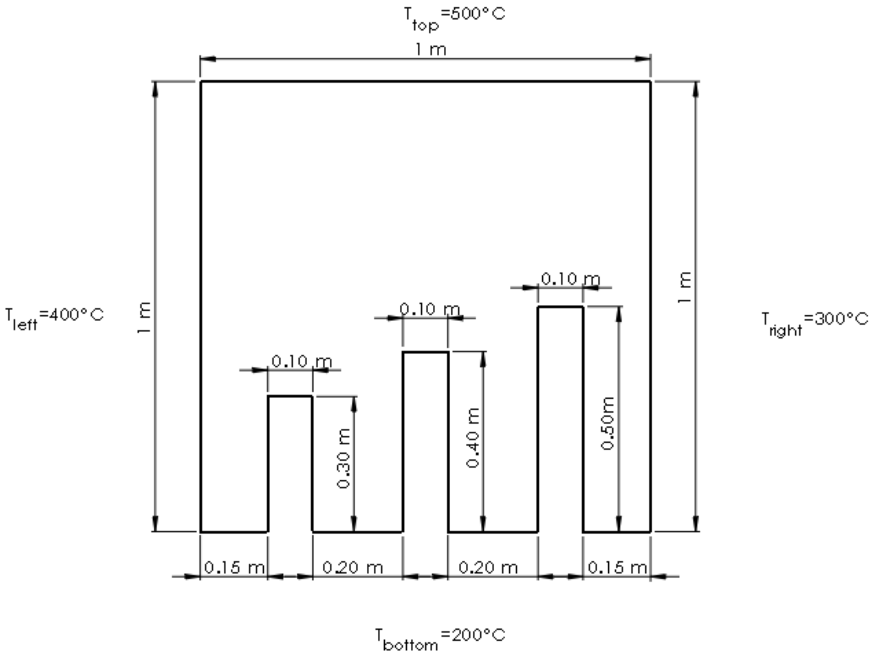

Figure 1: Plate Geometry and Dimensions

The temperatures at the edges are as follows:
$T_{top} = 500°C$, $T_{left} = 400°C$, $T_{right} = 300°C$, $T_{bottom} = 200°C$.

3 Solution Approach

3.1 Governing equation

Since the material has a constant thermal conductivity $k = 1000\,W/(m.K)$, (2.2) can be rewritten as:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{3.1}$$

The above equation is the two-dimensional Laplace's equation to be solved for the temperature field. (3.1) is a linear, homogeneous, elliptic partial differential equation (PDE) governing an equilibrium problem, i.e., steady-state heat conduction, within a closed domain.

3.2 The Finite Volume Method (FVM)

The following assumptions are made to ensure the two-dimensionality of the problem:

- The plate has a uniform thickness.
- There is no heat transfer through the thickness.

The fundamental step in the FVM is the integration of the transport equation (2.1) over a control volume. The domain is discretized into cells or control volumes as shown in Figure 2 to approximate the differential equation. The values of the desired properties are evaluated at the center of the control volume.

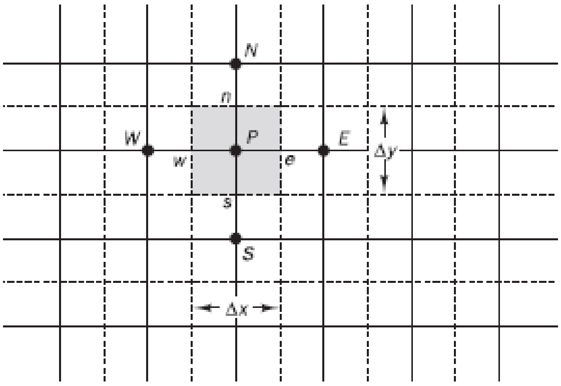

Figure 2: Domain Discretization

The integration of (2.1) over a control volume yields:

$$\int\limits_{\Delta V} \frac{\partial}{\partial x}\left(\Gamma\frac{\partial \Phi}{\partial x}\right) dx.dy + \int\limits_{\Delta V} \frac{\partial}{\partial y}\left(\Gamma\frac{\partial \Phi}{\partial y}\right) dx.dy + \int\limits_{\Delta V} S_\Phi \, dV = 0 \qquad (3.2)$$

In this case, the grid is uniform so all the cell faces have the same area A.
The integrated form of the above equation is:

$$\left[\Gamma_e A_e \left(\frac{\partial \Phi}{\partial x}\right)_e - \Gamma_w A_w \left(\frac{\partial \Phi}{\partial x}\right)_w\right] + \left[\Gamma_n A_n \left(\frac{\partial \Phi}{\partial y}\right)_n - \Gamma_s A_s \left(\frac{\partial \Phi}{\partial y}\right)_s\right] + \overline{S}\Delta V = 0 \qquad (3.3)$$

where the subscripts e, w, n, and s denote East, West, North, and South respectively.
The fluxes through the cell faces are approximated by a first-order finite difference formulation as follows:

$$\Gamma_w A_w \left.\left(\frac{\partial \Phi}{\partial x}\right)\right|_w = \Gamma_w A_w \frac{(\Phi_P - \Phi_W)}{\delta x_{WP}} \qquad (3.4)$$

$$\Gamma_e A_e \left.\left(\frac{\partial \Phi}{\partial x}\right)\right|_e = \Gamma_e A_e \frac{(\Phi_E - \Phi_P)}{\delta x_{PE}} \qquad (3.5)$$

$$\Gamma_s A_s \left.\left(\frac{\partial \Phi}{\partial y}\right)\right|_s = \Gamma_s A_s \frac{(\Phi_P - \Phi_S)}{\delta y_{SP}} \qquad (3.6)$$

$$\Gamma_n A_n \left.\left(\frac{\partial \Phi}{\partial y}\right)\right|_n = \Gamma_n A_n \frac{(\Phi_N - \Phi_P)}{\delta y_{PN}} \qquad (3.7)$$

where P denotes the center of the control volume at which the property Φ is to be evaluated, and δx_{WP} denotes the distance between the west face and the cell center.
Substituting the obtained relations into (3.3) yields:

$$\Gamma_e A_e \frac{(\Phi_E - \Phi_P)}{\delta x_{PE}} - \Gamma_w A_w \frac{(\Phi_P - \Phi_W)}{\delta x_{WP}} + \Gamma_n A_n \frac{(\Phi_N - \Phi_P)}{\delta y_{PN}} - \Gamma_s A_s \frac{(\Phi_P - \Phi_S)}{\delta y_{SP}} + \overline{S}\Delta V = 0 \qquad (3.8)$$

The domain is discretized into square cells of side length dx and dy with $dy = dx$. Since, for the case at hand, the source term is absent, i.e, no heat generation by the plate, and that the diffusion coefficient is k, (3.8) becomes:

$$kA\frac{(T_E - T_P)}{dx} - kA\frac{(T_P - T_W)}{dx} + kA\frac{(T_N - T_P)}{dy} - kA\frac{(T_P - T_S)}{dy} = 0 \qquad (3.9)$$

The above equation represents the discretization of the governing transport equation at the cell centers, within the domain boundaries. The discretization of the temperature gradients in (3.3) requires a special treatment since for the cell adjacent to an edge boundary, its center is located at $dx/2$ or $dy/2$ from the boundary in a uniform grid.
The discretized domain is shown in the following figure:

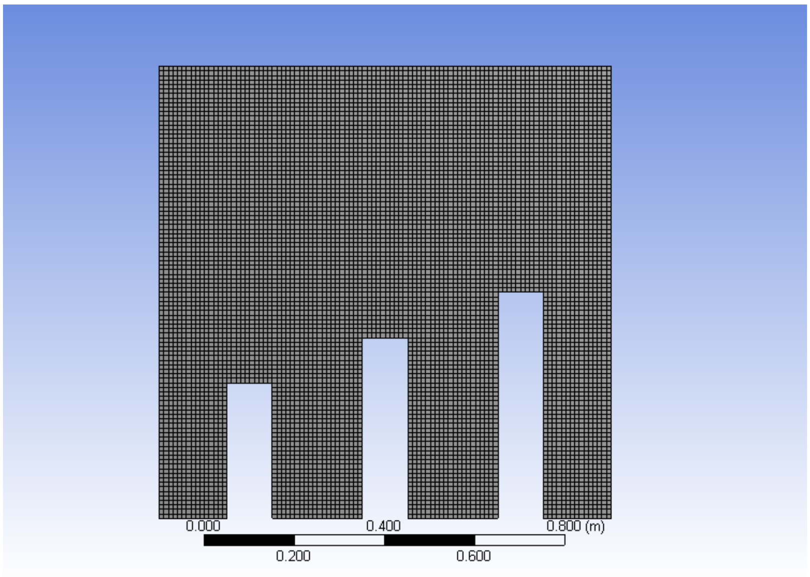

Figure 3: Meshed Plate

The plate was meshed using ANSYS Meshing module.
The discretization of the governing equation at an internal node is performed by considering all the surrounding cell centers, as shown in the following figure:

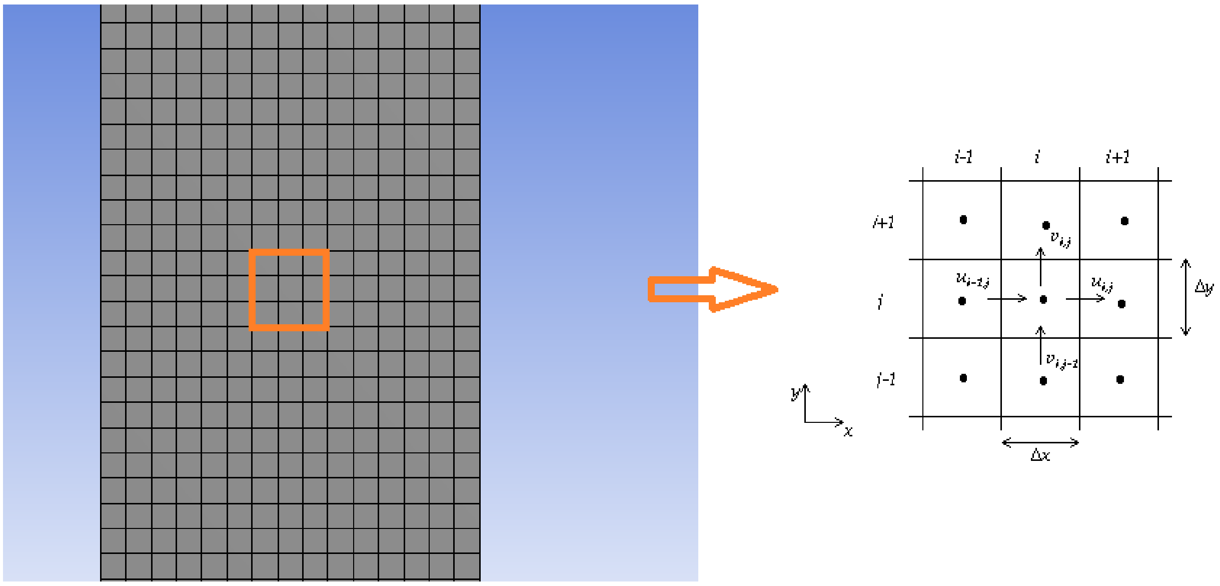

Figure 4: Discretization Process

The plate is uniformly discretized in a two-dimensional Cartesian coordinate system. Indexing the node of interest by (i, j),the discretization of the governing transport equation for the cell at the bottom left corner is as follows:

$$\frac{[T(2,1) - T(1,1)]}{\Delta x} + \frac{[T_{left} - T(1,1)]}{\Delta x/2} + \frac{[T(1,2) - T(1,1)]}{\Delta y} + \frac{[T_{bottom} - T(1,1)]}{\Delta y/2} = 0 \tag{3.10}$$

3.3 System of Equations

The resulting set of algebraic equations can be put into a matrix equation of the form:

$$\mathbf{Ax} = \mathbf{b} \tag{3.11}$$

where $\mathbf{A}$ is the coefficient matrix, $\mathbf{x}$ is the vector of nodal temperatures, and $\mathbf{b}$ is the right-hand-side vector.
The coefficient matrix $\mathbf{A}$ has the following properties:

- $\mathbf{A}$ is square of size $N \times N$, N is the number of equations.
- $\mathbf{A}$ is a banded tridiagonal matrix.
- $\mathbf{A}$ is sparse.

4 Numerical Implementation

The solution of the matrix equation requires appropriate numerical techniques since $\mathbf{A}$ is large and sparse, so as to reduce the computational cost. Matrix equations are chiefly solved using direct methods or iterative methods. Direct methods are computationally costly since the number of operations to reach a solution is of the order of N^3, which requires a high amount of storage. In contrast, iterative methods are more advantageous since they require fewer number of operations.
Various numerical techniques can be used to solve the matrix equation. Two of these were considered in this project: the Gauss-Seidel method (GSM) and the Conjugate Gradient method.

- The Gauss-Seidel method
  This method starts with an initial guess of the solution and uses updated values as soon as these are available. It is commonly used for the solution of linear systems of equations.
- The Conjugate Gradient Method
  The CGM belongs to a family of numerical methods referred to as *Krylov subspace methods*,i.e.,at every iteration step k, the methods searches for a good approximation to the solution of (3.11) from the subspace $span\{\mathbf{b}, \mathbf{Ab}, \mathbf{A^2b}, ..., \mathbf{A^{k-1}b}\}$ [3].

4.1 Code Structure

The structure of the script is as follows:

- The user is asked to enter n, the number of divisions along the top edge. The coefficient matrix is of size $n^2 \times n^2$. For this problem, it is desired to obtain the temperature distribution for $n = 100$.
- The edge lengths, problem constants, along with boundary temperatures are defined as constants.
- $\mathbf{A}$ and $\mathbf{B}$ are preallocated, $\mathbf{B}$ is of size $n^2 \times 1$.
- The matrices are filled with the coefficients of the discretized equations.
- The plate was considered as a full square to simplify the problem. The empty cells are assigned constant temperatures so that the desired solution will not be affected.
- The initial guess vector is defined along with the tolerance, and the maximum number of iterations.
- The matrix $\mathbf{A}$ is checked for symmetry and positive definiteness. If both conditions are satisfied, the CGM is implemented as the iterative scheme, and the GS otherwise.
- The solver iterates until the desired tolerance is reached by the residual.
- The obtained temperature vector is reshaped into a square matrix required for plotting the temperature distribution.
- The coordinates of the nodes are defined and a grid of the domain is generated and the temperature distribution is then plotted.

5 Results and Discussion

5.1 Temperature Distribution

The following figure shows the temperature distribution obtained with the developed FVM code and FLUENT®:

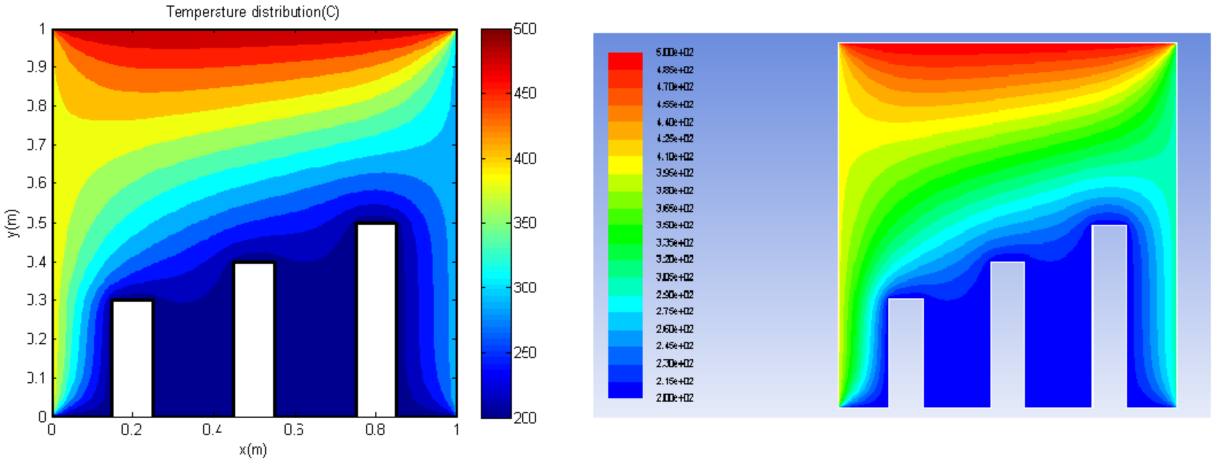

Figure 5: Temperature Distribution from code (left) and FLUENT ® (right)

It can be seen that the temperature distribution from the developed code is similar to the one from FLUENT ®.

5.2 Quantitative Comparison

A comparative analysis of the two solutions was performed by probing the contour plots. The locations of the points were arbitrarily selected and the corresponding temperature values recorded. The FLUENT® results were obtained from CFD-Post. The following table shows the temperature values from the code and ANSYS FLUENT®, at random coordinates.
In order to assess the accuracy of the numerical solution, a quantitative comparison was performed by probing random temperature values within the domain in the two solutions. The data are summarized in the following table:

Table 1: Probed Data Comparison

x (m)	y (m)	FVM temperature (°C)	FLUENT ® temperature (°C)	Percentage difference (%)
0.055	0.135	320.108	320.107	3.0×10^{-4}
0.205	0.795	414.92	414.995	2.3×10^{-2}
0.355	0.595	326.84	326.97	4.0×10^{-2}
0.675	0.725	348.33	348.405	2.15×10^{-2}
0.845	0.915	426.006	426.015	2.0×10^{-5}
0.955	0.825	325.244	325.251	2.15×10^{-3}

The table suggests that the developed program yields accurate results, compared to the FLUENT ® outputs, as the errors are less than 1%.

5.3 Comparison between Iterative Methods

To further assess the performance of the CGM, the solution times of the CGM, the Gauss-Seidel Method, and the Underrelaxation Method are compared:

Table 2: Probed Data Comparison

Iterative Method	Solution Time (sec)
CGM	0.570
Gauss-Seidel (ω=1)	163.72
Underrelaxation (ω=0.8)	243.517

It can be said that the CGM has a significantly fast convergence compared to the other iterative methods considered.

6 Conclusion and Future Work

This paper investigated a two-dimensional, steady-state conductive heat transfer problem using numerical method. The latter implements the FVM and the system of equations is solved using the Conjugate Gradient iterative method. The numerical solution was then compared with the solution from ANSYS FLUENT®, and it was observed that the code is highly accurate and the computation time was small. As future work, it is desired to develop a more generic program that can solve different types of heat transfer problems.

References

[1] D. A. Gismalla. Matlab software for iterative methods and algorithms to solve a linear system. *IJETR*, 2014.

[2] W. Malalasekera and H. K. Versteeg. , *An Introduction to Computational Fluid Dynamics The Finite Volume Method Second Edition*. Pearson Education, 2007.

[3] S. Sickel, M. C. Yeung, and M. J. Held. A comparison of some iterative methods in scientific computing. 2005.