International Journal of Computer Applications (0975 – 8887)
International Conference on Quality Up-gradation in Engineering, Science and Technology (ICQUEST-2014)

Computational Analysis and Simulation by CFD from Heat Transfer through Fins

Atul P. Gawai
M.E. Thermal, Mechanical Engineering Department, IBSS COE, Amravati

ABSTRACT

Employing extended surfaces, or 'fins', are, undoubtedly the most widely used method for dissipation of excess heat generated in reciprocating and rotary machine components. Moreover, the amount of heat extracted by a fin from the machine surface depends, to a great extent, on the cross-sectional shape. This research, aims at studying the said dependence of a fin's thermal dissipation capacity on its geometry. For simplicity, the most commonly used cylindrical pin fin, or spine, was used as a standard and the characteristics of three alternative shapes were compared against those of the former. Furthermore, to prove the consistency of the trend, experiments were carried out on two materials- aluminum and copper. Numerical simulation included the CFD analysis of the problem using ANSYS Fluent v13. The results obtained through both the approaches were compared against each other and those available from the surveyed literature. The investigation helps understand how fin shapes affect their behavior and could lend an insight into realizing the utility of each fin from a designer's perspective.

Keywords

Rotary Machine, Cross Sectional Shape, Numerical simulator

1. INTRODUCTION

In the study of heat transfer, a fin is a surface that extends from an object to increase the rate of heat transfer to or from the environment by increasing convection. The amount of conduction, convection, or radiation of an object determines the amount of heat it transfers. Increasing the temperature difference between the object and the environment, increasing the convection heat transfer coefficient, or increasing the surface area of the object increases the heat transfer. Typically fin material has high thermal conductivity and so they are made up of materials like copper, aluminium and iron. Other applications include IC engine cooling, such as fins in a car radiator and compressors. Fins improve heat transfer in two ways as follows:

1. Creating turbulent flow through fin geometry which reduces thermal resistance (Inverse of convective heat transfer co-efficient) through the nearly stagnant film that forms when a fluid flows parallel to a solid surface.
2. Increasing fin density which increases the heat transfer area that comes in contact with the fluid. Fins come in varying shapes and sizes. Different types of fins available are cylindrical, parabolic, conical, trapezoidal, annular, step fin, rectangular etc.

2. COMPUTATIONAL FLUID DYNAMICS

Computational fluid dynamics, abbreviated as CFD, uses numerical techniques and algorithms to solve fluid flow and heat transfer problems. Being a numerical approach to solving problems involving complex differential equations, CFD yields only approximate solutions. However, the estimated responses help the user get an idea of the nature of the solution. Moreover, accuracy of the solution can be varied, as per the requirement and availability of resources, by the user. CFD involves breaking down the working environment, known as control volume, into a one, two or three dimensional grid and computing the values of the equation variables at each of the grid points using discretized forms of the governing differential equations, subject to boundary conditions as specified by the problem. Since the discretized equations are simpler, algebraic equations, it is possible to solve them numerically and therefore, obtaining solutions to such equations is easier and can save much time. CFD problems are solved in the following three fundamental steps:

2.1 Pre-Processing

Pre-processing involves strategizing a design and making preliminary decisions for obtaining the desired solution. It would entail understanding the nature of the problem at hand and deciding the differential equations to be discretized and, subsequently, solved. The simulation generally starts from an initial solution and uses an iterative method to reach the final solution. In addition to creating a geometric abstraction of the experimental apparatus, the user may choose to create his own grid for computation. The finer the grid is chosen, the higher the accuracy of the solution. Determination of spatial and temporal discretization is a crucial part of CFD analysis and dictates the accuracy and, in case of transient problems, the stability of the solutions.

2.2 Processing

This phase involves the actual solving of the discretized equations, using provided parameters in the form of boundary conditions, at each grid point at each time step. The simulation is performed with various possible options for interactive or batch processing and distributed processing.

2.3 Post-Processing

The project aims at determining the effectiveness of CFD as a design tool by attempting to verify obtained data for the thermal diffusion-convection problem are. The problem concerning the project is one of heat transfer and, therefore, the said numerical methods were used primarily to solve the Energy equations for three dimensional, steady-state, diffusion with convective heat transfer, using both, natural and forced drafts. The geometry of the fins and the duct was modeled using CATIA V5 and a standard mesh was generated in Fluent using the Auto-mesh feature.

3. MESHING

Meshing is an integral part of any Computer Aided Engineering simulation process. The more automated mesh tool is the better is the solution. ANSYS provides powerful pre - and post - processing tools for mesh generation. The highly automated meshing tools in ANSYS make it simple to generate complex meshes of the following types:

Tetrahedral, Hexahedral, Pyramid, Prismatic, Arbitrary Polyhedral, Triangular, Quadrilateral

For simpler geometries a quadrilateral/hexahedral meshes can provide high quality solutions as compared to triangular/tetrahedral meshes.

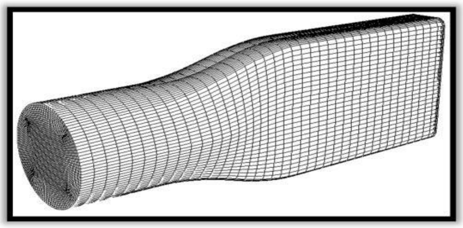

Figure 1: Quadrilateral Mesh

Whereas, for complex geometries quadrilateral/hexahedral meshes show no numerical advantage and hence a triangular/tetrahedral mesh may be used.

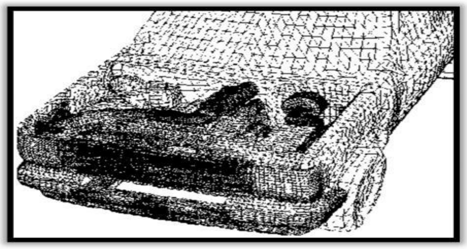

Figure 2: Triangular/Tetrahedral Mesh

3.1 Types of Meshes

3.1.1 Patch Conforming

Patch conforming is a meshing technique in which all faces and their boundaries within a very small tolerance are respected for a given part. Mesh based defeaturing is used to overcome difficulties with complex geometries.

3.1.2 Patch Independent

Patch independent meshing is a meshing technique in which the faces and their boundaries are not necessarily respected unless there is a load, boundary conditions or other scoped object to the faces or edges or vertices. The below figure shows the meshed model of the fins:

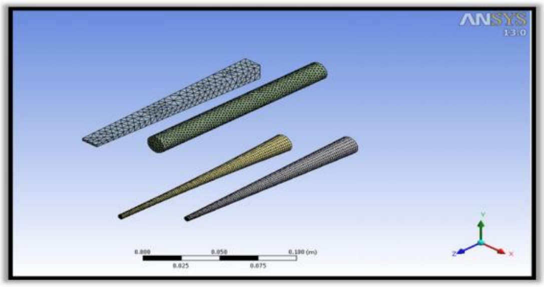

Figure 3: Meshed model of Fins

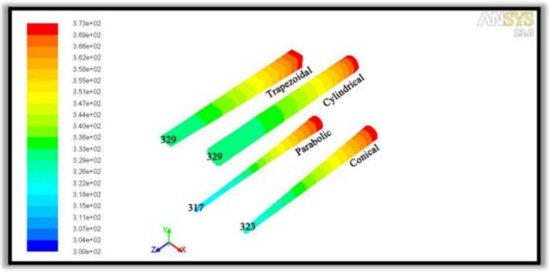

Figure 4: CFD Analysis Temperature Contour - Copper

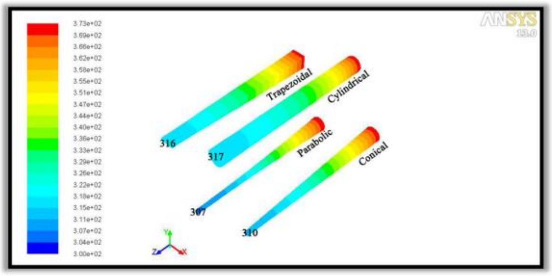

Figure 5: CFD Analysis Temperature Contour - Aluminium

4. PROCESSING AND POST PROCESSING

Here we have considered K-€ turbulence model. This model focuses on the mechanisms that affect the turbulent kinetic energy per unit mass (k). The instantaneous kinetic energy k(t) is the sum of the mean kinetic energy K and the turbulent kinetic energy K=0.5(U2+V2+W2), k=0.5(u2+v2+w2)

K(t)=K+k

€ is the rate of dissipation of the turbulent kinetic energy.

5. FINITE ANALYSIS USING C++

Any fluid flow or heat transfer system is governed by differential equation. In any CFD software when we analyze such systems to obtain the values of important parameters like temperature, velocity, pressure, the software solves these differential equations to display the results. In CFD, there are three methods to solve differential equations namely,

Finite Difference Method.

Finite Element Method.

Finite Volume Method.

5.1 Cylindrical Fin

```
#include<iostream.h>

#include<conio.h>

#include<math.h>

#include<iomanip.h>

void main()

{

clrscr();

int n;// Number of elements

float a[1700];

float b[1700];
```

```
float c[1700];
float d[1700];
float T[1700];
float To;// Ambient temperature
float Tb;// Base temperature
float l;
float h;
float m;
float A;
float P;
float k;
float he;
float D;
cout<<"Enter Length"<<endl;
cin>>l;
cout<<"Enter Number of divisions"<<endl;
cin>>n;
cout<<"Enter fin base temperature"<<endl;
cin>>Tb;
cout<<"Enter ambient temperature"<<endl;
cin>>To;
he=l/n;
cout<<"Enter diameter of the fin"<<endl;
cin>>D;
A=3.142*D*D/4;
P=3.142*D;
h=7;
k=401.5;
m=sqrt((h*P)/(k*A));
float k1 = 1/he;
float k2 = m*m*he/3;
float k3 = k*A/2/he;
int i;
a[1] = 0;
b[1] = k1+2*k2;
c[1] = k3+k2-k1;
d[1] = Tb*k3;
a[n] = k2-k1+k3;
b[n] = k1+2*k2;
c[n] = 0;
d[n] = To*k3;
for (i=2; i<=n-1; i++)// Loop for calculating the co-efficients
{
a[i] = -k1+k2;
b[i] = 2*k1+4*k2;
c[i] = -k1+k2;
d[i] = 0;
}
//TDMA code
for(i=2; i<=n; i++)
{ double x= a[i]/b[i-1];
b[i] = b[i] - x*c[i-1];
d[i] = d[i] - x*d[i-1];
}
T[n] = d[n]/b[n];
for(i= n-1; i>0; i--)
T[i] = (d[i]-c[i]*T[i+1])/b[i];
for( i=1; i<=n; i=i+50)
cout<<"\n Temeperature at node "<<i<<" = "<<T[i];
getch();
}
```

Following are the ‘C++’ codes for fins of varying cross sections. The only change in the code for each fin is that the variation of area is different in each case. Here we coded only cylindrical, conical and parabolic fins due space reason.

5.2 Conical Fin

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<iomanip.h>
void main()
{
clrscr();
int n; //Number of divisions in which we want to divide the fin length
float a[1000];
float b[1000];
float c[1000];
float d[1000];
float T[1000];
float P[1000];
float A[1000];
float h[1000];
float To; //Surrounding temperature
54
float Tb; //Base temperature
float l; //Length of the fin
```

```
float H=7.0; //Co-efficient of convective heat transfer
int i;
float k=250; //Co-efficient of conductive heat transfer.
float he;
float D1;
float Dn;
cout<<"Enter Length"<<endl;
cin>>l;
cout<<"Enter Number of divisions"<<endl;
cin>>n;
cout<<"Enter fin base temperature"<<endl;
cin>>Tb;
cout<<"Enter ambient temperature"<<endl;
cin>>To;
he=l/n;
cout<<"Enter diameter of the fin at the base"<<endl;
cin>>D1;
cout<<"Enter diameter of the fin at the tip"<<endl;
cin>>Dn;
A[1]=3.142*(D1/2)*(D1/2);
A[n]=3.142*(Dn/2)*(D1/2);
P[1]=3.142*D1;
}
a[2]=0;
b[2]=((-2*k*A[2])/(he*he))-(H*P[2]);
c[2]=((k*A[2])/(he*he))+((k*(A[3]-A[1]))/(he*he));
float d1=H*P[2]*To;
float d2=((k*A[2])/(he*he))-((k*(A[3]-A[1]))/(he*he));
d[2]=-(d1)-(d2*Tb); //Co-efficients of the 1st equation
a[n]=(-2*k*A[n]);
b[n]=H*P[n];
c[n]=0;
d[n]=(H*P[n]*To)-(2*k*A[n]*To); // Co-efficients of the last equation
for(i=3; i<=n-1; i++) //Loop for calculating the intermediate co-efficients
{
a[i]=((k*A[i])/(he*he))-((k*(A[i+1]-A[i-1]))/(he*he));
b[i]=-((2*k*A[i])/(he*he))-(H*P[i]);
c[i]=((k*A[i])/(he*he))+((k*(A[i+1]-A[i-1]))/(he*he));
d[i]=-(H*P[i]*To);
}
float B[1000];
float G[1000];
B[2]=b[2];
//TDMA code
G[2]=d[2]/B[2];
for(i=3; i<=n; i++)
{
B[i]=b[i]-(a[i]*c[i-1]/B[i-1]);
G[i]=(d[i]-a[i]*G[i-1])/B[i];
}
T[n] = G[n];
for(i=n-1; i>=2; i--)
T[i] = G[i]-((c[i]*T[i+1])/B[i]);
T[1]=Tb;
for(i=1; i<=n; i=i+50)
cout<<"\n Temperature at node "<<i<<" = "<<T[i]<<"( "<<T[i]-273<<" )";
getch();
}
```

5.3 Parabolic Fin

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<iomanip.h>
void main()
{
clrscr();
int n;
float a[1000];
float b[1000];
float c[1000];
float d[1000];
float T[1000];
float P[1000];
float A[1000];
float y[1000];
float To;
float Tb;
float l;
float H=7.0;
int i;
float k=250;
float he;
cout<<"Enter Length"<<endl;
```

```
cin>>l;
cout<<"Enter Number of divisions"<<endl;
cin>>n;
cout<<"Enter fin base temperature"<<endl;
cin>>Tb;
cout<<"Enter ambient temperature"<<endl;
cin>>To;
he=l/n;
for(i=1; i<=n; i++) //Loop for calculating the area and perimeter at each node
{
y[i]=((0.000244*he*(i-1)*he*(i-1))-(0.073*he*(i-1))+(7.5))*(0.001);
A[i]=3.142*y[i]*y[i];
P[i]=2*3.142*y[i];
}
a[2]=0;
b[2]=((-2*k*A[2])/(he*he))-(H*P[2]);
c[2]=((k*A[2])/(he*he))+((k*(A[3]-A[1]))/(he*he));
float d1=H*P[2]*To;
float d2=((k*A[2])/(he*he))-((k*(A[3]-A[1]))/(he*he));
d[2]=-(d1)-(d2*Tb);
a[n]=(-2*k*A[n]);
b[n]=H*P[n];
c[n]=0;
d[n]=(H*P[n]*To)-(2*k*A[n]*To);
59
for(i=3; i<=n-1; i++)// Loop for calculating the co-efficients of each equation
{
a[i]=((k*A[i])/(he*he))-((k*(A[i+1]-A[i-1]))/(he*he));
b[i]=-((2*k*A[i])/(he*he))-(H*P[i]);
c[i]=((k*A[i])/(he*he))+((k*(A[i+1]-A[i-1]))/(he*he));
d[i]=-(H*P[i]*To);
}
float B[1000];
float G[1000];
//TDMA code
B[2]=b[2];
G[2]=d[2]/B[2];
for(i=2; i<=n; i++)
{
B[i]=b[i]-(a[i]*c[i-1]/B[i-1]);
G[i]=(d[i]-a[i]*G[i-1])/B[i];
}
T[n] = G[n];
for(i=n-1; i>=2; i--)
T[i] = G[i]-((c[i]*T[i+1])/B[i]);
T[1]=Tb;
for(i=1; i<=n; i=i+50)
cout<<"\n Temperature at node "<<i<<" = "<<T[i]<<"( "<<T[i]-273<<" )";
getch();
}
```

6. RESULTS

Table 1: Finite Analysis Temperature

	Copper			Aluminium		
	50 mm	100 mm	150 mm	50 mm	100 mm	150 mm
Cylindrical	360	339	323	349	327	311
conical	372	372	369	372	371	369

6.1 Deviation and Observation From

6.1.1 Simulated Results

Following are the possible causes that could explain these deviations to simplify calculations, the following approximations and assumptions had to be made:

1. In all C++ analysis, the heat transfer was assumed to follow a 1D diffusion-convection regime, while in fact, the mode of heat transfer is a 3D Diffusion-convection model. This means that the radial variations, in case of pin fins, and transverse variation, in case of the trapezoidal fin, of temperature have been neglected. However, this approximation may be justified on the basis of the fact that the fin dimensions are small compared to the larger values of thermal conductivities and diffusivity of the two materials.
2. Furthermore, in the FEA analysis of the cylindrical fin, the fin was assumed to be made up of a number of linear elements. This has certainly cost us some accuracy and using a higher order element in the C++ code would make it more efficient.
3. Also, in the C++ analysis, the surface heat transfer coefficient was assumed constant over the entire length of the fin, which in practice is not the case. This further leads to errors in calculating the temperatures at each point.

7. CONCLUSION

1. From the results, it is observed that for the same fin material, the trapezoidal fin experienced the least temperature drop over its length, while the parabolic pin fin consistently had the largest temperature drop.
2. Also, as heat flux entering each fin is same owing to their equal base area; parabolic fins are able to achieve least tip temperature for the same material. Hence, the parabolic fin is the most efficient in dissipating heat to its surroundings.

3. Due to their greater thermal conductivity, copper fins are, in general, more efficient than aluminium fins.
4. However, a designer must consider the trade-off between the greater conductivity and the larger density of copper before selecting it to fabricate fins.

8. REFERENCES

[1] Heat Transfer Simulation by CFD from Fins of an Air Cooled Motorcycle Engine under Varying Climatic Conditions - Pulkit Agarwal, Mayur Shrikhande.

[2] Fundamentals of Heat & Mass Transfer - Incropera, DeWitt, Bergman, Lavine

[3] Heat & Mass Transfer - D. S. Kumar

[4] Heat transfer through extended surfaces subject to variable heat transfer coefficient Esmail M.A. Mokheimer

[5] Experimental and computational analysis and Optimization for heat transfer through fins with different types of notch - S.H. Barhatte, M. R. Chopade, V. N. Kapatkar Journal of Engineering Research and Studies,M11

[6] Optimization of the Design Process for an Automotive Air Cooling System –Dhritiman Subha Kashyap, University of California San Diego, Dec 2007

[7] Performance of annular fins with different profiles subject to variable heat transfer coefficient - Esmail M.A. Mokheimer.