NAMES: Joshua Emerling PERSON #: 50104912 **LAB** SECTION: H2
Drew Wentka 50110060 H2

EAS-230 Spring 2018

Programming Project # 1 (PP1) Due Dates:

Sections C, D, G and H: Monday 4/29

Sections A, B, E, F and I: Tuesday 5/1

Score: -------/100

Directions:

1. This project can be done in groups of maximum 2 students. You can choose your partner, however, he/she must be in your class section. Every group will submit only one set of files and one report that has both names of students worked on it. Each partner individually will submit a peer review show his/her contribution to every section of the project in percent. Each partner is fully responsible for every part of the project and accordingly he/she must know how to do each section individually. Students will be randomly selected to be present before the professor and will be tested on their understanding of each section of the project.
2. Each group must have a name that combines the 2 UBitNames of the 2 partners as UBitName1_UBitName2. You must submit your group details (Names, UBitNames, lab sections and person numbers) to your TAs in your lab section of the following week after posting this project on UBlearns).
3. You must save your 4 m-files (3 scripts & 1 function) in addition to the .dat file with the exact names as in the text of this assignment.
4. A paper copy (the report) including all scripts and functions, the display in the command window, all plots, the results and analysis must also be written (a template of the report will be posted on UBlearns). Your report must be turned-in at the start of your first lecture on the due dates shown above. Be sure to write your name, person # and lab section on every page.
5. A scan of your report must be saved as a pdf with the name UBitname_PP1.pdf.
6. All files of part 1, above, in addition to the pdf file must be zipped in one zip file with your GroupName.zip and uploaded to UBlearns before 11:59 PM on the due date (Please do not include in your zip file any files ending in ".m~", ".sav" or ".mat".)
7. You must write your own code and follow all instructions to get full credit. You are not allowed to use codes or scripts found on the internet or any other references.
8. You must use good programming practices, including indentation, commenting your functions scripts and choosing meaningful variable names to make your programs selfdocumenting.
9. It is your responsibility to make sure that your functions/scripts work properly and are free of errors by utilizing the resources at your disposal.

Background:

Newton's law of cooling [1] gives the rate of heat transfer from a surface at a temperature, T_s, to the surrounding medium at T_∞ where A_s is the heat transfer surface area (depending on the geometry of the surface) and h is the convection heat transfer coefficient (depending on the material).

$$Q_{conv} = hA(T_s - T_\infty) \quad (1)$$

The two main methods to increase the rate of heat transfer are (1) to increase the convection heat transfer coefficient, h, or (2) to increase the surface area by adding extended surfaces called fins. Figure 5-20[1] shows the geometry of a triangular fin used to increase the rate of heat transfer. As shown in Figure, the fin length is divided into equally spaced nodes for the numerical analysis. [More details can be found in Example 5-2 of reference [1] which, as well, is provided in the file supplementary materials on UBlearns].

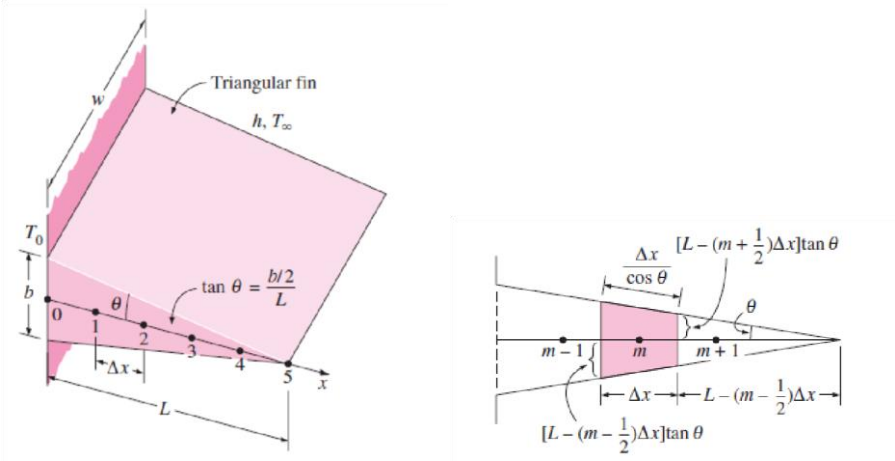


Figure 5-20[1]: Schematic for Example 5-2[1] and the volume element of a general interior node of the fin.

In practice, the temperature of the fin decreases along the length of the fin. Often, numerical methods are used to determine these changes in temperature along the length of the fin. One such method is called the finite difference method. In summary, we can divide the fin into a specific number of control volumes, identified at specific locations called nodes, and apply a set of equations to each node. This set of equations is known as the energy balance approach, generally defined in (2).

$$\begin{matrix}\textit{Rate of heat} \\ (\textit{conduction at the}\\ \textit{the node right side of the node the node}\end{matrix} + \begin{matrix}\textit{Rate of heat} \\ (\textit{conduction at the}\end{matrix} + \begin{matrix}\textit{Rate of heat} \\ (\textit{generation inside}) = \textit{left side of}\end{matrix}$$

$$\textit{Rate of change} \\ (\textit{of the energy content}) \quad (2)$$

of the node

Applying this equation at each node provides a system of equations, one equation per node, which is used to determine the temperature at each node showing the change in temperature along the length of the fin. In practice, the more nodes, the better the approximation to reality.

In Figure 5-20, the fin is divided into 6 nodes [0:5] representing 6 control volumes. Nodes 1, 2, 3, and 4 are called internal nodes, since they represent internal control volumes, while nodes 0 and 5 are boundary nodes representing control volumes at the edges of the fin. Note that the thickness of each internal node is identified by Δx while the thickness of each boundary nodes is $\frac{\Delta x}{2}$.

In order to be consistent with a MATLAB approach we will start at node number 1 resulting in our six nodes as [1:6] instead of [0:5]. Applying the energy balance approach for internal nodes, we end up with an equation for the sum of heat transfer from all side of the node (3).

$$\sum_{\text{all sides}} \dot{Q} = 0 \quad \rightarrow \quad kA_{\text{left}} \frac{T_{m-1} - T_m}{\Delta x} + kA_{\text{right}} \frac{T_{m+1} - T_m}{\Delta x} + hA_{\text{conv}}(T_\infty - T_m) = 0 \tag{3}$$

This equation can be used for nodes 2, 3, 4, and 5 and accordingly four equations can be written with six unknowns, T_1 through T_6, where T_1 is the temperature at boundary node 1, T_2 through T_5 ate the temperatures at internal nodes, 2:5, and T_6 is the temperature at boundary node 6. m in the previous equation corresponds to the node number.

Using the specific geometry in Figure 5-20, equation (3) can be rewritten as equation (4) by substituting terms specific to the geometry and rearranging so that each temperature has a corresponding coefficient equation, where m = 2, 3, 4, and 5 or, in general, m = 2 to M-1, where M is the total number of nodes.

$$\left[1 - \left(m - \frac{1}{2}\right)\frac{\Delta x}{L}\right] T_{m-1} - \left[2 - 2m\frac{\Delta x}{L} + \frac{h(\Delta x)^2}{kL\sin\theta}\right] T_m + \left[1 - \left(m + \frac{1}{2}\right)\frac{\Delta x}{L}\right] T_{m+1} = -\frac{h(\Delta x)^2}{kL\sin\theta} T_\infty \tag{4}$$

For the left boundary node, node 1, the temperature is always given as T_0 which is the base temperature of the fin and usually equal to the temperature of the surface where the fins are attached.

$$T_1 = T_0 \tag{5}$$

For the right boundary node, node 6, another second energy balance is performed with the resulting equation shown in (6)

$$kA_{\text{left}}\frac{T_4 - T_5}{\Delta x} + hA_{\text{conv}}(T_\infty - T_5) = 0 \qquad (6)$$

where

$$A_{\text{left}} = 2w\frac{\Delta x}{2}\tan\theta \qquad \text{and} \qquad A_{\text{conv}} = 2w\frac{\Delta x/2}{\cos\theta}$$

Upon substitution of terms and rearranging, equation (6) can be rewritten in general form for M nodes as:

$$T_{M-1} - \left[1 + \frac{h\,\Delta x}{k\sin\theta}\right]T_M = -\frac{h\,\Delta x}{k\sin\theta}T_\infty \qquad (7)$$

Together, equations 4, 5 and 7 provide *M* linear equations with *M* unknown temperatures which can be solved simultaneously to determine the temperature field T(1:M).

The total heat transfer from the fin can be determined from

$$\dot{Q}_{\text{fin}} = \sum_{m=0}^{5}\dot{Q}_{\text{element},\,m} = \sum_{m=0}^{5} hA_{\text{conv},\,m}(T_m - T_\infty)$$

Noting that the heat transfer surface area is $w\Delta x/\cos\theta$ for the boundary nodes 0 and 5, and twice as large for the interior nodes 1, 2, 3, and 4, we have

$$\dot{Q}_{\text{fin}} = h\frac{w\Delta x}{\cos\theta}[(T_0 - T_\infty) + 2(T_1 - T_\infty) + 2(T_2 - T_\infty) + 2(T_3 - T_\infty) + 2(T_4 - T_\infty) + (T_5 - T_\infty)]$$

This equation can be rewritten for the *M* nodes as:

$$\dot{Q}_{fin} = h\frac{w\Delta x}{\cos\theta}\left[(T_1 - T_\infty) + (T_M - T_\infty) + 2\sum_{m=2}^{M-1}(T_m - T_\infty)\right] \qquad (8)$$

In addition, the fin efficiency can be determined from (9) where the total heat transfer, Q_{fin}, is determined by equation (8) and the maximum heat transfer, Q_{max}, can be determined by equation (10) which assumes that the whole fin surface is at T_0.

$$\eta_{\text{fin}} = \frac{\dot{Q}_{\text{fin}}}{\dot{Q}_{\text{max}}} \qquad (9)$$

$$\dot{Q}_{max} = hA_{fin,\ total}(T_0 - T_\infty) = h(2wL/\cos\theta)(T_0 - T_\infty) \quad (10)$$

For more details about the numerical methods, described previously, please refer to the pages of reference [1] in the provided supplementary material on UBlearns.

The analytical solution of the fin heat transfer problem of triangular profile [2] can be written as

$$\frac{T - T_\infty}{T_0 - T_\infty} = \frac{I_0\left(N\sqrt{L(L-x)}\right)}{I_0\left(N\sqrt{L^2}\right)} \quad (11)$$

where, L is the length of the fin, x is the distance from the base of the fin, and $N = 2\sqrt{h/k\delta}$ where δ is one half of the thickness at the base ($\delta = b/2$), and I_0 is the modified Bessel function of the first kind of order zero.

The rate of heat transfer based on the analytical solution of the temperature profile is defined as:

$$\dot{Q}_{fin} = 2w\sqrt{hk\delta}(T_0 - T_\infty)\frac{I_1\left(2\sqrt{hL^2/k\delta}\right)}{I_0\left(2\sqrt{hL^2/k\delta}\right)} \quad (12)$$

Where w is the width of the fin and I_1 is the modified Bessel function of the first kind of order one.

The fin efficiency can then be determined based on the exact solution as:

$$\eta_{fin} = \frac{\dot{Q}_{fin}}{\dot{Q}_{max}} = \frac{\dot{Q}_{fin}}{2h\sqrt{L^2 + \delta^2}[w(T_0 - T_\infty)]} \quad (13)$$

PP1P1(40 pts):

Assume that we are working with an aluminum alloy (k = 180 W/m^{o}C) triangular fin with a length, L = 5 cm, base thickness, b = 1 cm, a very large width, w = 1 m. The base of the fin is maintained at a temperature of T_0 = 200^{o}C (at the left boundary node). The fin is losing heat to the surrounding air/medium at T_∞ = 25^{o}C with a heat transfer coefficient of h = 15 W/m^{2o}C. Using the finite difference numerical method described previously, perform the following steps.

1- Rewrite the system of equations described by equations (4), (5), and (7) in the format A**T** = **b** where **T** is the temperature vector T(1:M). Start with 6 nodes which is the same number of nodes in Example 5-2 for comparison.

2- Solve this system using MATLAB and determine the temperature field T(1:6) and compare your results with that in example 5-2 of reference [1]. To check your answers, your resulting temperatures should match those seen in Example 5-2 in reference [1] provided in the supplementary background material on UBlearns.

3- Determine the fin rate of heat transfer Q_{fin} and the fin efficiency $_{fin}$ assuming a width, w = 1 m.

4- In order to determine a more precise temperature field you need to increase the number of nodes to M nodes, where M can be any integer number > 1. Rewrite the general system of equations for M nodes into the format A**T** = **b**, noting that equations (5) and (7) for boundary nodes will stay the same and equation (4) is M-2 equations.

5- Write a script file PP1P1.m that does the following

   a. Define all variables given in the statement above such as L, w, k, etc. with the same values given in the example.

   b. Preallocate matrix *A* and vector **b** using the **`zeros`** function.

   c. Generate the entries of matrix *A* and vector **b** for rows [2:M-1] using for loops.

   d. Entries for row 1 and row M can be entered manually.

   e. Solve the system A**T** = **b** for M = 11, 21, and 101.

   f. For M=101, plot the temperature **T** vs **x** where x is measured from the fin base, **x**(1) = 0. **x** can be determined using node location and should have corresponding units associated with it. Compare with the temperature calculated from analytical solution, equation (11), for the same **x**, and plot the analytical **T** on the same plot of the numerical **T**. Use different line styles, line width and Fully annotate your plot (titles, axis labels, legends, etc). Note: Use function **`besseli()`** in MATLAB to determine the modified Bessel function I_0 and I_1 (Hint: **`besseli(nu,Z)`** computes the modified Bessel function of the first kind, $I_v(z)$, for each element of the array **`Z`**. The order **`nu`** need not be an integer, but must be real. The argument **`Z`** can be complex. The result is real where **`Z`** is positive.)

   g. Find the rate of heat transfer Q_{fin} and the fin efficiency $_{fin}$ and compare with that of the analytical solution.

   h. Run your script with 101 nodes one more time for copper with constant thermal conductivity $k = 380$ W/m²K and another time for steel with constant thermal conductivity $k = 60$ W/m²K. Compare between the temperature distribution in the three metals (aluminum alloy, copper, and steel) using one fully annotated plot (title, legend, grid, labels, etc.) with different line style and colors for each alloy. Which metal of the three gives the

best thermal performance in terms of the rate of heat transfer and the fin efficiency, explain?

1-4)

A =

```
   1.0000         0         0         0         0         0
   0.7000   -1.2017    0.5000         0         0         0
        0    0.5000   -0.8017    0.3000         0         0
        0         0    0.3000   -0.4017    0.1000         0
        0         0         0    0.1000   -0.0017   -0.1000
        0         0         0         0    1.0000   -1.0084
```

b =

```
  200.0000
   -0.0419
   -0.0419
   -0.0419
   -0.0419
   -0.2094
```

T_values =

```
  200.0000
  198.5602
  197.1259
  195.6964
  194.2670
  192.8612
```

Qfin =

```
  258.406437
```

nfin =

0.979520

5)

e)

>> PP1P1

node_limit =

11

T_values =

200.0000
199.2813
198.5641
197.8483
197.1340
196.4211
195.7097
194.9996
194.2907
193.5819
192.8789

>> PP1P1

node_limit =

21

T_values =

200.0000
199.6409
199.2821
198.9237
198.5657
198.2081
197.8508
197.4939
197.1374
196.7812
196.4254
196.0700
195.7149
195.3602
195.0059
194.6519
194.2983
193.9450
193.5921

193.2391

192.8876

>> PP1P1

node_limit =

101

T_values =

200.0000

199.9282

199.8564

199.7847

199.7129

199.6412

199.5695

199.4977

199.4261

199.3544

199.2827

199.2111

199.1394

199.0678

198.9962

198.9246

198.8531

198.7815

198.7100

198.6384

198.5669

198.4954

198.4240

198.3525

198.2810

198.2096

198.1382

198.0668

197.9954

197.9240

197.8526

197.7813

197.7100

197.6386

197.5673

197.4960

197.4248

197.3535

197.2823

197.2110

197.1398

197.0686

196.9974

196.9263

196.8551

196.7840

196.7129

196.6417

196.5706

196.4996

196.4285

196.3575

196.2864

196.2154

196.1444

196.0734

196.0024

195.9315

195.8605

195.7896

195.7187

195.6478

195.5769

195.5060

195.4351

195.3643

195.2935

195.2226

195.1518

195.0811

195.0103

194.9395

194.8688

194.7981

194.7274

194.6567

194.5860

194.5153

194.4447

194.3740

194.3034

194.2328

194.1622

194.0916

194.0211

193.9505

193.8800

193.8095

193.7390

193.6685

193.5980

193.5275

193.4571

193.3867

193.3162

193.2458

193.1754

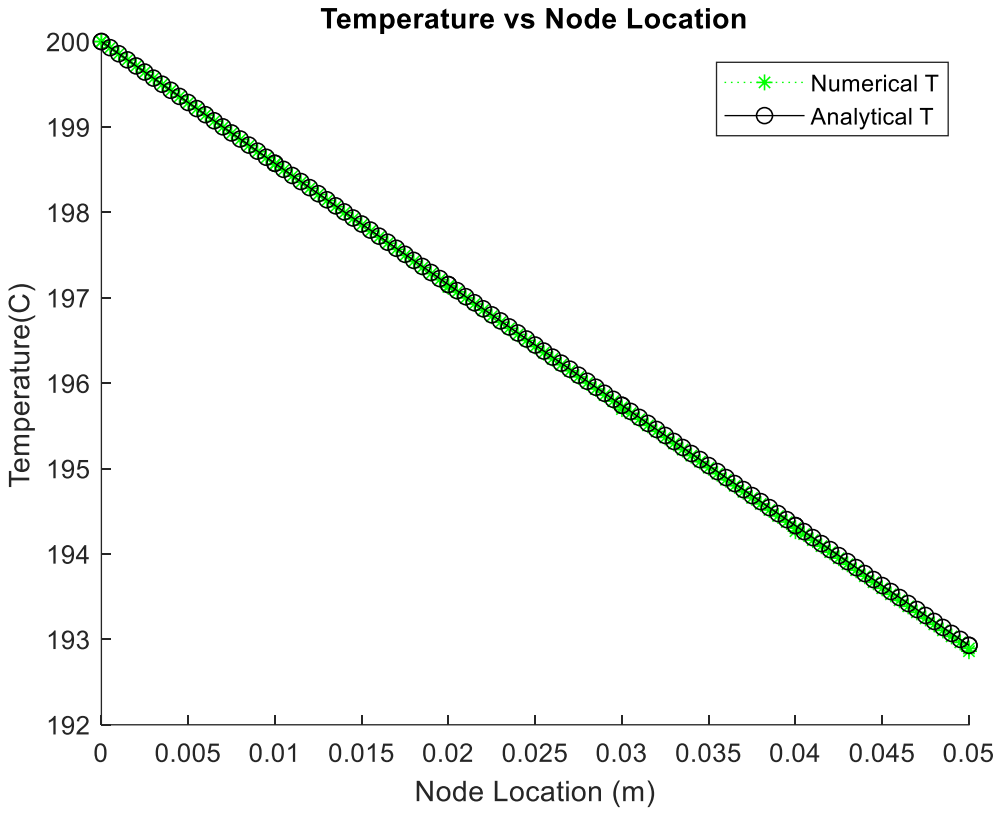
193.1051

193.0347

192.9643

192.8940

f)



g)

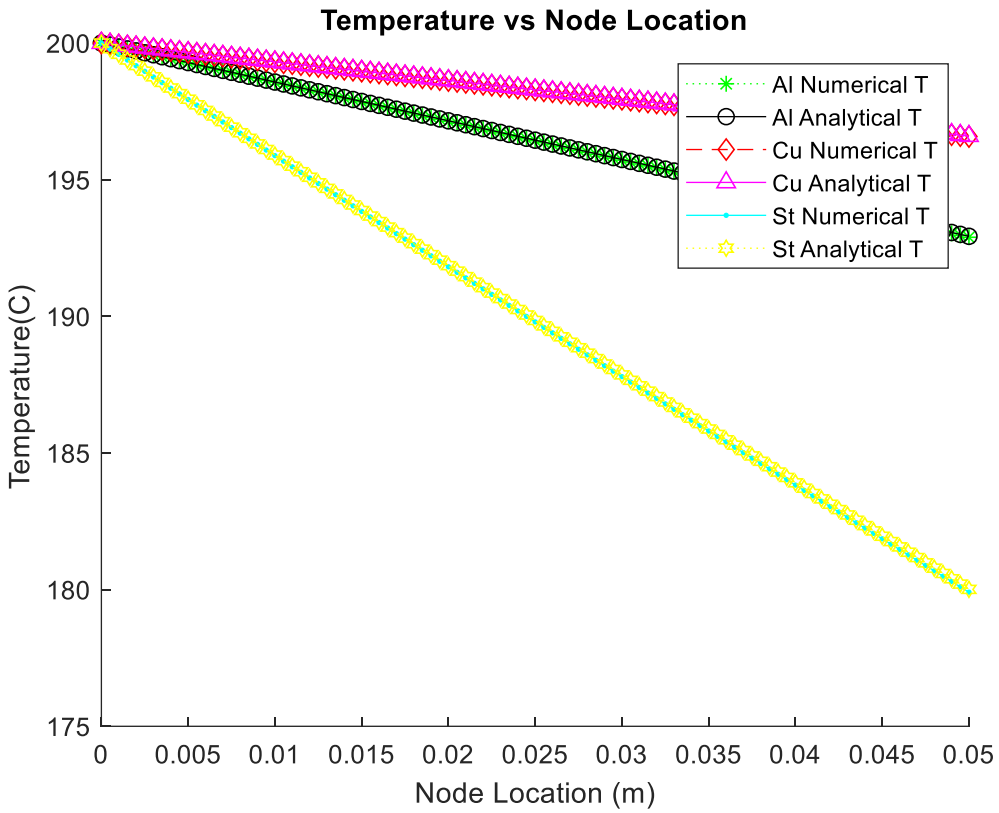
>> PP1P1

Numerical : Rate of Heat Transfer 258.434578, Fin Efficiency 0.979627

Analytical : Rate of Heat Transfer 257.178930, Fin Efficiency 0.974867

Our numerical and analytical solutions were almost the same with Rate of heat transfer differing by less than 1 and Fin Efficiency differing by less than 0.05.

h)



>> hold on

>> %Aluminum Alloy

>> PP1P1

Numerical : Rate of Heat Transfer 258.434578, Fin Efficiency 0.979627

Analytical : Rate of Heat Transfer 257.178930, Fin Efficiency 0.974867

>> %Copper

>> PP1P1

Numerical : Rate of Heat Transfer 261.226747, Fin Efficiency 0.990211

Analytical : Rate of Heat Transfer 259.943168, Fin Efficiency 0.985345

>> %Steel

>> PP1P1

Numerical : Rate of Heat Transfer 248.507640, Fin Efficiency 0.941997

Analytical : Rate of Heat Transfer 247.352785, Fin Efficiency 0.937620

>> legend('Al Numerical T','Al Analytical T','Cu Numerical T','Cu Analytical T','St Numerical T','St Analytical T')

>> hold off

Out of the 3 metals Copper has the Highest Rate of Heat Transfer and the best Fin Efficiency, followed by Aluminum Alloy. Steel has the lowest Rate of Heat Transfer and the worst Fin Efficiency.

PP1P2 (30 pts):

The thermal conductivity, k, of the fin material is usually a function of the temperature as shown in Table 1 for different alloys of Aluminum, Copper, and Steel respectively.

Table 1 Temperature dependent thermal conductivity of some metallic alloys [3-4]

Metal	Alloy Name	Thermal Conductivity (k) [W/m.K]	Temperature Range [K]
Aluminum	Al2	$k_s = 149.7 + 0.0809T - 1 \times 10^{-4}T^2$	$298 \leq T \leq 840$
	Al2	$k_s = 76.64 + 0.2633T - 2 \times 10^{-4}T^2$	$298 \leq T \leq 773$
	Al3	$k_s = 124.7 + 0.56T + 1 \times 10^{-5}T^2$	$298 \leq T \leq 890$
Copper	Cu1	$k = 453.9 - 0.1054\,T$	$100 \leq T \leq 1200$
	Cu2	$k_s = 140.62 + 112.14 \times 10^{-4}T$	$460 \leq T \leq 1188$
	Cu3	$k_s = 16.041 + 438.9 \times 10^{-4}T$	$T \leq 1443$
Steel	St1	$k = 76.63 - 0.0459\,T$	$400 \leq T \leq 1000$
	St2	$k_s = 6.31 + 27.2 \times 10^{-3}T - 7.0 \times 10^{-6}T^2$	$298 \leq T \leq 1573$
	St3	$k_s = 20 + 61.5 \times 10^{-4}T$	$T \leq 1727$

Write a function **`ThCond.m`** that takes the temperature in K as an array (scalar, vector or matrix) and the alloy name as a string and returns the thermal conductivity, k, as an array of the same size as T. The function should check the alloy name and the corresponding temperature range and provide an error message using the **`error`** function if either the alloy name is not found or the temperature entered is invalid. Use an if-else structure and/or switch case structure to choose the right equation, to check about the alloy name and to check about the temperature range.

I. Test you function in the command window for the following cases:
   1. T = 400 K and Pl1
   2. T = 300 K and St1

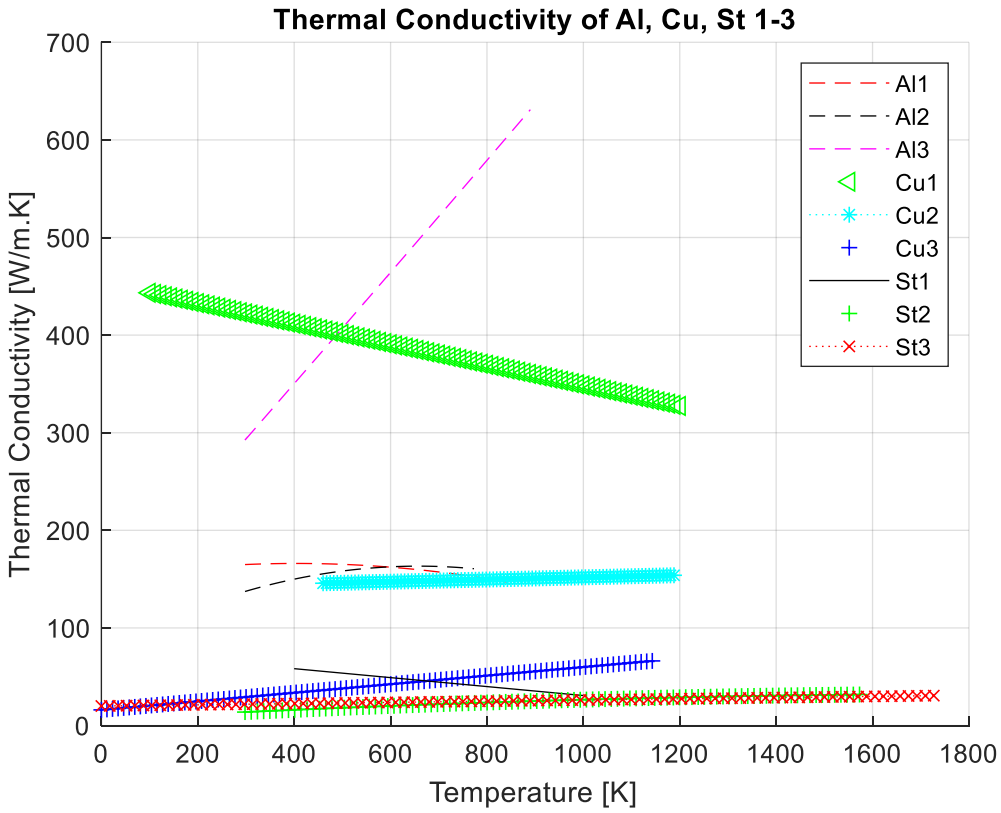
II. Write a script file PP1P2.m to plot the conductivity as a function of temperature for every alloy of table 1 in its temperature range. For every alloy, create a variable T_Alloyname that covers the temperature range of that alloy with 100 points using the **`linspace()`** function. The _Alloyname may be _Al1, Al_2, Cu3, etc. Use the function ThCond with the variable T_Alloyname to determine the thermal conductivity of that alloy as a function of temperature and assign it to a variable k_Alloyname. By the end of this file you must have 9 pairs of data: T_Al1 and k_Al1, T_Al2 and k_Al2, …, T_St3 and k_St3. Plot the 9 line graphs (one line graph for each T and its corresponding k) in one plot using different line styles, colors, etc. Fully annotate you figure and add legend and grid. Use logarithmic scale of the y axis which is the axis of the thermal conductivity k.

I

```
>> ThCond(400,'Pl1')
Error using ThCond (line 173)
Not a valid Alloy

>> ThCond(300,'St1')
Error using ThCond (line 128)
Temperature out of Range for Alloy
```

II


Thermal Conductivity of Al, Cu, St 1-3
Thermal Conductivity [W/m.K]
Temperature [K]
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
1400
1600
1800
Al1
Al2
Al3
Cu1
Cu2
Cu3
St1
St2
St3

PP1P3 (30 pts):

Substituting the thermal conductivity, k, as a function of the temperature makes the system of equations a non-linear system. Still this system can be solved analytically and numerically. The numerical solution in this case will be an iterative solution where you first assume that the fin is initially at the base temperature, T_0, everywhere. This temperature field can be called T_{old} and accordingly the thermal conductivity, k, can be determined using the **`ThCond`** function at T_{old} and the alloy name. You must note that the temperature value entered to the **`ThCond`** function must be in K while the temperature used in your script is in ^oC. To avoid any errors, when you use the function, type **`ThCond(Told+273,'---').`** The k value determined at T_{old} is then input into the system of equations to determine a new temperature field (vector), T_{new}. The error can be calculated as $\|T_{new} - T_{old}\|$. At the end of every pass T_{new} is assigned to T_{old} and this process is then repeated as long as the error is greater than a tolerance of $1E-8$. In order to avoid an infinite loop, a counter can be implemented to determine the number of iteration. The loop should continue until the convergence criterion is reached (error is less than a tolerance) or the number of iterations reaches 100 whichever is first. (See Figure 1)

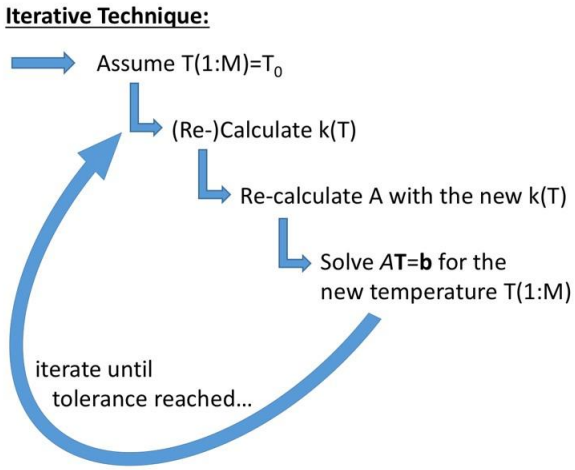


Figure 1: Logic flow for iterative technique for solving the nonlinear system.

Write a script file PP1P3.m that does the following

a. Define all variables as previously described, such as L, w, etc. Do not assign values for the thermal conductivity, k, instead it will be determined as a function of temperature from the function **`ThCond()`**.
b. Repeat parts b, c, and d of PP1P1-5.

c. Solve the system $A\mathbf{T} =$ for M = 101 and variable thermal conductivity, k, using the iterative technique and the function **`ThCond()`**. Run your script PP1P3.m for alloys Al1, Cu1, and St1, separately, i.e., a run for each alloy. You can do this by prompting the user to enter the alloy name at the beginning of your script and assign this to a variable name **`Alloy_name`**.

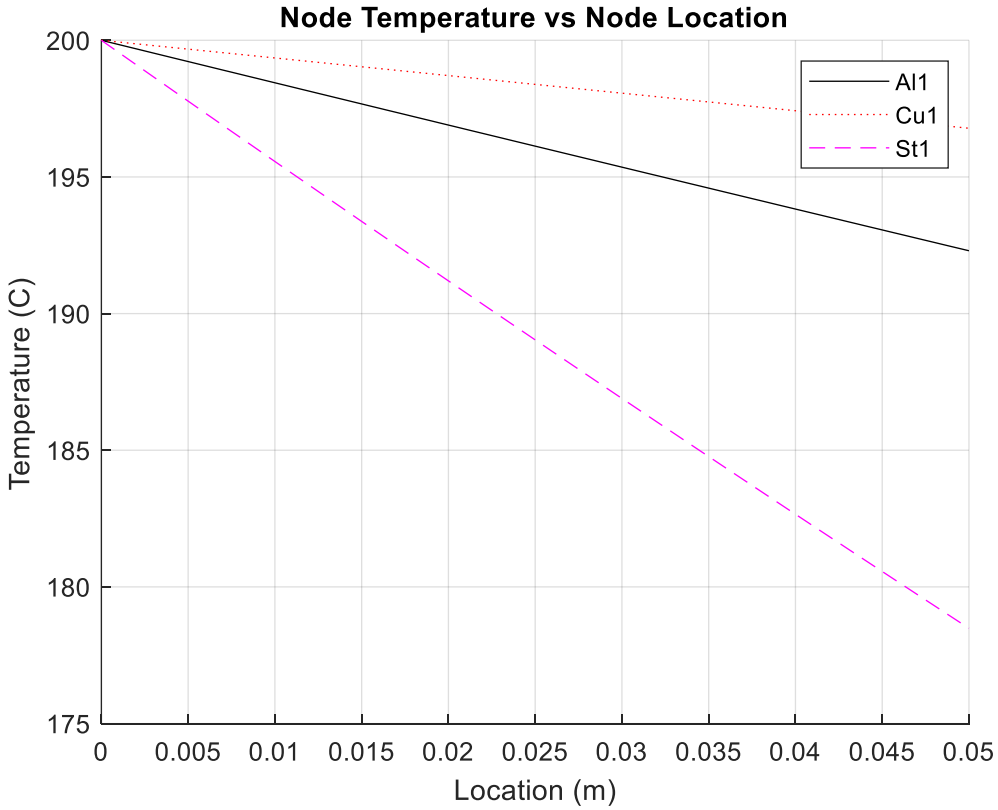
d. Save the distance vector, **x**, and the corresponding temperature vector, **T**, from each run in the same "PP1P3.dat" ascii file. Transpose the vectors before saving and you can use **`-append`** in the **`save`** command to add variables to a pre-existing file.

e. In the command window, load your data file and plot the temperature **T** vs **x** for the three alloys in one plot. Fully annotate your plot.

f. Report the rate of heat transfer Q_{fin} and the fin efficiency $_{fin}$ for the three metals.

e)

```
>> data = load('PP1P3.dat','-ascii');
>> Al_x = data(1:101);
>> Al_T = data(102:202);
>> Cu_x = data(203:303);
>> Cu_T = data(304:404);
>> St_x = data(405:505);
>> St_T = data(506:606);
>> hold on
>> plot(Al_x,Al_T,'k');
>> plot(Cu_x,Cu_T,'r:');
>> plot(St_x,St_T,'m--');
>> title('Node Temperature vs Node Location');
>> ylabel('Temperature (C)');
>> xlabel('Location (m)');
>> legend('Al1','Cu1','St1');
>> grid on
>> hold off
```



f)

>> PP1P3

Enter an Alloy Name 'Al1'

Rate of Heat Transfer 257.982891, Fin Efficiency 0.977915

>> PP1P3

Enter an Alloy Name 'Cu1'

Rate of Heat Transfer 261.379897, Fin Efficiency 0.990791

>> PP1P3

Enter an Alloy Name 'St1'

Rate of Heat Transfer 247.387577, Fin Efficiency 0.937752

References:

[1] Y. Cengel, "Heat Transfer – A Practical Approach," 2nd edition, McGraw-Hill.

[2] A. Karimi, "Use of spreadsheets solving heat conduction problems in fins. American Society for Engineering Education," (2008). AC 2008-1867.

[3] J.J. Valencia and K.O. Yu, "Thermophysical properties," In Modeling for casting and solidification processing, (Ed. K.-O. Yu) 2002 (Marcel Dekker, Inc., New York).

[4] F.P. Incropera and D.P. DeWitt (2002). "Fundamentals of heat and mass transfer." New York: J. Wiley.