IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

Application of Simulation Method for Analysis of Heat Exchange in Electronic Cooling Applications

D.Ramya[1], Vivek Pandey[2]

Department of Mechanical Engineering, Raghu Engineering College[1]

Associate Professor, Department of Mechanical Engineering, Raghu Engineering College[2]

Abstract: For the purpose of manufacturing cooling devices for electronic applications, the best design is to be considered. For that purpose a trial and error method is not suitable as it traditionally led to lot of time waste and also a waste of resources, energy and manpower. Now we use CAD/CAM tools for the design process. One of the CAD tools is a computer aided design tool with application in electronic cooling circuits. There are few software that can do FEA and even less that can do CFD/Heat transfer simulations. So we want to integrate/combine this CFD/Heat transfer analysis capability in the present software's are commercial, so we want to develop open source integrated CAD/CAM software with CFD/FEA and Heat transfer analysis. In our work we have used computer program developed in MATLAB to simulate the effectiveness of a electronic cooling device like heat exchanger. This can be used to develop an integrated CAD system or software where simulations are embedded in the CAD/CAM software to aid/augmented the design process.

Keywords: Rapid Development, Electronics Components, Typical Cooling Methods, Heat Transfer Process, Personal Digital Assistant (PDA), Designed Temperature Specifications

1. INTRODUCTION

In general ANSYS tools have minimum basic tools of cad i.e., they are not fully extended. Similarly cad tools cannot do analysis. S o we integrate these two tools using open soft ware like octave. As we know that octave is very similar to Matlab and also us have licensed software lead this work to done in Matlab

1.1. **Matlab:** A high-level language and interactive environment for numerical computation, visualization, and programming, MATLAB is an engineering design tool preferred by millions of engineers and scientists worldwide. With MATLAB, you can explore and visualize ideas and collaborate across disciplines to put your ideas into action.

2. GMSH

It is the task of a mesh generator to create node locations and element topology so as to create high quality meshes. Gmsh is a "3D finite element grid generator with a building CAD engine and post- processor. Its design goal is to provide a fast, light and user friendly meshing tool with parametric input and advanced visualization capabilities. Furthermore, Gmsh can be used as a 1–, 2– and 3–dimensional mesh generator for use with Fluidity. Gmsh is developed by Geuzaine and Remacle (2009) and distributed under the terms of the GNU General Public License

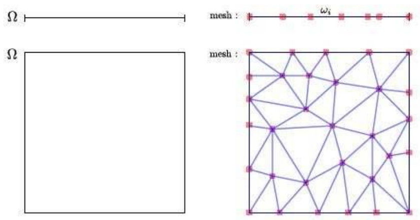

Figure 1: Examples of meshes in one-dimensional and two-dimensional domains.

2.1. Architecture of Gmsh:

Gmsh's architecture is centered around four modules:

1. Geometry
2. Mesh
3. Solver
4. Post-processing

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

As suggested by their names the geometry module is used to define geometrical objects, such as points, lines, surfaces and volumes, while the mesh module is used to create meshes (nodes and element topology).

2.2 Results:

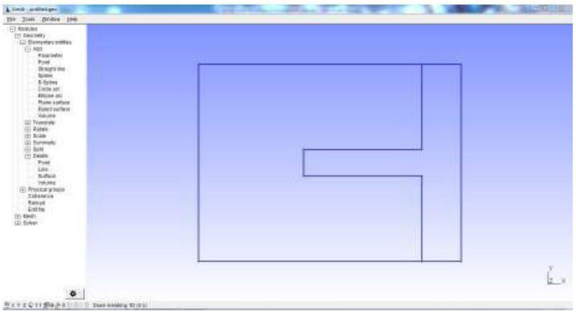

Figure 2: Creating geometry in GMSH

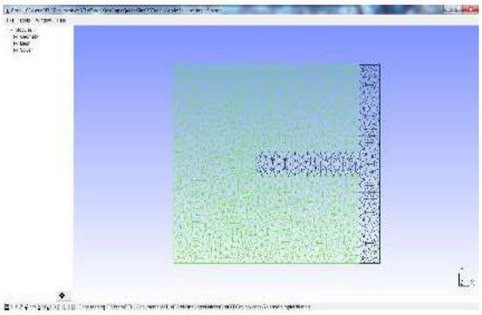

Figure 3: Mesh generated in GMSH

3. MATLAB

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is a high-performance language for technical computing. It integrates computation, visualization,and programming environment Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive s y s t em whose basic data e l ement is an array that d o e s not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering. Main characteristics of MATLAB

3.1. M-File Scripts: A script file is an external f i l e that c o n t a i n s a sequence of MATLAB statements. Script files have a filename extension .m and are often called M-files. M-files can be scripts that simply execute a series of MATLAB statements, or they can be functions that can accept arguments and can produce one or more outputs.

Anatomy of a M-File function:

This simple function shows the basic parts of an M-file.
function f = factorial(n) (1)
% FACTORIAL(N) returns the factorial of N. (2)
% Compute a factorial value. (3)
f = prod(1:n); (4)

The first line of a function M-file starts wi t h the keyword function. It gives the function name and order of arguments. In the case of function factorial, there are up to one output argument and one input argument. Summarizes the M-file function.

4. ANALYSIS OF HEAT EXCHANGE IN ELECTRONIC COOLING APPLICATION

4.1. Problem Definition: This session is going to show how to solve a usual heat conduction problem, which can normally be solved with genericHeatSolver2D except for the fact that conductive medium to be composed of two distinct materials, such as concrete and Styrofoam. Geometry of the domain is depicted in figure below and has been created in millimeters. Thus, we will need to recalculate it to meters. You will see an additional scaling step in the script.

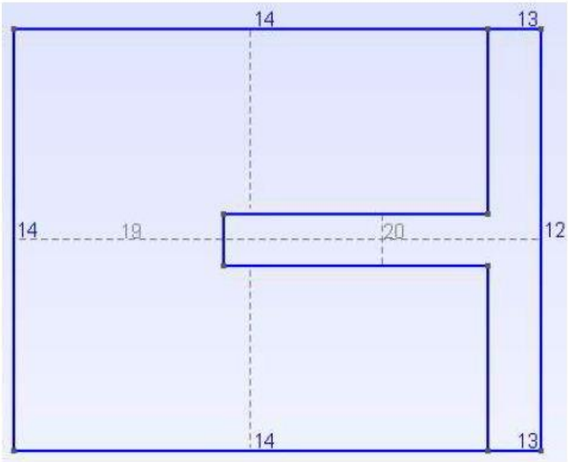

Figure 4: Geometry of the domain

We assume the following boundary conditions:

- Line 12: Constant heat flux of 10 000 W/m^2
- Line 13: Insulated
- Line 14: Heat convection with htc = 20 W/(m2K) and ambient temperature of 293K

The initial condition is a temperature of 293 K everywhere. Thermal conductivity of the Styrofoam equals 0.03 W/(mK) and thermal conductivity of the concrete equals 0.35 W/(mK).

4.2Mesh Preparation: Mesh for our problem has been generated in GMSH. The significant difference, when preparing mesh for a simulation of heat transfer in multiple materials is that the solver must be able to determine, which finite elements represent which material. To make this possible, you have to assign in GMSH two different Physical Surfaces to each of the sub-domains. Ids of the physical surfaces created in GMSH will later be passed to CFD Toolbox thanks to import Mesh GMSH function.

Resulting mesh is shown in figure below. Sub-domain containing concrete has got physical surface id = 89 and sub-domain with Styrofoam physical surface id= 19. We will need these values, when specifying distinct thermal conductivities of each material.

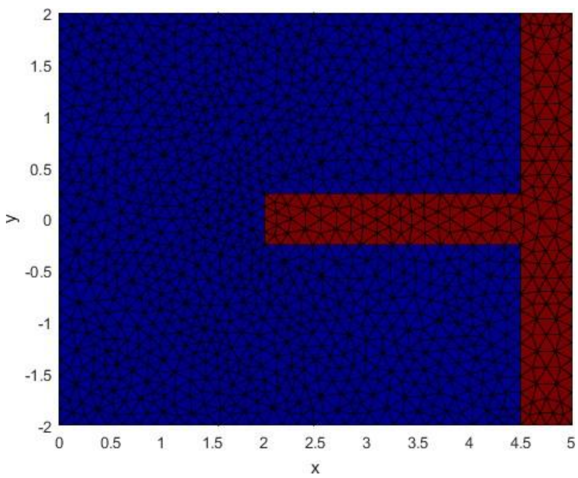

Figure 5: Mesh of the domain

DOI 10.17148/IARJSET.2018.5713

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

4.3Mathematical Model: Mathematical model of a heat conduction problem is Laplace's (diffusion) equation casted below:

$$\nabla \cdot (\lambda \nabla T) = 0$$

This partial differential equation must be supplemented with appropriate boundary conditions, which for our problem are as follows: Insulated boundaries:

$$n \cdot \nabla T = 0$$

At the top and bottom wall we impose a convection boundary condition, known in heat transfer also as third kind boundary condition or Robin boundary condition. The usual engineering formulation of that condition is as follows

$$q_{in} = \alpha(T_{out} - T_{wall})$$

where q_{in} stands for heat flux density entering the solid, α stands for heat transfer coefficient, T_{out} for ambient air or fluid temperature and T_{wall} actual wall temperature.

Contrary to this very popular engineering definition of a heat convection boundary condition, it is recognized as Robin boundary condition described in maths with the following formula:

$$n \cdot \lambda \nabla T + b_2 \cdot T = b_1$$

First term of the above sum represents exactly the incoming heat flux density q_{in}. We only need to redefine two other terms to have consistent definitions. Please note that few transformations lead to the observation that choosing:

$$b_1 = \alpha T_{out}$$

And

$$b_2 = \alpha$$

Yield identical definitions. CFD Toolbox works with the proper definition of Robin boundary condition, so once solving a heat transfer problem, you have to remember to recalculate ambient temperature and heat transfer coefficient. You can also find a corresponding example in help to the imposeScalarBoundaryConditions2D function.
lambdaInNodes = moveDataFromElementsToNodes(p,t,lambda);
displaySolution2D (p,t,lambdaInNodes,'Thermal conductivity[W/(mK)]');

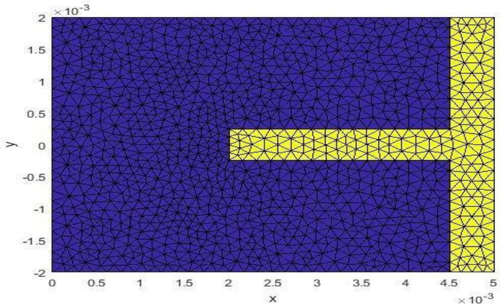

Figure 6: displayMesh2D in MATLAB

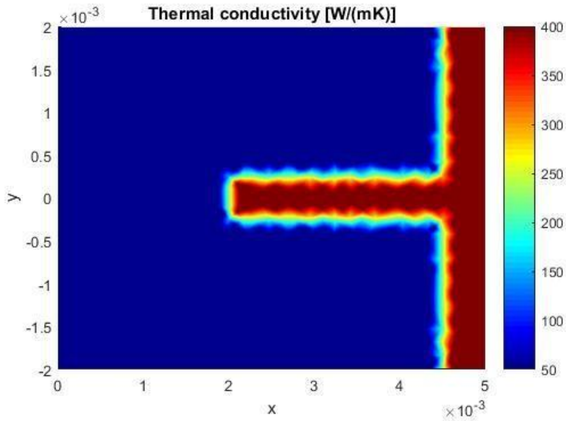

Figure 7: Thermal conductivity of domain in W/Mk

We will now assemble the whole problem matrix and impose convection bound- ary conditions

Assemble problem matrix

[D,F] = assembleDiffusionMatrix2D(p,t,lambda);

Apply boundary conditions

[D,F] = imposeScalarBoundaryCondition2D (p,e,D,F,12,'flux',10000);

[D,F] = imposeScalarBoundaryCondition2D (p,e,D,F,14,'robin',24*253,24);

Solve equations T

Display solution

displaySolution2D(p,t,T,'Temperature [deg C]');

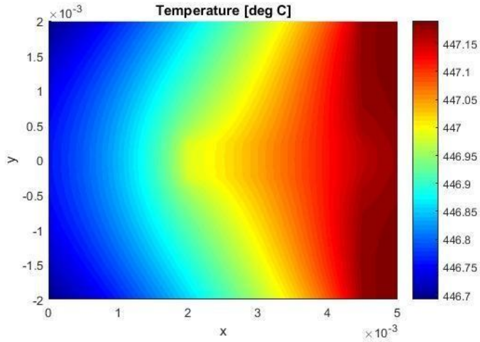

Figure 8: Temperature distribution in the domain in ^{0}c

4.4 Post processing: We will only do basic post processing of results here by checking the balance of heat flux coming into the domain and coming out. To do that, we first compute the heat flux q= −λ∇ T. We compute the gradient (the function returns a matrix with two columns - the x-component of heat flux vector is stored in the first column and the y-component in second column). That's why we have to replicate the values of lambda in each node into two columns (MATLAB will be then able to perform element- by-element multiplication). To copy a 1-column vector lambdaInNodes into matrix of two identical vectors, we use the repmat function. Eventually, we extract the x- component of q and its y-component and pass it to boundaryFlux 2D function to compute total heat flux through a Certain edge Please note that total heat flux on insulated boundaries (id 88), which should be zero is orders of magnitude lower than that on upper and lower boundary. The latter two are equal to each other within discretization and numerical accuracy and show that the total energy balance is conserved.

```
Q= repmat(lambdaInNodes,1,2).*solutionGradient2D(p,t,T);
qx = q(:,1);
qy = q(:,2);
Qin = boundaryFlux2D(p,e,[qx;qy],12)
Qout = boundaryFlux2D(p,e,[qx;qy],14)
Qinsulated = boundaryFlux2D(p,e,[qx;qy],13)
```

CONCLUSION

In general ANSYS tools have minimum basic tools of cad i.e., they are not fully extended. Similarly cad tools cannot do analysis. So we integrate these two tools using open software like octave. As we know that octave is very similar to Matlab and also we have licensed software lead this work to done in Matlab An initial step was taken to integrate designing, analysis and simulating of an industrial component or machine in one designing software. A simple part of a mechanical component was designed in GMSH software. The .msh file obtained from GMSH was exported to the MATLAB. A programming code was developed which was used to solve the present problem. The code was run in MATLAB in order to get the required results The results obtained are shown above. The programming code can be modified as the problem definition changes. So the project says that without using high level software an industrial level analysis can be done with the use of open soft ware. So the project concludes that the motto of project was achieved.

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

REFERENCES

[1]. http://amcg.ese.ic.ac.uk/files/gmsh_tutorial.pdf
[2]. http://gmsh.info/doc/texinfo/gmsh.html
[3]. http://dragonfly.tam.cornell.edu/teaching/mae5230-cfd-intro-notes.pdf
[4]. An Introduction to Computational Fluid Dynamics, The Finite Volume Method H K Versteeg and W Malalasekera, Pearson Education, 2008.
[5]. Computational Fluid Flow and Heat Transfer K.Muralidhar and T.Sundararajan, Narosa, 2007
[6]. Wessel, P. and Smith, W. H. F. (1996). A global self-consistent, hierarchical, high- resolution shoreline database. Journal of Geophysical Research, 101(B4):8741–8743.
[7]. http://www.ee.columbia.edu/introduction-matlab-engineering-students
[8]. www.scribd.com/document/228620580/Introduction
[9]. www.mathworks.com/videos/writing-a-matlab
[10]. https://gitlab.onelab.info/gmsh/gmsh
[11]. http://onelab.info/mailman/listinfo/gmsh
[12]. https://en.wikibooks.org/wiki/MATLAB_Programming
[13]. https://www.mathworks.com/help/matlab/programming-and-data-types.html