EXTENDED SURFACES: FIN ANALYSIS - A CASE STUDY

In many heat transfer applications, it is desirable to increase the surface area that is available for the heat transfer process; this is particularly true when one desires to dissipate heat to a low conductivity medium such as air. Often the fin geometry may be quite complex, and analytical solutions may not be available to describe the heat transfer process. Instead, one needs to resort to various numerical techniques. In this module, we will analyze a two-dimensional fin with temperature-independent properties. We will start by formulating the problem as a one-dimensional problem. The fin’s geometry is sufficiently simple so that this problem can be solved analytically. The solution requires, however, the use of special functions of mathematical physics such as Bessel functions. Although you encountered these functions in your mathematics courses, you may not be fluent in their use. Here Maple™ will come to our aid and help us solve the equations. Unfortunately, many fin problems are not amenable to analytic solution. Hence, we will proceed and solve the same problem numerically and then compare the numerical and analytical solutions. The purpose of this exercise is to help us build confidence in the numerical solution. Subsequently, we will develop a back of the envelope approximate solution, known as an integral solution, and compare it again with the analytic solution to assess how well it works. Finally, we note that the fin analysis is based on the assumption that the temperature is nearly uniform at each cross-section of the fin. In order to evaluate how accurate this approximation is and when it can be used, we will resort to a two-dimensional, finite element solution. This will give us the opportunity to get acquainted with finite elements software. You are very likely to use such software in your workplace.

EXAMPLE: STRAIGHT FIN OF TRIANGULAR SHAPE

Fig. 1: A triangular, two-dimensional fin of length L and base height b dissipates heat to an ambient at temperature, $T_{\infty}$. The base temperature is $T_b$ and the heat transfer coefficient between the fin and the ambient is h.

In many applications (i.e., avionics), it is important to minimize the mass of the fin. When one is seeking an optimal fin, the question arises as to whether a mass advantage can be gained by using a shape other than rectangular. Here we consider a two-dimensional fin with a triangular cross-section. See the figure below. The fin’s base has height b, its length is L, and its width (the dimension that is
perpendicular to the page) is \( W \). \( W \gg b \) is large. The fin’s thermal conductivity is \( k \). For mathematical convenience, we locate the origin of the coordinate \( (x) \) at the fin’s tip. The fin’s base \( (x=L) \) is maintained at a constant, uniform temperature \( T_b \). The fin interacts thermally with an ambient at \( T_\infty \) through a heat transfer coefficient \( h \). We wish to find how much heat flows into the fin through its base.

To make the example more concrete, consider the outer surface of an aircraft oil heater fitted with straight fins with triangular profiles. The heater has a uniform temperature of \( T_b = 150 \text{C} \). The ambient temperature is \( T_\infty = 19 \text{C} \). The heat transfer coefficient is estimated to be \( h = 500 \text{W/m}^2\cdot\text{K} \); the material’s thermal conductivity is \( k = 26 \text{W/m}\cdot\text{K} \); the fin’s length is \( L = 0.01 \text{m} \); and \( b = 0.003 \text{m} \). We define the non-dimensional group \( m = 2hL^2/kb = 1.28 \). The largest Biot number, \( Bi = hb/k = 0.06 \) is much smaller than unity. Thus one is justified in assuming that the temperature of each cross-section is nearly uniform.

This is a steady-state heat transfer problem. An energy balance leads to the second-order, ordinary differential equation for the temperature:

\[
\frac{d}{dx} \left( A(x) \frac{dT}{dx} \right) = \frac{hP}{k} (T - T_\infty),
\]

(1)

where \( P \) is the fin’s perimeter and \( A(x) = bW \frac{x}{L} \) is the cross-sectional area. When \( W \gg b \), the perimeter \( (P) \) is approximately \( P = 2W \). It is convenient to introduce non-dimensional quantities. Let \( \xi = x/L \) be the non-dimensional length and \( \theta(\xi) = \frac{T(\xi)-T_\infty}{T_b-T_\infty} \), the non-dimensional temperature. Accordingly, \( 0 \leq \xi \leq 1 \) and \( 0 \leq \theta \leq 1 \). In terms of these non-dimensional variables, equation (1) can be rewritten as

\[
\xi \frac{d^2 \theta}{d\xi^2} + \frac{d \theta}{d\xi} - m \theta = 0,
\]

(2)

with the boundary conditions \( \theta(0) \) finite and \( \theta(1) = 1 \). You may be tempted to replace the boundary condition at \( \xi = 0 \) with \( \frac{\theta(0)}{d\xi} = 0 \); but as we shall show shortly, this is incorrect. In the above, \( m = 2hL^2/(kb) \).

We wish to compute the temperature distribution in the fin and the heat flow through the fin’s base. This problem is sufficiently simple as to facilitate an exact solution. Many fin problems cannot,
however, be solved exactly. In this example, we will obtain both exact and numerical solutions and compare the two.

**EXACT SOLUTION**

If you were paying attention in your math classes, you may recognize equation (2) as a modified Bessel equation. In other words, the solution for this equation is given in the form of modified Bessel functions. These are special functions of mathematical physics and they are tabulated just as trigonometric functions are. Fortunately, various mathematical software programs such as Maple, Mathematica, and Matlab are acquainted with special functions. Here, I will use Maple to help me obtain the analytical solution.

First, I insert the equation into Maple’s worksheet.

\[
\text{> fin_eq:=xi*diff(theta(xi),xi$2)+diff(theta(xi),xi)-m*theta(xi)};
\]

The command, \text{diff}, implies differentiation and \text{xi$2$} means that the differentiation needs to be carried out twice.

Next, I notify Maple that \((m)\) is strictly positive.

\[
\text{> assume(m>0);}
\]

Note: in the succeeding output, “\(m\)” will now appear as “\(m~\)” to denote that this variable has been constrained.

Now, Maple is ready to solve the equation.

\[
\text{> sol:=dsolve(\{fin_eq=0, theta(0)=finite, theta(1)=1\}, theta(xi));}
\]

\[
sol := \theta(\xi) = \frac{\text{BesselI}(0, 2\sqrt{m~\xi})}{\text{BesselI}(0, 2\sqrt{m~})}
\]
The solution is obtained in the form of modified Bessel functions. In a more familiar notation, \( \text{BesselI}(0,x) = I_0(x) \). The equation actually has two solutions, \( I_0 \) and \( K_0 \). The second solution is the modified Bessel function of the second kind and of order zero \( (K_0) \). This solution \( (K_0) \) was eliminated because it is unbounded (goes to infinity) at \( \xi = 0 \), which is not plausible.

For Maple to be able to use \( \theta(\xi) \), we need to do one more operation:

\[
> \text{assign(sol)}:
\]

Let’s now take a look at the temperature profile:

\[
> \text{plot(subs(m=1.28, theta(xi)), xi=0..1, labels=['xi','theta'])};
\]

![Fig. 2: The temperature distribution \( \theta(\xi) \) as a function of \( \xi \) as predicted by the exact solution. \( m=2hL^2/kb=1.28 \).](image)

Witness that the temperature decreases from magnitude one at the base to the tip temperature 0.363.

\[
> \text{tip_temp:= 1/BesselI(0,2*sqrt(m))};
> \text{evalf(subs(m=1.28,tip_temp))};
\]

\[.3632715228\]
We would expect the tip temperature to decrease as the length of the fin (L) increases, the heat transfer coefficient (h) increases, and the fin’s thermal conductivity (k) decreases. All the above trends are reflected in (m) increasing. Plot the tip temperature as a function of (m) and see for yourself.

Maple can also be used to obtain the Frobenius solutions to the ODE (2), which gives the series expressions for I_0 and K_0. More on that later.

NUMERICAL SOLUTION

In the previous section, we computed the exact solution for the triangular fin problem. The exact solution has the advantage that it provides us with an explicit relationship between various parameters that we can use for optimization purposes. Unfortunately, in many cases of practical interest, an exact solution is not attainable. When an exact solution is not feasible, we resort to numerical techniques. Here, we will use both Maple and Matlab to solve the triangular fin numerically. Since an exact solution is available, we will be able to compare the numerical results with the exact ones. Before we can go to the computer and ask it to solve this problem, we need to bring it to a format appropriate for a numerical solution.

The task is to solve equation (2) with the appropriate boundary conditions. Witness that the boundary conditions are specified at the two ends of the interval. Such a problem is referred to as a two-point boundary value problem. Two-point boundary value problems pose a difficulty because numerical solvers for ordinary differential equations such as the Runge Kutta techniques available in Maple and Matlab require that all boundary conditions be specified at the same point. When boundary conditions are specified at two different points, in general, one needs to resort to either shooting techniques (in which one provides an intelligent guess for the missing boundary condition) or finite element/finite difference methods. Our problem is, however, linear and we can overcome the difficulty by using superposition.

Briefly, we will decompose the unknown function \( \theta(\xi) \) into the sum of two (unknown) functions:

\[
\theta = C_1 \theta_1 + C_2 \theta_2.
\]  

Each of the functions \( \theta_1 \) and \( \theta_2 \) is required to satisfy the differential equation (2). We have a fair amount of freedom in specifying the boundary conditions for \( \theta_1 \) and \( \theta_2 \). For example, we choose:
Numerical Solutions to ODEs and PDEs via Maple and Matlab - A Heat Transfer Application

\[ \theta_1(1) = 1 \quad \frac{d\theta_1(1)}{d\xi} = 0 \]
\[ \theta_2(1) = 0 \quad \frac{d\theta_2(1)}{d\xi} = 1 \]  
(6)

The constants \( C_1 \) and \( C_2 \) are chosen in such a way as to satisfy the boundary conditions of the original problem. The first boundary condition, \( \theta(1)=1 \), requires that \( C_1=1 \). \( C_2 \) needs to be determined from the second boundary condition at \( \xi=0 \).

The second boundary condition requires that \( \theta(0) \) be finite, and it is a bit too abstract for direct implementation in a numerical procedure. To obtain an alternative statement, we examine equation (2) carefully. We can argue on physical grounds that the second derivative \( \frac{d^2\theta}{d\xi^2} \) at \( \xi=0 \) is finite. Therefore, we see from equation (2) that at \( \xi=0 \), \( \frac{d\theta}{d\xi} - m\theta = 0 \). This provides us with the missing boundary condition.

Once we have computed \( \theta_1(\xi) \) and \( \theta_2(\xi) \), we will obtain \( C_2 \) from:

\[ \frac{d\theta_1(0)}{d\xi} + C_2 \frac{d\theta_2(0)}{d\xi} - m^{*} (\theta_1(0) + C_2 \theta_2(0)) = 0 \]  
(7)

NUMERICAL SOLUTION WITH MAPLE

Before carrying out the numerical process, we need to specify \( m \). Let’s select \( m=1.28 \) and substitute it in the differential equation:

\( \text{> fin_eq1:=subs(m=1.28,xi*diff(theta1(xi),xi$2)+diff(theta1(xi),xi)-m*theta1(xi)):} \)

\( \text{> s1:=dsolve({fin_eq1=0,theta1(1)=1,D(theta1)(1)=0},theta1(xi), type=numeric, output=listprocedure);} \)
(t1) stands for $\theta_1$.

> t1:=subs(s1,theta1(xi)):

Since the numerical solver detects an apparent singularity at $\xi=0$, we will not be able to compute function values at $\xi=0$; but we can get as close as we wish. Here I assume that $\xi=0.0001$ is close enough.

> t1(.0001);

18.14103233205950

> dt1:=subs(s1,diff(theta1(xi),xi)):

Next we solve the problem for $\theta_2$. In Maple, we refer to $\theta_2$ as t2.

> fin_eq2:=subs(m=1.28,xi*diff(theta2(xi),xi$2$)+diff(theta2(xi),xi)-m*theta2(xi)):

> s2:=dsolve({fin_eq2=0,theta2(1)=0,D(theta2)(1)=1},theta2(xi),type=numeric, output=listprocedure);

> t2:=subs(s2,theta2(xi)):

> t2(.0001);

-21.33436087188668

> dt2:=subs(s2,diff(theta2(xi),xi)):

> C2:= solve(dt1(.0001)+C2*dt2(.0001)-1.28*(t1(.0001)+C2*t2(.0001))=0, C2);

C2 := .8332900642

> t_num:=t1+C2*t2:

> t_num(.0001);

.36331805

The numerical results are in excellent agreement with the analytical ones. See the table below.
NUMERICAL SOLUTION WITH MATLAB

First, I created a "m-file" for the differential equations. The second order equation must be rewritten as a set of two first order differential equations. In order to assure that the dependent variables are evaluated at the same x locations, I integrate the two equations simultaneously (although the equations are not coupled and they could have been integrated individually). Below, I used the following notation: y(1)=θ₁, y(2)=dθ₁/dξ; y(3)=θ₂, and y(4)=dθ₂/dξ. Save the m-file under the name fin4.m in your own directory. Here I assume that you saved this file on a floppy disk inserted in drive A. Before you can load the file into the Matlab session, you will need to change the current directory.

>> cd A:

function dy = fin4(x,y)
dy = zeros(4,1);     % a column vector
dy(1) = y(2);
dy(2) = (y(2)+1.28*y(1))/(1-x);
dy(3) = y(4);
dy(4) = (y(4)+1.28*y(3))/(1-x);

m-file containing the description of the differential equations written as a set of first order differential equations.
Since Matlab expects the initial conditions to be given at the beginning of the interval, I changed the independent coordinate ($\xi$) to $x=1-\xi$. Now, the fin's base is located at $x=0$ (rather than $\xi=1$) and the fin's tip is at $x=1$. Witness that the equations appear to be singular at $x=1$ and we can anticipate numerical difficulties as $x$ approaches 1. We will, therefore, ask Matlab to solve the equations in the interval $[0, 0.9999]$. The following commands were executed in the Matlab work space.

```matlab
» [x, y] = ode45('fin4', [0 .9999], [1; 0; 0;1]);
```

In the above, fin4 is the name of the m-file containing the differential equations, $[0 .9999]$ is the solution interval, and $[1; 0; 0;1]$ are the initial conditions at $x=0$. The vector $x$ contains the $x$-values at which the temperature was computed. $y$ is a four-column matrix. The number of rows in $y$ is the same as the length of $x$. The columns in $y$ correspond to $y(1)$, $y(2)$, $y(3)$, and $y(4)$.

For example, in our calculation, the length of $x$ is:

```matlab
» length(x)
ans = 109
```

The last entry corresponds to the $x=0.9999$.

```matlab
» x(109)
ans = 0.9999
```

Witness that $y$ is not given explicitly as a function of $x$; but $y(1,i)$ is the value of $y1$ at $x(i)$. Now, we are ready to calculate $C$ (equation 7).

```matlab
» c=-(y(109,2) + 1.28* y(109,1))/ (y(109,4)+1.28*y(109,3))
c =  -0.8333
```

Next, we define two vectors, temp and flux, to store the temperature and flux data.

```matlab
» temp=y(:,1)+c*y(:,3);
» flux=y(:,2)+c*y(:,4);
```
The tip's temperature is:

```
» temp(109)
ans = 0.3633
```

Of course, it would be interesting to compare Matlab's numerical solution with the exact one. We have two different ways of implementing the exact solution in Matlab; we can either call Maple commands from Matlab or we can compute the exact solution directly. I choose here the latter. Recall that in Matlab we used the transformation $x=1-\xi$.

```
>> xx=0:0.1:1 % the vector xx consists of the x values at which the exact solution will be computed.
>> t_exact= besseli(0, 2*sqrt(1.28*(1-xx))) / besseli(0, 2*sqrt(1.28)) % t_exact is a vector containing the exact solution.
>> plot(1-x,temp,1-xx,t_exact,'o')
```

In the figure below, the solid line and the circles correspond, respectively, to Matlab's numerical solution and the exact solution. The agreement appears to be very good.

Comparison between Matlab's numerical solution (solid lines) and the exact solution (circles). The temperature is depicted as a function of $\xi$. 
FINITE ELEMENT SOLUTION

Thus far, we took advantage of the fact that the Biot number is small and that the process is essentially one-dimensional. But how good is this approximation? In this section, we will use finite elements to construct a two-dimensional solution for the fin problem. To this end, we will use Matlab’s pde (partial differential equation) toolbox. As a side benefit, this section will introduce you to Matlab’s pde toolbox.

Let’s start by specifying the two-dimensional problem. We are using here \( L=0.01\text{m} \); \( b=0.003\text{m} \); \( h=500\text{W/m}^2\text{-K} \); and \( k=26\text{W/m-K} \). The x-coordinate will be aligned along the length of the fin. Taking advantage of symmetry, we will consider only the upper half of the fin. All the equations and boundary conditions are written in a dimensionless form with \( L \) being the length scale and \( T_b - T_\alpha \) the temperature scale. This is consistent with what we have used in our one-dimensional analysis. The non-dimensional length \( L^* = 1 \) and the base height \( b^* = 0.3 \).

We wish to solve the time-independent, heat conduction (Laplace) equation with uniform, temperature-independent properties.

\[
\left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \theta = 0 \quad (0 \leq x \leq 1, \quad 0 \leq y \leq \frac{b}{L} x)
\]

The boundary conditions are:
(i) At the fin’s base: \( \theta(1,y)=1 \)
(ii) The symmetry condition: \( \left( \frac{\partial \theta}{\partial y} \right)_{y=0} = 0 \) at \( y=0 \).
Numerical Solutions to ODEs and PDEs via Maple and Matlab - A Heat Transfer Application

(iii) The convective (Robbins) condition at the fin’s surface:

\[- \left( \hat{n} \bullet \nabla \theta \right) \bigg|_{x=b/L} = Bi_L \theta \left( x, \frac{b}{L} x \right),\]

where $\hat{n}$ is a unit vector normal to the surface, $\nabla$ is the gradient operator, and $Bi_L = \frac{hL}{k} = 0.192$.

In order to get the toolbox’s User Graphical Interface, type in Matlab’s work space,

```
>> pdetool
```

and hit Enter. The pde toolbox GUI will open. You should see a window similar to the one shown below.
Below, I sketch the major steps that are needed to solve the two-dimensional fin problem. From time to time, you may need to refer to the reference manual. We use the OPTIONS pull down menu to specify a convenient coordinate system. Subsequently, using the free-hand drawing tool, we draw the fin. See the figure below.

From the over bar menu, select a generic equation type. Now, we are ready to specify the boundary conditions. Click on the boundary button (the one with the omega sign). The triangle’s edges will turn red. By double clicking on each of the edges, we will be able to specify the appropriate boundary conditions. Let’s start with the triangle’s hypotenuse. Select a Neumann type boundary condition and set $g=0$ and $q=(Bi_h)=0.192$. This is the convective boundary condition.
Second, we will specify that the fin’s base temperature has magnitude $u=1$. To this end, we set in the window below $h=r=1$. The $h$ above is an internal Matlab variable, and it has nothing to do with the heat transfer coefficient.

After clicking on "OK," double click on the bottom boundary (the symmetry line) and select Neumann type boundary condition. Select $g=q=0$ to set the normal derivative on the axis of symmetry equal to zero.

Now, specify the coefficients in the differential equation. Click on the PDE button and set $c=1$ and $a=f=0$. $c=1$ represents the dimensionless thermal conductivity and $f=0$ specifies the absence of a heat source. The term $a*u$ does not play a role in our problem, and it is suppressed by specifying $a=0$. 
Next click on the mesh button (the one with a triangle on it) and setup the finite element mesh. Refine the mesh three times. The figure below depicts the meshed fin.

In order to solve the equation, click on the solution button (the equal sign) and Matlab will compute the temperature distribution and generate a plot in which the different colors represent different
temperatures.

You can select various options in PARAMETERS under the PLOT menu. For example, I directed Matlab to plot contour lines. In our case, these are the constant temperature lines or isotherms. Witness that the constant temperature lines are nearly straight lines and they are perpendicular to the x-axis. This behavior is consistent with the one-dimensional approximation that we used earlier.

We can also generate a three-dimensional plot depicting the temperature distribution as the surface $\theta(x, y)$.

One very important advantage of the PDE toolbox is that it allows us to export the computational results to Matlab and then manipulate and analyze the data in ways that are not available in the GUI. Many of the pull down menus in the GUI have an "export" option. When you select the export option, a dialogue box will open. PDE toolbox will display the default names of the various variables and will give you the option to change them. Once you press "OK" the information will be exported into the Matlab workspace, and it will be stored under your chosen variable names. Here, I use the default variables.

Below is my Matlab session. The comments next to the Matlab commands clarify what I am doing.
Numerical Solutions to ODEs and PDEs via Maple and Matlab - A Heat Transfer Application

» xx=0:0.1:1; % define a vector xx. The vector xx stores x values ranging from 0 to 1 with increments of 0.1.

» u_center_line=tri2grid(p,t,u,xx,0); % tri2grid is an interpolating function that takes as arguments the element information and the function value and interpolates them to the specified x and y value. Here, we interpolate u to the x values specified by the vector xx and y=0 (the fin's axis of symmetry).

» t_exact= besseli(0, 2*sqrt(1.28*xx)) / besseli(0, 2*sqrt(1.28)); % It would be interesting to compare the finite elements results with the one-dimensional solution. t_exact is the one-dimensional solution evaluated at the same x values specified by xx.

» plot(xx,u_center_line,xx,t_excat,'o') % To compare the finite element and one-dimensional solutions, both are plotted on the same graph with the finite element solution depicted with the (default) solid line, and the one-dimensional solution depicted with circles.

Fig: The temperature distribution along the fin's axis is depicted as a function of x. The solid line and the circles correspond to the 2-D, finite element solution and the one-dimensional analytic solution.

The fin's tip temperature:

» tip_temperature=tri2grid(p,t,u,0,0)

tip_temperature =

0.3606
This value can be compared with 0.363 predicted by the one-dimensional analysis.

The figure shows an excellent agreement between the two-dimensional finite element solution and the one-dimensional approximation, illustrating the merits of the one-dimensional fin analysis. Of course, the fin analysis is justified here because of the smallness of the Biot number that leads to nearly uniform temperature at various x-locations. In other words, the temperature depends strongly on x but is a weak function of y. To see this more clearly, we will plot below the temperature at a fixed x-location (i.e., 0.75) as a function of y.

yy=0:0.005:0.075; % define a vector yy. The temperature will be computed at the specified yy locations, i.e., at y values starting at zero and incremented by 0.005 until y reaches the value of 0.075.

» uy=tri2grid(p,t,u,0.5,yy); % uy represents the temperature evaluated at x=0.5 (at the fin's mid length) and at the specified yy values.

» plot(yy,uy)

The plot depicts nearly a straight line suggesting that θ(0.5,y) is independent of y. This is slightly misleading. The temperature depends on y but weakly so. In order to see this dependence, we need to zoom on the figure.
It would be illuminating to take a look at the direction of the heat flux. To this end, we go back to the GUI and indicate in the PARAMETERS dialogue box that we wish to see the vector field.

Finally, another quantity of interest is the heat flux through the fin’s base.
Numerical Solutions to ODEs and PDEs via Maple and Matlab - A Heat Transfer Application

>> [fx fy] = pdecgrad(p, t, c, u); % calculate the temperature gradient at the elements’ centers

>> fxn = pdeprtni(p, t, fx); % interpolate the gradient df/dy to the nodal points

>> fx0 = tri2grid(p, t, fxn, 1, 0); % interpolate the gradient to x=0 (the fin’s base).

fx0 =

0.8109

This can be compared with the results of the one-dimensional calculations, 0.8333.

SUMMARY

In this simple example, we had an opportunity to observe various tools of analysis ranging from one-dimensional fin analysis to two-dimensional finite element numerical solution. There is a clear advantage in having an analytic solution since one can obtain general, closed form expressions that can be subsequently used in optimization studies. In contrast, each numerical solution is specific to a particular combination of variables. In our example, given the smallness of the Biot number, the two-dimensional calculations and the one-dimensional results were in excellent agreement. This agreement will deteriorate as the Biot number increases. Check this for yourself.