NON-ITERATIVE SOLUTION OF ORDINARY AND PARTIAL DIFFERENTIAL EQUATIONS USING SPREADSHEETS

Ahmad Fakheri
Department of Mechanical Engineering
Bradley University

Mohammad H. Naraghi
Department of Mechanical Engineering
Manhattan College

Abstract

Typically, in a basic course, the equations are derived and the solutions are presented in tabular or chart from. Spreadsheets provide an attractive option, as most students have access to and are familiar with their use. In this paper, a classic algorithm, called Thomas algorithm, used for the solution of set of algebraic equations whose coefficient matrix is tri-diagonal, is adopted to spreadsheets. It is used to obtain solutions to a number of classical problems in fluid mechanics and heat transfer non-iteratively, and in some cases where the governing equations are non-linear via some iterations. Without needing much programming skills, or needing to learn specialized software, undergraduate students can use this approach and easily obtain the solution to many otherwise difficult problems and study the impact of different parameters.

Nomenclature

G	irradiation
h	heat transfer coefficient
k	thermal conductivity
l	straight fin length
L	characteristic length
p	fin circumference
r	radius (radial coordinate)
r_i	inner radius of an annular fin
r_o	outer radius of an annular fin
t	time and straight fin thickness
t^*	Fourier Number, $t^* = \frac{\alpha t}{L^2}$
T	temperature
T^*	$T^* = \frac{T - T_\infty}{T_b - T_\infty}$ or $\frac{T - T_\infty}{T_i - T_\infty}$
T_b	base temperature
W	straight fin width
x	axial coordinate
x^*	$x^* = \frac{x}{L}$

Greek symbols

ε	emissivity
σ	Stefan Boltzmann constant

Introduction

Mathematical formulation of physical phenomena often results in ordinary or partial differential equations. These equations typically do not have analytical solutions or the solution methods are beyond the scope of undergraduate courses. Typically, in a basic course, the equations are derived and the solutions are presented in tabular or chart form.

Numerical solutions are an alternative, often the only option, for obtaining solutions and cover a number of techniques that can be used to obtain the solution of ordinary and partial differential equations. Availability of the computers has turned numerical methods into a very powerful technique, and in many cases the only one, for the solution of the problems.

An analytical solution allows the determination of the solution at any point or the solution is continuously defined in the domain of interest. The finite difference solution allows the determination of approximate solution only at discrete, pre-selected points in the domain of interest. The solution at other points is obtained by interpolation.

Numerical solutions have the disadvantage that they need the knowledge of either a programming language or specialized software. The standard language programming courses are vanishing in undergraduate engineering programs. Few programs require courses in FORTRAN, Basic or C languages. Equation solvers are increasingly being used for obtaining numerical solutions, and the choice of what to use is primarily determined by the availability of the software and preference of the faculty. The use of equation solvers still requires learning of the software(s) used by the different faculty and some structured programming skills.

Spreadsheets provide an option that at least eliminates the need for learning a programming language, as most students have access to and are familiar with their use. They also provide a natural environment for numerical solutions, as each cell becomes a node. The built in plotting capabilities are also great help in presenting and understanding the results. One drawback of using spreadsheets for numerical solution is the user's lack of much control over spreadsheets' built in iterative schemes. This results in the slow convergence rates and limits their use to solving only simple problems.

Most differential or partial differential equations encountered in the mechanical engineering are second order boundary value problems, whose boundary conditions are specified at two or more points over the domain. The finite difference solution of these equations results in sets of algebraic equations whose coefficients matrix is tri-diagonal.

In this paper, a classic algorithm, called Thomas algorithm, used for the solution of set of algebraic equations whose coefficient matrix is tri-diagonal, is adopted to spreadsheets. It is used to obtain solutions to a number of problems in fluid mechanics and heat transfer non-iteratively, and in some cases where the governing equations are non-linear via some iterations.

Without needing much programming skills, or needing to learn specialized software, undergraduate students can use this approach and obtain the solution to many otherwise difficult problems easily and study the impact of different parameters.

Analysis

A tri-diagonal matrix is a matrix with nonzero elements only on the main diagonal and on the diagonals immediately above and immediately below the main one as shown below. Linear algebraic equations with tri-diagonal coefficient matrix frequently arise in the finite difference solutions.

$$\begin{bmatrix} B_1 & C_1 & & & & \\ A_1 & B_2 & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & & & & \\ & & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & & A_n & B_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \\ R_{n-1} \\ R_n \end{bmatrix} \quad (1)$$

Figure 1 shows a FORTRAN algorithm, known as the Thomas algorithm [**1**], provides an efficient method for solving such equations. It essentially transforms the tri-diagonal matrix into an upper triangular one which is then solved by back substitution.

```
C PROGRAM TRIDI SOLVES A SET OF EQUATIONS
C WITH TRIDIAGONAL COEFFICIENT MATRIX
      SUBROUTINE TRIDI(A,B,C,X,R,N)
      REAL A(100),B(100),C(100),R(100),X(100)
      A(N)=A(N)/B(N)
      R(N)=R(N)/B(N)
      DO 1 I=2,N
      II=-I+N+2
      BN=1/(B(II-1)-A(II)*C(II-1))
      A(II-1)=A(II-1)*BN
 1    R(II-1)=(R(II-1)-C(II-1)*R(II))*BN
      X(1)=R(1)
      DO 2 I=2,N
 2    X(I)=R(I)-A(I)*X(I-1)
      RETURN
      END
```

Figure 1. FORTRAN Code for Tri-diagonal Matrix.

Visual Basic Applications, VBA, has been integrated in Excel. Visual Basic is a high-level programming language that allows development of among other things User Defined Functions in Excel. Figure 2 is the VBA implementation of Thomas algorithm. The function TRIDI is an array function that returns N values for X_1 through X_N. Because TRIDI is an array function, you need to select N cells in a column before you enter the function. You also must press Control-Shift-Enter when entering the function. The syntax of TRIDI is:

=TRIDI(*starting cell of the column containing array A: Ending cell of the column containing array A,*
starting cell of the column containing array B: Ending cell of the column containing array B,
starting cell of the column containing array C: Ending cell of the column containing array C,
starting cell of the column containing array X: Ending cell of the column containing array X)

```
Option Base 1
Function TRIDI(ByVal Ac As Range, ByVal Bc As Range, ByVal Cc As Range, _
ByVal Rc As Range) As Variant
    Dim BN As Single
    Dim i As Integer
    Dim II As Integer
    Dim A() As Single, B() As Single, C() As Single, R() As Single, X() As Single

    N = Ac.Rows.Count
    ReDim A(N), B(N), C(N), R(N), X(N)
    For i = 1 To N
      A(i) = Ac.Parent.Cells(Ac.Row + i - 1, Ac.Column)
      B(i) = Bc.Parent.Cells(Bc.Row + i - 1, Bc.Column)
      C(i) = Cc.Parent.Cells(Cc.Row + i - 1, Cc.Column)
      R(i) = Rc.Parent.Cells(Rc.Row + i - 1, Rc.Column)
   Next i
     A(N) = A(N) / B(N)
     R(N) = R(N) / B(N)
   For i = 2 To N
      II = -i + N + 2
      BN = 1 / (B(II - 1) - A(II) * C(II - 1))
      A(II - 1) = A(II - 1) * BN
      R(II - 1) = (R(II - 1) - C(II - 1) * R(II)) * BN
   Next i
      X(1) = R(1)
   For i = 2 To N
      X(i) = R(i) - A(i) * X(i - 1)
   Next i
     TRIDI = Application.WorksheetFunction.Transpose(X)
End Function
```

Figure 2. VBA Code for Solving a Tri-diagonal Matrix Using Excel.

For example assume that there are 21 nodes, and the columns A, B, C, and D contain values of A, B, C, and R. To use the TRIDI function enter values of A, B, C and R in cells A1 through D21. Then select the cells in the column that will hold the values of X, in this case E1:E21 and type

=TRIDI(A1:A21, B1:B21, C1:C21, D1:D21)

in cell E1, then press Control-Shift-Enter (command Enter on Mac) to complete the array function. Excel places { } characters around the function to indicate it is an array function and computes the values of X and puts them in cells E1:E21.

As described in [2] to use this function in an Excel workbook:

1. Start a new workbook.
2. Start VBA (Press Alt+F11)
3. Insert a new module (Insert > Module)
4. Copy and Paste the Excel User Defined Function (UDF). For example all the text in Fig. 2.
5. Exit VBA (Press Alt+Q)
6. Use the functions (They will appear in the Paste Function dialog box, Shift+F3, under the "User Defined" category)

To use a UDF in more than one workbook, you need to save your function as a custom add-in:

1. Save your excel file that contains your VBA functions as an add-in file (.xla).
2. Load the add-in (Tools > Add-Ins...).

Note that unless you provide your UDF to others, they would not be able to use your spreadsheets that use to your UDFs [2]. A spreadsheet containing function TRIDI can be downloaded from [3] and can be turned into a USF using the above procedure.

Examples

The approach is demonstrated by several examples.

Transient heat conduction

The first example is transient conduction in a wall. The nondimensional equations are:

$$\frac{\partial^2 T^*}{\partial x^{*2}} = \frac{\partial T^*}{\partial t^*} \tag{2}$$

$$x^* = 0 \quad \frac{\partial T^*}{\partial x^*} = 0 \tag{3}$$

$$x^* = 1 \quad \frac{\partial T^*}{\partial x^*} = -BiT^* \tag{4}$$

$$t^* = 0 \quad T^* = 1 \tag{5}$$

Dropping the asterisks for simplicity, the finite difference of Eq. (2) is

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{(\Delta x)^2} = \frac{T_i - T_i^o}{(\Delta t)} \tag{6}$$

which simplifies to

$$T_{i-1} - \beta T_i + T_{i+1} = -\alpha T_i^o \tag{7}$$

$$T_1 - T_2 = 0. \tag{8}$$

$$-T_{N-1} + (1 + Bi\Delta x)T_N = 0 \tag{9}$$

$$\alpha = \frac{(\Delta x)^2}{(\Delta t)} \tag{10}$$

$$\beta = 2 + \alpha$$

Therefore,

$$\begin{aligned} A_i &= 1 \\ B_i &= -\beta \\ C_i &= 1 \\ R_i &= -\alpha T_i^o \end{aligned} \quad for \quad 1 < i < N \tag{11}$$

And

$$\begin{aligned} &B_1 = 1, C_1 = -1, R_1 = 0 \\ &A_N = -1, B_N = (1 + Bi\Delta x), R_1 = 0 \end{aligned} \tag{12}$$

Note that A_1 and C_N are not used and are typically set to be 1. In matrix form, the finite difference equations become

$$\begin{bmatrix} 1 & -1 & & & & \\ 1 & -\beta & 1 & & & \\ & 1 & -\beta & 1 & & \\ & & & & & \\ & & & 1 & -\beta & 1 \\ & & & & -1 & (1 + Bi\Delta x) \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \\ T_{n-1} \\ T_n \end{bmatrix} = \begin{bmatrix} 0 \\ -\alpha T_2^o \\ -\alpha T_3^o \\ \\ -\alpha T_3^o \\ 0 \end{bmatrix} \tag{13}$$

The *x* direction was discritized using 21 nodes and subroutine TRIDI was used to obtain the solution directly. The spreadsheet implementation is shown in Appendix A. The solution was also obtained using the iterative capabilities of Excel by solving for temperature of node *i* from Eq. (7)

$$T_i = \frac{T_{i+1} + T_{i-1} + \alpha T_i^o}{\beta} \tag{14}$$

and entering it in cells corresponding to nodes 2 to N-1.

The content of the cell corresponding to node 1 is set equal to the one for node 2, to satisfy Eq. (3). To satisfy Eq. (4), the content of cell corresponding to the temperature of node N is set to the content of the cell corresponding to node N-1, divided by $1+Bi\Delta x$, or

$$T_N = \frac{T_{N-1}}{1+Bi\Delta x} \tag{15}$$

It took around 700 iterations to obtain the results up to t*=1, using Δt of 0.1 and 21 nodes in the x direction, using a convergence tolerance of 10^{-5}. The maximum difference between the iterative and direct solutions was less than 0.2%

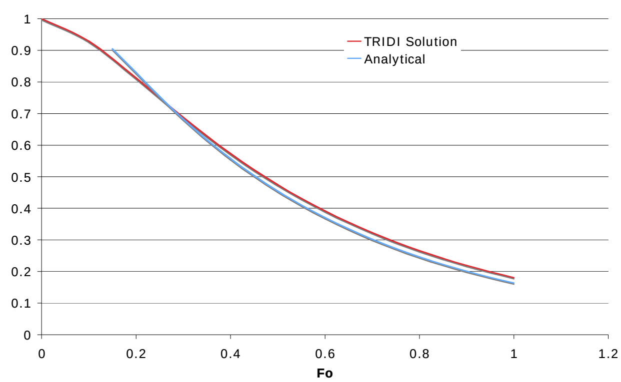

Figure 3. Comparison of Numerical and Analytical Solutions.

Figure 3 shows a comparison of the numerical and analytical solutions. The analytical solution is obtained by taking the first four terms of the infinite series, which as expected should provide accurate results for Fourier number larger than 0.2. As can be seen, there is close agreement between the numerical and analytical solutions.

Convective fins

This method is an excellent approach for analysis of fins (extended surfaces). The governing equation for fins is given by:

$$\frac{d}{dx}\left(kA_x\frac{dT}{dx}\right) - h\frac{dA_s}{dx}(T-T_\infty) = 0 \tag{16}$$

where $A(x)$ is the cross-sectional area of the fin and $A_s(x)$ is the surface area through which heat is transferred by convection. The boundary conditions are:

$T(0) = T_b$ and at x=L, $dT/dx = 0$ (insulated tip)

Equation (16) can be written as:

$$\frac{d^2T}{dx^2} + \frac{1}{A_x}\frac{dA_x}{dx}\frac{dT}{dx} - \frac{h}{k}\frac{1}{A_x}\frac{dA_s}{dx}(T-T_\infty) = 0 \tag{17}$$

The nondimensional form of equation (17) is:

$$\frac{d^2T^*}{dx^{*2}} + \frac{1}{A_x}\frac{dA_x}{dx}\frac{dT^*}{dx^*} - \frac{hL^2}{k}\frac{1}{A_x}\frac{dA_s}{dx^*}T^* = 0 \tag{18}$$

And the finite difference form of equation (18) is:

$$\frac{T^*_{i+1} - 2T^*_i + T^*_{i-1}}{(\Delta x^*)^2} + \frac{1}{A_x}\frac{dA_x}{dx^*}\frac{T^*_{i+1} - T^*_{i-1}}{2\Delta x} - \frac{hL^2}{k}\frac{1}{A_x}\frac{dA_s}{dx^*}T^*_i = 0 \tag{19}$$

Or

$$A_iT^*_{i-1} + B_iT^*_i + C_iT^*_{i+1} = R_i \tag{20}$$

where

$$A_i = \frac{1}{(\Delta x^*)^2} - \frac{1}{A_x}\frac{dA_x}{dx^*}\frac{1}{2\Delta x^*} \tag{21}$$

$$B_i = -\left(\frac{hL^2}{k}\frac{1}{A_x}\frac{dA_s}{dx^*} + \frac{2}{(\Delta x^*)^2}\right) \tag{22}$$

$$C_i = \frac{1}{(\Delta x^*)^2} + \frac{1}{A_x}\frac{dA_x}{dx^*}\frac{1}{2\Delta x^*} \tag{23}$$

and

$$R_i = 0 \tag{24}$$

The boundary conditions in finite difference form are:

$$T^*_1 = 1. \tag{25}$$

and

$$T^*_{N-1} = T^*_N \tag{26}$$

where N is the total number of nodes (the last node).

Hence, for the first and last nodes we have:

$A_1 = 1,\ B_1 = 1,\ C_1 = 0$, and $R_1 = 1$ (for the base) (27)
$A_N = 1,\ B_N = -1,\ C_N = 0$, and $R_N = 0$ (for the tip) (28)

This model is used to analyze straight, circular and triangular convective fins. For a straight fin, A_x is constant. Therefore

$$A_i = \frac{1}{(\Delta x^*)^2} \qquad (29)$$

$$B_i = -\left(M + \frac{2}{(\Delta x^*)^2}\right) \qquad (30)$$

$$C_i = \frac{1}{(\Delta x^*)^2} \qquad (31)$$

Where $M = \frac{hL^2}{k}\frac{p}{A}$

The dimensionless temperature distribution for this fin based on the present approach and that of exact solution are plotted in Figure 4. As shown in this figure the results of temperature distribution based on the present approach and that of exact solution are almost identical (relative error less than 0.5%).

For circular fins $A(r) = 2\pi r t$ and $A_s = 2\left(\pi r^2 - \pi r_i^2\right)$. Note that for circular fins *x* is replaced by *r*. The tri-diagonal matrix elements for circular fins become:

$$A_i = \frac{1}{(\Delta r^*)^2} - \frac{1}{2r^*\Delta r^*} \qquad (32)$$

$$B_i = -\left(M + \frac{2}{(\Delta r^*)^2}\right) \qquad (33)$$

$$A_i = \frac{1}{(\Delta r^*)^2} + \frac{1}{2r^*\Delta r^*} \qquad (34)$$

Where $M = \frac{hL^2}{k}\frac{2}{t}$

The dimensionless temperature distributions along the radial direction are shown in Figure 5.

The results are for M=1.33 and for an annular fin whose outer radius is twice its inner one. As shown in this figure the resulting temperature distribution based on the present approach and that of exact solution are very close (less than 1% relative error).

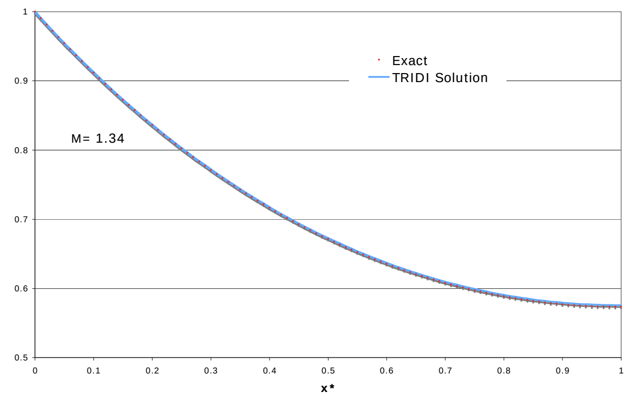

Figure 4. Dimensionless temperature distribution for a rectangular straight fin.

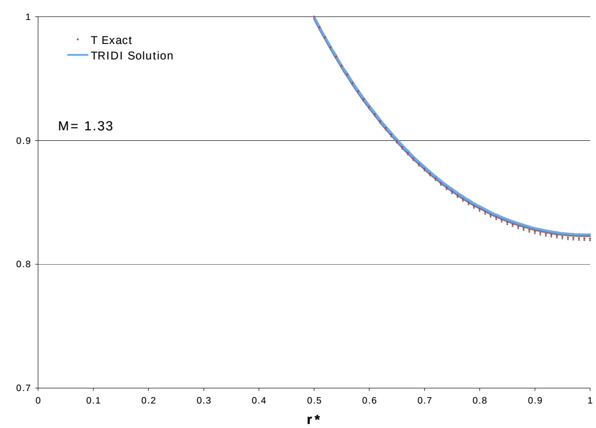

Figure 5. Dimensionless temperature distributions for a circular fin.

The next fin considered is a triangular fin shown in Fig. 6. The governing differential equation for the temperature distribution is given by [4]:

$$x^*\frac{d^2T^*}{dx^{*2}} + \frac{dT^*}{dx^*} - MT^* = 0 \qquad (35)$$

where $M = \frac{2hL^2}{kb}$ (36)

The finite difference form of the triangular fin equation is:

$$A_iT^*_{i-1} + B_iT^*_i + C_iT^*_{i+1} = R_i \qquad (37)$$

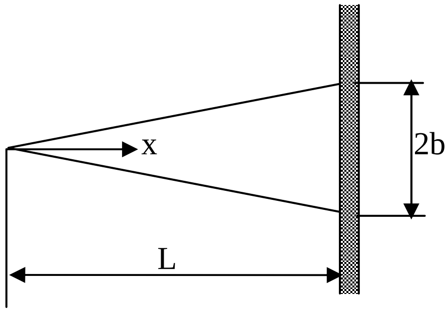

Figure 6. Schematics of a triangular fin.

where

$$A_i = \frac{x_i^*}{\Delta x^{*2}} - \frac{1}{2\Delta x^*} \tag{38}$$

$$B_i = -\left[\frac{2x_i^*}{\Delta x^{*2}} + M\right] \tag{39}$$

$$C_i = \frac{x_i^*}{\Delta x^{*2}} + \frac{1}{2\Delta x^*} \tag{40}$$

$$R_i = 0 \tag{41}$$

The exact solution for triangular fins as reported in the literature [4,5] is given by:

$$T^* = \frac{I_0(2\sqrt{Mx^*})}{I_0(2\sqrt{M})} \tag{42}$$

A finite, unspecified temperature is used for the tip boundary condition. The cross-sectional area of the triangular fin is zero at its tip; hence the heat flux at the tip is zero. Therefore, the actual boundary condition must be zero heat flux at the tip. The heat flux at the tip predicted by Equation (39) is finite. This is not a physically realistic solution and appears to have been used due to its ease in obtaining an analytical solution.

For the present numerical approach the physically realistic boundary condition of zero heat flux at the tip, i.e., $dT/dx = 0$, is used. The finite difference form of this boundary condition leads to $T_1^* = T_2^*$.

The resulting temperature distributions based on the present approach (with insulated tip) and exact solution (with finite temperature at the tip) are shown in Figure 7. As shown in Figure 7 there is close agreement between the results of both methods.

This approach can be used to model a wide range of extended surfaces with convection. Two recent publications, [6] and [7], show applications of numerical methods for analyzing various fins, which resulted in equations that can be easily, solved using the present approach.

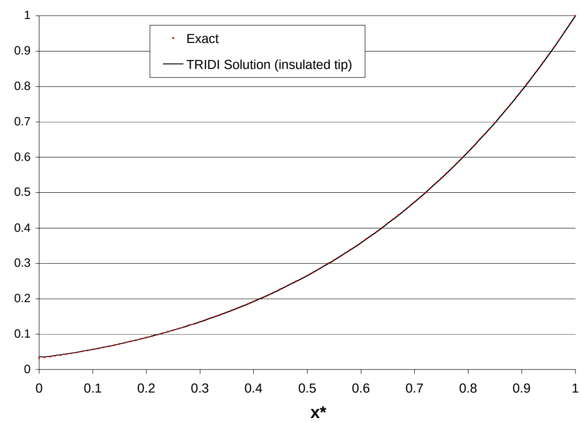

Figure 7. Dimensionless temperature distributions for a triangular fin.

Radiation fins

A more challenging problem is radiation fins, which result in nonlinear differential equations. An example of such fins is given in [8]. The resulting differential equation for a rectangular radiation fin is given by (see [8]):

$$\frac{d^2T}{dx^2} - \frac{\varepsilon\sigma}{kw}\left(T^4 - \frac{\alpha G}{\varepsilon\sigma}\right) = 0 \tag{43}$$

Where, ε and α are total surface emissivity and absorptivity, G is the external irradiation (typically solar irradiation), k is the fin conductivity and w is the thickness of the fin. The equilibrium temperature that a surface would achieve if it is insulated and is subject to irradiation G is given by:

$$T_s^4 = \frac{\alpha G}{\varepsilon\sigma} \tag{44}$$

Using the equilibrium temperature and dimensionless variables:

$$T^* = \frac{T}{T_b},\ T_s^* = \frac{T_s}{T_b},\ x^* = \frac{x}{L} \text{ and } \lambda = \frac{\varepsilon\sigma T_b^4 L^2}{kw} \qquad (45)$$

leads to the following differential equation

$$\frac{d^2T^*}{dx^{*2}} - \lambda(T^{*4} - T_s^{*4}) = 0 \qquad (46)$$

The boundary conditions to be satisfied are:

$$T^* = 0 \text{ at } x^* = 0 \qquad (47)$$

$$\frac{dT^*}{dx^*} = 0 \text{ at } x^* = 1 \qquad (48)$$

To solve this problem using the present approach the governing equation is linearized as

$$\frac{d^2T^*}{dx^{*2}} - h_r(T^* - T_s^*) = 0 \qquad (49)$$

where

$$h_r = \lambda(T^{*2} + T_s^{*2})(T^* + T_s^*) \qquad (50)$$

is the radiative heat transfer coefficient. This heat transfer coefficient is a function of fin temperature. The finite difference form of the above equation is given by:

$$A_i T_{i-1}^* + B_i T_i^* + C_i T_{i+1}^* = R_i \qquad (51)$$

Where

$$A_i = \frac{1}{(\Delta x^*)^2} \qquad (52)$$

$$B_i = -\left(\frac{2}{(\Delta x^*)^2} + h_{r_i}\right) \qquad (53)$$

$$C_i = \frac{1}{(\Delta x^*)^2}, \qquad (54)$$

$$R_i = -h_{r_i} T_s^* \qquad (55)$$

This case was solved by taking $\Delta x^* = 0.01$ and the initial value of h_{r_i} is estimated based on $T^* = 0.5$ at all locations and fixed values of λ and T_s^*. The problem can be easily solved using the TRIDI function to obtain the temperature distribution. Then h_{r_i} values are recalculated using the new temperature distributions. The equations are solved based on the revised h_{r_i}. This can be done by simply copy and paste of the cell for the initial temperature calculation. The cells can be copied and pasted a few times until the temperature gradient, i.e. $\frac{dT^*}{dx^*}$ at the base, between two consecutive revisions of h_{r_i} becomes negligibly small. It should be noted that for $\lambda \le 1.5$ it takes about four revisions of h_{r_i} to get temperature gradient between two consecutive calculations identical to five decimal places. For larger values of λ about ten revisions of temperatures are needed.

The efficiency of radiative fins is expressed by [8]:

$$\eta = \frac{q}{(\varepsilon\sigma T_0^4 - \alpha G)L} = \frac{q}{\varepsilon\sigma L T_0^4(1 - T_s^4)} \qquad (56)$$

Substituting for q and making it dimensionless, the above equation becomes:

$$\eta = -\frac{1}{\lambda(1 - T_s^4)}\left(\frac{dT}{dx}\right)_{\xi=0} \qquad (57)$$

Figure 8 shows a graph of radiative fin efficiencies versus λ with T_s as parameter. This graph is produced based on the present approach and it is a replica of what is given in [8]. The graph given in heat transfer textbook by Chapman [8] is based on a relatively complex numerical approach given in a NASA report [9].

Conclusion

In this paper, it is shown that the spreadsheets with a tri-diagonal matrix inversion module can be used to obtain solutions to a wide range of heat transfer problems. Use of this method does not require programming skills, or to learn specialized software. In most Thermal/Fluids courses the emphasis is on the physical concepts or numerical models, not programming skills.

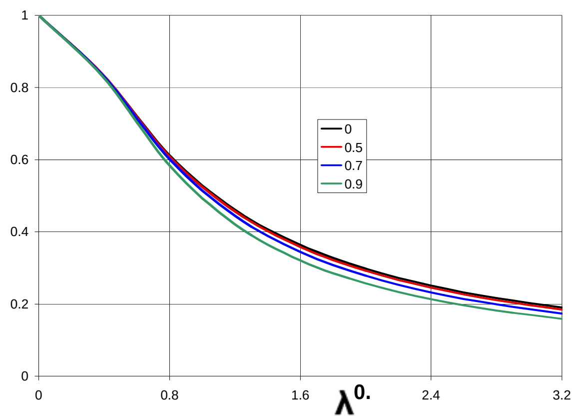

Figure 8. Efficiency of radiative fin of uniform thickness versus different values of λ with θ_s as parameter.

The students can use this approach to obtain the solution and study the impact of different parameters.

References

1. http://en.wikipedia.org/wiki/Thomas_algorithm
2. http://www.vertex42.com/ExcelArticles/user-defined-functions.html
3. http://energy.bradley.edu
4. Kakac, S. and Yener, Y., Heat Conduction, Hemisphere Publishing Corporation, 2nd Edition, 1988
5. Arpaci, V. S., Conduction Heat Transfer , Assison-Wessley (1966).
6. Campo, A , Kirkmam, R.D., and McNabb, G.," Avenues for computing the heat transfer from a circular fin of hyperbolic cross-sectional profile" International Journal of Mechanical Engineering Education, pp. 258-270, 2006.
7. Yovanovich, M.M., "Heat Balance Method for Spines, Longitudinal and Radial Fins with Contact Conductance and End Cooling," AIAA paper 2004-2569, presented at the 37th AIAA Thermophysics Conference, Portland OR, 2004.
8. Chapman, A.J., Fundamentals of Heat Transfer, Macmillan Publishing Company, 1987.
9. Lieblein, S., Analysis of Temperature Distribution and Radiant Heat Transfer Along Rectangular Fin of Uniform Thickness, NASA Tech. Note D-196, Washington, D.C., Nov. 1959.

Biographical Information

Dr. Ahmad Fakheri is a professor of Mechanical Engineering at Bradley University. He completed his undergraduate and graduate degrees all in mechanical engineering at the University of Illinois. His academic career has covered teaching and research in the area of thermal sciences, academic and professional leadership, and entrepreneurship. He has made breakthrough contributions in the field of heat exchangers, particularly on the application of the Second Law of Thermodynamics to heat exchangers. His work has led to the definition of the concept of thermal efficiency for heat exchangers and a new and simpler method for the design and analysis of heat exchangers. Dr. Fakheri is a Fellow of ASME and has served in a number of leadership positions and committees, including ASME's Process Industries and Heat Transfer Divisions, Manufacturing Group, ASME Board on Research and Technology Development, and ASME Education Center.

Dr. Mohammad Naraghi is a Professor of Mechanical Engineering at Manhattan College. Prior to joining Manhattan College, he was a Visiting Assistant Professor of Mechanical Engineering at University of Akron where he received his Ph.D. in Mechanical Engineering. Dr. Naraghi worked closely with NASA Glenn Research Center, through research grants and a number of Summer Faculty Fellowships, to develop a comprehensive Rocket Thermal Evaluation code (RTE). Because of this code, he received a certificate of recognition from NASA for the creative development of technically significant software which has been accepted and approved for dissemination to the public by NASA. Since the first release of RTE through NASA's COSMIC library (July 1991), Dr. Naraghi's research is in Thermal/Fluids area and he has published more than sixty articles in ASME, AIAA and international journals and conferences. He is recipient of a number of research grants from NASA and Air Force. Dr. Naraghi is a Fellow of ASME and a member of AIAA's Liquid Propulsion Technical Committee.

Appendix A

Table A1.

	A	B	C	D	E	F	G	H	I
1	N	21		TRIDI Solution					
2	dx	0.05		**I**	**A**	**B**	**C**	**R**	**R**
3	dt	0.10		1	0	1	-1	0.00000	0.00000
4	alfa	0.025		2	1	-2.025	1	-0.02322	-0.02031
5	Bi	10		3	1	-2.025	1	-0.02317	-0.02023
6	beta	-2.025		4	1	-2.025	1	-0.02308	-0.02009
22				20	1	-2.025	1	-0.025	-0.00845
23				21	-1	1.5	1	0	0
24									
25							**t**	**0**	**0.1**
26							x	**T**	**T**
27				1			0	1	0.9287
28				2			0.05	1	0.9287
29				3			0.1	1	0.9269
46				20			0.95	1	0.3380
47				21			1	1	0.2253
48									

Table A2 Formulas in the worksheet.

	A	B	C	D	E	F	G	H	I
1	N	=21		TRIDI Solution					
2	dx	=1/(N-1)		**I**	**A**	**B**	**C**	**R**	**R**
3	dt	0.1		1	0	1	-1	0	0
4	alfa	=dx^2/dt		2	1	=beta	1	=-alfa*H28	=-alfa*I28
5	Bi	=10		3	1	=beta	1	=-alfa*H29	=-alfa*I29
6	beta	=-2-alfa		4	1	=beta	1	=-alfa*H30	=-alfa*I30
22				20	1	=beta	1	=-alfa*H46	=-alfa*I46
23				21	-1	=1+Bi*dx	1	0	0
24									
25							**t**	**0**	=H25+dt
26							x	**T**	**T**
27				=D3			=(D27-1)*dx	1	=TRIDI($E3:$E23,$F3:$F23,$G3:$G23,H3:H23)
28				=D4			=(D28-1)*dx	1	=TRIDI($E3:$E23,$F3:$F23,$G3:$G23,H3:H23)
29				=D5			=(D29-1)*dx	1	=TRIDI($E3:$E23,$F3:$F23,$G3:$G23,H3:H23)
46				=D22			=(D46-1)*dx	1	=TRIDI($E3:$E23,$F3:$F23,$G3:$G23,H3:H23)
47				=D23			=(D47-1)*dx	1	=TRIDI($E3:$E23,$F3:$F23,$G3:$G23,H3:H23)
48									