NAME: **LAB** SECTION:

EAS 230 – Spring 2020
Programming Project
Due Friday, April 17th, 11:59PM on UBlearns
SCORE: _____/150

Directions:

- This project is recommended to be done in groups of **two** students at maximum. Your partner can be from section G, H, or I. These are sections given by Dr. Hammond.
  - All students must fill out this survey on UBlearns. This is where you will
    - request a partner if you do not have one,
    - name a partner you want to work with, or
    - identify that you want to work alone.
  - If you request a partner, you will be randomly assigned another student. If you do not fill out the survey, it is assumed you will be working as an individual.
- Your final project will be submitted as **single zip file (*.zip)** containing eight *.m files (five functions and three script files) and your final report as a *.pdf.
  - Script files should be named in the following format: PP_Partx_ubit1ubit2.m where x indicates the part letter and ubit1 and ubit2 indicate the ubitnames of each partner.
  - Function files should be named according to the name of the function specified in the HW document.
  - A template for your final report is posted on UBlearns. You must read the template first to determine what your final report should contain.
- 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.
- You must use good programming practices, including indentation, commenting your functions scripts and choosing meaningful variable names to make your programs self-documenting.
- 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.
- You must upload your zip file and only your zip file to UBlearns by the end of the day (11:59PM) on Friday, April 18th. No late submissions will be accepted. Only one submission per group is required.

Numerical Analysis of Annular Radial Fins of Uniform Thickness[1,2]

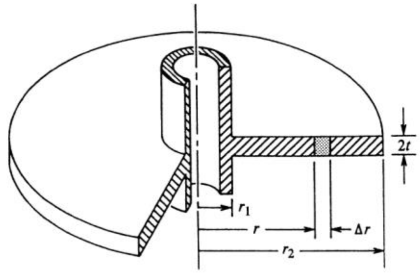

Figure 1: *Annular radial fin of uniform thickness.*

Background:

Figure 1 shows an annular radial fin used to increase the surface area of a circular tube for the purpose of increasing the rate of heat transfer to/from the surface. The rate of heat transfer, $\dot{Q}_{conv}$, from a surface at temperature T_s to the surrounding environment at a temperature T_∞ is determined from Newton's law of cooling[3] seen in equation 1 where A_s is the surface area and h is the convective heat transfer coefficient.

$$\dot{Q}_{conv} = hA_s(T_s - T_\infty) \qquad \text{(equation 1)}$$

The surface temperature along the fin length, T_s in the previous equation, is not constant, but varies along the fin from its base to its tip. Engineers are usually interested in determining the temperature distribution (how the temperature changes) along the fin and the rate of heat transfer through the fin, $\dot{Q}_{fin}$. To achieve this goal, engineers can use numerical methods and/or analytical methods.

Numerical methods:

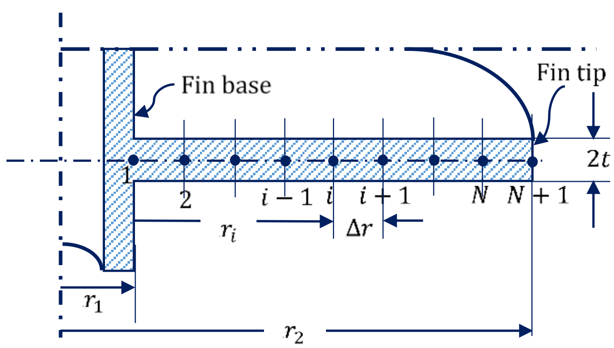

Figure 2: *Schematic for the nodes used for the numerical analysis.*

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

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

[3] A. Compo and S. Chikh, "Reproduction of the Fin Efficiency Diagram for Annular Fins of Uniform Thickness by Solving Systems of up to Four Algebraic Equations," International Journal of Mechanical Engineering Education, Volume: 34-1, pp 85-92, 2006.

One numerical method is known as the finite difference method where the fin length is divided into a finite number of equally spaced nodes, or points along the fin length, as seen in Figure 2. The fin is divided into sections of equal lengths (ΔR) with $N+1$ nodes. Nodes 2, 3, …, N are called internal nodes, while nodes 1 and $N+1$ are boundary nodes at the boundaries of the fin. The temperature at each node or point is determined by an equation derived from the energy balance at each node as

$$\begin{pmatrix} \textit{Rate of heat} \\ \textit{conduction at the} \\ \textit{left side of the node} \end{pmatrix} + \begin{pmatrix} \textit{Rate of heat} \\ \textit{conduction at the} \\ \textit{right side of the node} \end{pmatrix} + \begin{pmatrix} \textit{Rate of heat} \\ \textit{generation at} \\ \textit{the node} \end{pmatrix} = \begin{pmatrix} \textit{Rate of change} \\ \textit{of the energy content} \\ \textit{of the node} \end{pmatrix}$$

Since each node has its own equation, this produces a system of $N+1$ linear equations that can be solved using the linear algebra techniques taught in class. In practice, the more nodes, the more equations, the better approximation to the exact/analytical solution.

The equation, in dimensionless form, for each internal node (i goes from 2 to N) can be derived from equation 2 by plugging in the corresponding i value.

$$\left(\frac{1}{(\Delta R)^2} - \frac{1}{R_i(2\Delta R)}\right)\theta_{i-1} - \left(\frac{2}{(\Delta R)^2} + \gamma^2\right)\theta_i + \left(\frac{1}{(\Delta R)^2} + \frac{1}{R_i(2\Delta R)}\right)\theta_{i+1} = 0$$

(equation 2)

The distance to each node, R_i, can be generated using equation 3.

$$R_i = C + (i-1)\Delta R \; where \; \Delta R = \frac{1-C}{N} \qquad \text{(equation 3)}$$

Note that the boundary nodes, node (i = 1) and node (i = N+1) require their own equation depending on the assumptions made at each boundary, known as the boundary conditions. At the fin base, Node 1, the temperature is assumed to be the same as that of the tubular surface where the fin is mounted on. This is represented by equation 4.

$$\theta_1 = 1 \qquad \text{(equation 4)}$$

At the fin tip, Node N+1, one of three different assumptions can be made, resulting in three different equations (equations 5A-5C). Only one of these equations at a time can be used to complete the system of equations.

Table 1: *Boundary conditions at the fin tip and the corresponding equation used to complete the system of equations*

Boundary condition assumption	**Equation**	
No heat flow from the fin tip (perfect insulation)	$\theta_{N-1} - 4\theta_N + 3\theta_{N+1} = 0$	(equation 5A)
Finite heat flow from the fin tip by convection	$\theta_{N-1} - 4\theta_N + \left(3 + \gamma^2\tau(2\Delta R)\right)\theta_{N+1} = 0$	(equation 5B)
Infinitely too long fin	$\theta_{N+1} = 0$	(equation 5C)

Variables in equations 2 to 5:

- θ_i is the dimensionless temperature and represents the temperature at each node (these are the unknowns in the system of equations that you will solve)
- $N + 1$ is the number of nodes and i indicates a specific node, $i = 1, 2, 3, \ldots, N, N + 1$
- C is the ratio between the inner and outer radii of the fin ($C = \frac{r_1}{r_2}$)
- R_i is the ratio between the distance to each node and the outer radius of the fin ($R_i = \frac{r_i}{r_2}$) where $R_1 = C$ and $R_{N+1} = 1$
- γ^2 is known as the enlarged Biot number and represents how quickly heat transfers through the fin $\left(\gamma = \sqrt{\frac{h}{kt} r_2^2}\right)$
- τ is defined as the ratio between the actual thickness and the outer radius of the fin ($\tau = \frac{t}{r_2}$) and typically is in the range of 0.01 to 0.1

Problem PP PartA: Solving the radial fin problem with numerical methods

1. Using equations 2, 4, and 5A, write the system of equations for 6 nodes ($N = 5$) using the provided variables:
   $N = 5$ resulting in i = 1, 2, 3, 4, 5, 6
   $C = 0.2$
   $\gamma = \sqrt{10}$
2. Solve your system of equations using MATLAB to determine the temperature distribution. Compare your results with the provided results below using `format long`:

$$\theta_1 = 1.000000000000000$$
$$\theta_2 = 0.486132897434888$$
$$\theta_3 = 0.260949304501633$$
$$\theta_4 = 0.153710622042675$$
$$\theta_5 = 0.104256326792430$$
$$\theta_6 = 0.087771561709016$$

3. Generalize your calculations to solve the system of equations for any number of nodes.
   - Generate your **R** vector using equation 3 for all i values.
   - Pre-allocate your A matrix and **b** vector using the **zeros** function.
   - Fill in the middle rows of A and **b** (rows 2 to N) using **for** loop(s) according to equation 2.
   - Fill in row 1 of A and **b** according to equation 4.
   - Fill in row $N+1$ of A and **b** according to equations 5A-5C depending on the provided boundary condition using a branching structure.
   - Once A and **b** are filled, solve for θ using left division.

Create a new function file and save it as ***RadFin_Numerical.m****. Add in the recommended H1 line and help text lines using comments.*

4. Lastly, convert your calculations into a function named **`RadFin_Numerical`** with the following function definition line.

```
[R, T] = RadFin_Numerical(N, C, Tau, Gamma, BC)
```

   - R is a <u>column</u> vector of radii calculated for each node using equation 3
   - T is a <u>column</u> vector of dimensionless temperatures θ at each node calculated using the numerical method.
   - N, C, Tau (τ), and Gamma (γ) are defined above.
   - BC is a variable for the boundary condition at the tip of the fin that may have a value of the integers 1, 2, or 3 only. If BC = 1, use equation 5A, if BC = 2, use equation 5B, and if BC = 3, use equation 5C in your system of equations.

Note: you may use any additional variables that you need and you are free to name your variables any valid names.

Analytical methods:
The numerical method gives us an approximation of the analytical/exact solution. The analytical solution of the radial fin heat problem can be found in equations 6A-C depending on the boundary condition.

The analytical solution for no heat flow from the tip of the fin (perfect insulation/negligible heat loss from the tip) is defined in equation 6A.

$$\theta = \frac{K_1(\gamma)I_0(\gamma R) + I_1(\gamma)K_0(\gamma R)}{K_1(\gamma)I_0(\gamma C) + I_1(\gamma)K_0(\gamma C)} \qquad \text{(equation 6A)}$$

The analytical solution for finite heat flow from the fin tip by convection is defined in equation 6B where $\gamma_1 = (1+\tau)\gamma$.

$$\theta = \frac{K_1(\gamma_1)I_0(\gamma_1 R) + I_1(\gamma_1)K_0(\gamma_1 R)}{K_1(\gamma_1)I_0(\gamma_1 C) + I_1(\gamma_1)K_0(\gamma_1 C)} \qquad \text{(equation 6B)}$$

The analytical solution for infinitely long fin ($r_2 \gg t$) is defined in equation 6C.

$$\theta = \frac{K_0(\gamma)I_0(\gamma R) - I_0(\gamma)K_0(\gamma R)}{K_0(\gamma)I_0(\gamma C) - I_0(\gamma)K_0(\gamma C)} \qquad \text{(equation 6C)}$$

In equations 6A-C, $I_v(z)$ denote the modified Bessel functions of the first kind of order v and $K_v(z)$ denote the modified Bessel functions of the second kind of order v and can be calculated using the **`besselk`** and **`besseli`** functions in MATLAB. Note that **R** and $\boldsymbol{\theta}$ are both vectors containing the radii and the dimensionless temperature, respectively.

Problem PP_PartB: Solving the radial fin problem with analytical methods:

1. Generate your **R** vector using equation 3 for all i values.
2. Perform the analytical calculation based on the boundary condition defined in equations 6A-C using a branching statement. Use the **`besseli`** and **`besselk`** functions to determine $I_v(z)$ and $K_v(z)$ respectively.

   Hint: ***`besseli(nu,z)`*** *computes the modified Bessel function of the first kind,* $I_v(z)$*, for each element of the array z and* ***`besselk(nu,z)`*** *computes the modified Bessel function of the second kind,* $K_v(z)$*, for each element of the array z.*

Create a new function file and save it as ***RadFin_Analytical.m****. Add in the recommended H1 line and help text lines using comments.*

3. Convert your calculations into a function named **`RadFin_Analytical`** with the following function definition line.

   `[R, T] = RadFin_Analytical(N, C, Tau, Gamma, BC)`

   - R is a <u>column</u> vector of radii calculated for each node using equation 3
   - T is a <u>column</u> vector of temperatures at each node using the analytical method
   - N, C, Tau (τ), and Gamma (γ) are defined before
   - BC is a variable for the boundary condition at the tip of the fin. If BC = 1, use equation 6A, if BC = 2, use equation 6B, and if BC = 3, use equation 6C in your system of equations.

Note: you may use any additional variables that you need and are free to name your variables any valid names.

Fin efficiency:
The fin efficiency, η_{fin}, is a measure of the fin performance. It represents how much heat is actually transferred by the fin out of the maximum heat that can be transferred through the ideal fin with $0 \leq \eta_{fin} \leq 100\%$.

The fin efficiency can be calculated using equation 7 where the limits on the finite integral indicate the start of the fin (C) to the end of the fin[4].

$$\eta_{fin} = \frac{2\int_C^1 \theta R dR}{(1 - C^2)} \qquad \text{(equation 7)}$$

This equation requires the calculation of a definite integral. Most of the time in computations (programming), calculating an integral requires numerical integration methods where the integral is estimated through summation of function values. For this project, we will use a composite method by calculating two different integrals and adding them together.

Method 1 can be used to calculate the integral of a function sampled at an even number of intervals and is found in equation 8.

$$I(f) = \int_a^b f dx \cong \frac{h}{3}\left[f(a) + 4\sum_{i=2,4,6,..}^{M} f(x_i) + 2\sum_{j=3,5,7,..}^{M-1} f(x_j) + f(b)\right] \qquad \text{(equation 8)}$$

Method 2 can be used to calculate the integral of a function sampled at a number of intervals divisible by 3, i.e. 3, 6, 9, 12, etc, and is found in equation 9.

$$I(f) = \int_a^b f dx \cong \frac{3h}{8}\left[f(a) + 3\sum_{i=2,5,8,..}^{M-1} [f(x_i) + f(x_{i+1})] + 2\sum_{j=4,7,10,..}^{M-2} f(x_j) + f(b)\right] \qquad \text{(equation 9)}$$

The number of function intervals, M, is defined as the number of samples in the function minus one (M = length(function vector) – 1). f is a subset of the θR vector such that $h = \frac{b-a}{M}$ where a and b are the limits on the finite integral such that $f(a)$ is the first number in the

[4] A. Compo and S. Chikh, “Reproduction of the Fin Efficiency Diagram for Annular Fins of Uniform Thickness by Solving Systems of up to Four Algebraic Equations,” International Journal of Mechanical Engineering Education, Volume: 34-1, pp 85-92, 2006.

sub-vector and $f(b)$ is the last number in the sub-vector. A combination of the two methods can be used to calculate the full integral.

Example:

If the θR resulted in a vector of 30 values, the following illustrates how the $\int_C^1 \theta R dR$ integral can be estimated. The number of intervals ($M = 30 - 1 = 29$) is not even nor divisible by 3; however, it can be broken up into two subsets: one that is 3 (divisible by 3) and one that is 26 (even). Effectively, $\int_C^1 \theta R dR = \int_C^{R_4} \theta R dR + \int_{R_4}^1 \theta R dR$ where R_3 is the radius at the 3rd node. The first integral is calculated using method 2 with 3 intervals (4 function values) and the second integral is calculated using method 1 with the remaining 26 intervals (27 function values).

Problem PP_PartC: Determining the radial fin efficiency using numerical integration

*Create a new function file and save it as **Integral_Numerical_1.m**. Add in the recommended H1 line and help text lines using comments.*

1. Create a function that performs numerical integration using method 1 (equation 8) with the following function definition line:

```
I = Integral_Numerical_1(f, M, a, b)
```

- I is the calculated integral
- f if the vector of function values (θR)
- M is the number of function values in f, M must be an even number
- a is the lower limit of the finite integral
- b is the upper limit of the finite integral

*Create a new function file and save it as **Integral_Numerical_2.m**. Add in the recommended H1 line and help text lines using comments.*

2. Create a second function that performs numerical integration using method 2 (equation 9) with the following function definition line:

```
I = Integral_Numerical_2(f, M, a, b)
```

- I is the calculated integral
- f if the vector of function values (θR)
- M is the number of function values in f, M must be divisible by 3
- a is the lower limit of the finite integral
- b is the upper limit of the finite integral

3. Calculate the fin efficiency (η_{fin}) using your **`Integral_Numerical_1`** and **`Integral_Numerical_2`** functions.

- Calculate the θR vector using the two inputs.
- Check the length of the θR vector and calculate the number of intervals. Select the proper numerical integration method to use:
    - If M is an even number, use **`Integral_Numerical_1`** to calculate the integral.
    - If M is divisible by 3, use **`Integral_Numerical_2`** to calculate the integral.
    - If M is not even or divisible by 3, divide the integral into two separate integrals, I1 and I2, where I1 is calculated with the first four values in θR with **`Integral_Numerical_2`** and the I2 is calculated with the last M-4 values in θR with **`Integral_Numerical_1`**. The total integral is determined as the sum of both integrals, I = I1 + I2.
- Calculate the fin efficiency (η_{fin}) using equation 7. Note C is the first value in the **R** vector.

*Create a new function file and save it as **FinEfficiency.m**. Add in the recommended H1 line and help text lines using comments.*

4. Convert your calculations into a third function with the following function definition line:

```
nfin = FinEfficiency(R, T)
```

- nfin is the calculated fin efficiency
- R is the vector of radii
- T is the temperature distribution calculated using the numerical or analytical solutions

Note: you may use any additional variables that you need and are free to name your variables any valid names.

Problem PP_PartD: Comparison between the numerical and analytical solutions for radial fin

*Create a new script file and save it as **PP_PartD_ubit1ubit2.m**. Add the required comments at the top. You are free to use an old HW m file or the template from UBlearns.*

1. In your script file, prompt the user for the values specified in Table 2. For each value entered by the user, use a **`while`** loop to validate each value according to Table 2 to ensure that the user enters in an appropriate value. You may name your variables whatever you want.

Table 2: *Variables to be entered by the user identifying the prompts to use and the valid values that must be checked with a user-validation loop*

Value	Prompt	Valid values
N	`Enter in the number of nodes:`	N > 1
C	`Enter in the ratio between node 1 and node N:`	0 < C < 1
τ	`Enter in the ratio between the fin thickness and fin length:`	$0 < \tau < 1$
γ	`Enter in the square root of Biot number:`	$\gamma > 0$
BC	`Enter in the boundary condition (1 – insulated, 2 – convection, 3 – infinite fin):`	BC = 1, 2 or 3

2. Use your **`RadFin_Numerical`** and **`RadFin_Analytical`** functions to determine the temperature distribution from your numerical calculations and your analytical calculations, respectively.
3. Store your results as column vectors named **`R_num`** and **`T_num`** to store your numerical results and **`R_ana`** and **`T_ana`** to store your analytical results.
4. Calculate the fin efficiency (η_{fin}) using your **`FinEfficiency`** function using both your numerical and analytical results. Store your results in variables named **`nfin_num`** and **`nfin_ana`**, respectively.
5. Plot the results, θ versus R, for both the numerical results and analytical results in the same figure. Use different line styles and widths to distinguish between the two results. Fully annotate your plot with a title, axis labels, etc. Your legend should contain text in the following format where X.XXXXXX indicates your results from calculating η_{fin}.

```
Fin Efficiency from Numerical Solution = XX.XXX %
Fin Efficiency from Analytical Solution = XX.XXX %
```

6. Test your script with the following data:
   - Enter in invalid values to check your user-validation `while` loops
   - Use your program with the data from Table 3 and save your plots to insert in to your final report

Table 3: *Cases to be run for Part D to compare the results between the numerical solution and the analytical solution*

Case	N	C	τ	γ	Type of boundary condition
1	30	0.15	0.1	3	1
2	30	0.15	0.1	3	2
3	30	0.15	0.1	3	3
4	60	0.1	0.05	5	1
5	60	0.2	0.05	5	1
6	60	0.5	0.05	5	1

Problem PP_PartE: Developing the diagram of radial fin efficiency

*Create a new script file and save it as **PP_PartE_ubit1ubit2.m**. Add the required comments at the top. You are free to use an old HW m file or the template from UBlearns.*

1. In your script file, use the following values. You may name your variables whatever you want.
   $N = 40$
   $\tau = 0.05$
   $BC = 2$ assuming a convective boundary condition
   $C =$ a vector from 0.1 to 0.9 with 5 values
   $\gamma =$ a vector from 0.0 to 3 with 16 values
2. Calculate the fin efficiency (η_{fin}) for every (C, γ) combination using your **`RadFin_Numerical`** and **`FinEfficiency`** functions.
   - Initialize a matrix named **`nfin_partE`** as a **16 x 5 array** of zeros.
   - Use nested for loops (one for C and one for γ) to go through all possible combinations of C and γ. Store your results in **`nfin_partE`** where each column contains the fin efficiencies for every γ for a single C value.
3. In one figure, plot your results as γ versus η_{fin} resulting in one curve per every C value, i.e. your plot will be 5 lines in one figure. Use different line/marker colors distinguish between results. Fully annotate your plot with a title, axis labels, legends, etc.

Problem PP_PartF: Calculate results using data from a file

*Create a new script file and save it as **PP_PartF_ubit1ubit2.m**. Add the required comments at the top. You are free to use an old HW m file or the template from UBlearns.*

1. Read in the data from the **`RadFin.txt`** file. The data in the file is arranged as shown in the Table 4 below. Each row represents a different radial fin.

Table 4: *The data stored in the file RadFin.txt*

r_1 [m]	r_2 [m]	t [m]	h [W/m²K]	k [W/mK]	T_b [C]	T_∞ [C]
0.05	0.5	0.005	30	50	75	20
0.06	0.5	0.005	35	55	80	21
0.07	0.5	0.005	40	60	85	22
⋮	⋮	⋮	⋮	⋮	⋮	⋮

2. Use the data from the file and the following equations to calculate the fin efficiency (η_{fin}) for each row of data. Store your results in a **21 x 1 vector** named **`nfin_partF`**. All other variables can be named whatever you want.
$N = 40$
$BC = 2$ assuming a convective boundary condition at the tip
$$\tau = \frac{t}{r_2}$$
$$C = \frac{r_1}{r_2}$$
$$\gamma = \sqrt{\frac{h}{kt} r_2^2}$$

3. Calculate the overall rate of heat transfer ($\dot{Q}_{fin}$) for the fin in each row using the following equation. Store your results in a **21 x 1 vector** named **`qfin_partF`**.

$$\dot{Q}_{fin} = \eta_{fin}[2\pi(r_2^2 - r_1^2)h(T_b - T_\infty)]$$

4. Combine the original data with your calculations for γ, C, τ, η_{fin}, and $\dot{Q}_{fin}$ so that your final matrix is a **21 x 12** where the columns represent the values according to the table header below. Write out your final set of data to a new file named **`RadFin_Results.txt`**.

r_1 [m]	r_2 [m]	t [m]	h [W/m²K]	k [W/mK]	T_b [C]	T_∞ [C]	γ	C	τ	η_{fin}	$\dot{Q}_{fin}$

Problem PP_PartG: Project final report

1. Download the programming project report template from UBlearns.
2. Write your introduction and description of code sections as described in the document.
3. Insert the following results.
   - The six figures comparing the numerical versus analytical results generated using your PP_PartD_ubit1ubit2.m script file.
   - The figure comparing the fin efficiency's generated across different C and γ values generated from your PP_PartE_ubit1ubit2.m.
   - A table showing the data calculated in the RadFin_Results.txt file from your PP_PartF_ubit1ubit2.m script file. Your table must include all the data with an appropriate header row.
4. Write your conclusion section as described in the document.
5. Finish up by completing your summary/abstract section, list of tables and figures (8 figures and 1 table at minimum from your results), and table of contents.
6. Remove any remaining red text and complete the title page.
7. Save your report as a PDF and name it PP_Report_ubit1ubit2.pdf.

Submission Process

Zip all your files into a zip file named PP_GroupName.zip. This file must contain all your .m files, text file, and your PDF report. Your zip file should contain the following files:

Table 5: *The files to be submitted in your zipped folder*

Filename	Type
RadFin_Numerical.m	Function file
RadFin_Analytical.m	Function file
Integral_Numerical_1.m	Function file
Integral_Numerical_2.m	Function file
FinEfficiency.m	Function file
PP_PartD_groupName.m	Script file
PP_PartE_groupName.m	Script file
PP_PartF_groupName.m	Script file
PP_Report_groupName.pdf	PDF file containing your report

Submit your zip file to the Programming Project assignment on UBlearns.

RUBRICS

25	**Part A: RadFin_Numerical.m**
1	Function file named RadFin_Numerical.m exisits
2	Number of input and output arguments are correct
1	H1 line contains an adequate description
2	Help text lines are complete describing the inputs, outputs, and example function call
2	The vector of node radii (R) is correctly generated
2	Array A and vector b were preallocated correctly
1	A(1,1) and b(1) were assigned correctly
5	for loop for A and b for internal nodes are written correctly
8	The switch structure for BC at the fin tip is written correctly
1	The temperature distribution is correctly calculated as T = A\b
15	**Part B: RadFin_Analytical.m**
1	Function file named RadFin_Analytical.m exisits
2	Number of input and output arguments are correct
1	H1 line contains an adequate description
2	Help text lines are complete describing the inputs, outputs, and example function call
1	The vector of node radii (R) is correctly generated
8	The branching structure is written correctly and the the temperature distribution is correctly calculated for each boundary condition
10	**Part C: Integral_Numerical_1.m**
1	Function file named Integral_Numerical_1.m exists
1	Number of input and output arguments are correct
1	H1 line contains an adequate description
1	Help text lines are complete describing the inputs, outputs, and example function call
6	Integral is correctly calculated
10	**Part C: Integral_Numerical_2.m**
1	Function file named Integral_Numerical_1.m exisits
1	Number of input and output arguments are correct
1	H1 line contains an adequate description
1	Help text lines are complete describing the inputs, outputs, and example function call
6	Integral is correctly calculated
10	**Part C: FinEfficiency.m**
1	Function file named FinEfficiency.m exisits
1	Number of input and output arguments are correct

1	H1 line contains an adequate description
1	Help text lines are complete describing the inputs, outputs, and example function call
6	The branching structure for using the numerical integral functions is written correctly and the fin efficiency is correctly calculated.
25	**Part D: PP_PartD_groupName.m**
1	Script file named PP_PartD_groupName.m exisits
2	Contains adequate comments throughout code
10	Prompts user for N, C, τ, γ, and the type of boundary condition with user validation loop for each input
3	Contains variables named R_num, T_num, and nfin_num and functions are called correctly
3	Contains variables named R_ana, T_ana, and nfin_ana and functions are called correctly
6	Plots the results
15	**Part E: PP_PartE_groupName.m**
1	Script file named PP_PartE_groupName.m exisits
2	Contains adequate comments throughout code
2	C, and γ vectors are assigned correctly
5	For loop for calling the fin efficiency and assigning value to a variable named nfin_partE with the correct values
5	Plots the results
20	**Part F: PP_PartF_groupName.m**
1	Script file named PP_PartF_groupName.m exisits
2	Contains adequate comments throughout code
8	Correctly loads the data and creates/calculates values for N, C, τ, γ, and the type of boundary condition
4	Contains a for loop for a variable named nfin_partF with the correct values
3	Contains a variable named qfin and calculated correctly
2	Writes out the results to a text file with correct values
20	**Report: PP_Report_groupName.pdf**
1	PDF file named PP_Report_groupName.m exisits
2	Title page, table of contents, and list of tables and figures complete
3	Summary/abstract
2	Introduction
4	Description of code
4	Results
4	Conclusions