PRELAB EXERCISE 2: EXTENDED SURFACES

ME 436 Heat Transfer

Introduction

The primary objective of this experiment is to theoretically and experimentally study an *extended surface* subjected to varying environmental conditions. More specifically, a heated cylindrical rod, lined with evenly-spaced thermocouples, will be examined under passive, medium, and high air flow rates - simulating free, medium, and high convective conditions. The temperature distributions and heat transfer rates will be calculated using three separate models; the most accurate of which will ultimately be selected. Finally, performance metrics are used to assess the effectiveness and efficiency of the extended rod for use as a fin.

This pre-lab assignment will act as a guide towards creating a mathematical model before attending lab – saving critical in-class time for performing experiments and analyzing data.

Prerequisites

Before attempting this pre-lab assignment, it is imperative that you:

- Review *textbook section 3.6*, particularly *Table 3.4 (pg. 161)*,
- Review *experiment procedures*, and
- Watch the *pre-lab videos* on Blackboard (Bb).

Getting Started

First, if you have not already, you will need to download the starter code from Bb (located in the 'Lab 2' directory) and unzip its contents into the directory where you wish to complete the exercise. Be sure to have completed all of the prerequisites before attempting this assignment.

Files included in this exercise:

Once you have unzipped the contents of the starter package, you should have the following files:

- **ex2.m**
- /lib
- caseA.m

[⋆] caseB.m
[⋆] caseD.m
[⋆] calc_eta.m
[⋆] calc_epsilon.m
[†] ex2_rad.m
[†] calc_rad.m
[†] calc_conv.m

⋆ indicates files that you will need to complete.
† indicates files that are optional.

Throughout this pre-lab exercise, you will be using the script **ex2.m**, but will only be required to edit *one section*, in which you will need to enter material properties, defined in the ***laboratory procedures***.

0. Environment Setup

Before we can get started, we need to setup our MATLAB environment properly. As always, if using the laboratory computers, be sure to be running your code from the C:\temp directory, and also make sure you have unzipped your code properly. Otherwise you may receive '`./lib not found`' errors.

0.1. MATLAB Script Outline

Once your environment is setup, open the file named **ex2.m** and read the instructions at the top of the script. In short, this script will plot your data and fit the math models described in ***Table 3.4*** to your collected dataset. A brief outline of the script is shown below, followed by a more detailed line-by-line explanation.

Now, in the MATLAB editor, scroll down to the SETUP section. This portion of the script is where the paths to our data are set. More specifically, pDIR is the *folder* in which **all three datasets** are stored. You may provide the absolute path to the folder, or simply drag and drop your data into the ./data folder.

> *Note:* - When performing the experiment, make sure you have ***all three sets of data*** collected before running this script.

```
clear; close all;
Set: path to data;
Set: properties;

for i = 1 to 3, do
    load: excel dataset(i);
        [dat] ← xlsread(path);
    Set: average SS temps;
        [Tm] ← mean(dat);

    Set: theoretical temps;
        [T_A, q_A] ← caseA();
        [T_B, q_B] ← caseB()*;
        [T_D, q_D] ← caseD()*;

    Set: performance;
        η ← calc_eta()*;
        ε ← calc_epsilon()*;
end
Plot:  T_{A,B,D} vs. T_m;
Print:  q & ε, η to screen;
```

As written, this script locates the data'data' folder and grabs **all excel files** therein. The command `dir( [...  '/*.xlsx'])` returns the directory (path), of each file that ends with `.xlsx`. This is denoted by the ' `*.xlsx`' syntax, in which the '`*`' is a *wildcard* command meaning, in this case, *'grab everything with the following suffix'*.

```
%% SETUP
% set path to folder containing excel sheets
pDIR = './data';

% grab all excel files in directory
xl_files = dir( [pDIR '/*.xlsx']);
NX = length(xl_files);
```

The Excel sheet paths are then stored and will be used later in the *loop* section. Next, we have a few *string* variables that will be used for plotting later. It's important that these are set correctly.

```
% plot title
titles{1,1} = 'Temperature Distribution: h = 9 [W/m^2 C]';
titles{1,2} = 'Temperature Distribution: h = 30 [W/m^2 C]';
titles{1,3} = 'Temperature Distribution: h = 40 [W/m^2 C]';

% export figure name
str_print{1,1} = 'h_09';
str_print{1,2} = 'h_30';
str_print{1,3} = 'h_40';
% set station numbers for each dataset
st = {'1';'3';'5';};
```

Note: The *wildcard* command ' ∗ ' will automatically sort your filenames based on numerical/alphabetical order. Hence, be sure to name your excel sheets accordingly, so that the titles above are correct.

1.   Input Properties

Now, we must input a few properties from the experiment procedures

```
%% =========== Part 1: Input Properties =============
% From the procedures document, fill in all of the necessary info
```

First, we *define* a few global variables. These are special variables that once set, will always be available to us. That is, we don't need to pass them in/out of functions.

```
% set globals
global D k P L Ac Af T_inf
```

Now, we arrive to the *fin properties* section.

To do: *Using information provided in the procedures document, locate and correct the incorrect values:*

```
% Fin Properties **FIX ME**
h = [9, 30, 40];            % [W/m^2 C]
D = 1;                      % (Diameter) [m]
k = 400;                    % (Conductivity, Brass) [W/m C]
P = pi * D;                 % (Perimeter) [m]
L = 1;                      % (Length of fin) [m]
Ac = 1;                     % (Cross Sectional Area) [m^2]
Af = pi * D * L;            % (Fin Surface Area) [m^2]

% Set Tinf ─── as measured in lab
T_inf = 23.5;     % [C]
```

Note: The `Ac` and `Af` values refer to the *cross-sectional area* and *total fin surface area*, respectively. These also need to be set correctly. Once this is done, we can move on to the main loop.

2.   Plotting Loop

This section now uses a simple `For-Loop` to process each data set automatically. While there are six stations in the lab, this assumes only ***three are being processed at once.***

As you walk through the inner-workings of the main loop, much of this should look familiar. You do not need to do anything here, but be sure to understand what is being done.

```
%% =========== Part 2: Loop over data & Plot =============

% loop over stations
for ii = 1:3
```

2.1. Plotting: Cases A, B, D

However, once we get down to the following lines:

```matlab
% The fin excess temperatures
theta_x = Tm − T_inf;
theta_b = Tb − T_inf;
```

We are now ready to set our *theoretical* temperature distributions, as defined by three boundary conditions (BC) at $(x = L)$, shown in **Table 3.4**.

- **Case A**: Convective BC,
- **Case B**: Adiabatic BC,
- **Case D**: Infinite Length BC.

Functions for each of these cases are implemented in the code below:

```matlab
% m & M
M = sqrt((h(ii) * P * k * Ac)) * theta_b;
m = ((h(ii) * P) / (k * Ac))^0.5;
mL = m * L;
h_mk = h(ii)/(m*k);

% Case A
[TA, qA(ii)] = caseA(x, theta_b, m, M, h(ii));

% Case B
[TB, qB(ii)] = caseB(x, theta_b, m, M);

% Case D
[TD, qD(ii)] = caseD(x, theta_b, m, M);
```

Here, we are setting the values from **Table 3.4 from the textbook**. In summary, we are now inserting the equations the *temperature distribution*, $T(x)$ and the *heat rate*, q_f. The quantities `m` & `M` have already been defined. What still needs to be done, is for you to complete **Cases B and D**; Case A has already been done as an example.

To complete this portion of the exercise, open the files **caseB.m** & **caseD.m** to complete the expressions below:

```matlab
function [T, q] = caseD(x,theta_b, m, M)
% CASED() Calculates fin temperature distribution and heat rate (see table
% 3.4 in text.)
```

```matlab
% ====================== YOUR CODE HERE ======================
% Instructions: Provide two expressions: Temperature distribution, T, and
% heat rate, q, from Table 3.4.
%
% Note: you will need to solve for T, not theta/theta_b
% Temperature Distributio
T = ????

% Heat Rate
q = ???
```

2.2. Efficiency & Effectiveness

Next, as a way to evaluate the performance of each fin, we compute **fin effectiveness** and the **fin efficiency**:

```matlab
% efficiency (eta)
eta_A(ii) = calc_eta(qA(ii), theta_b, h(ii));
eta_B(ii) = calc_eta(qB(ii), theta_b, h(ii));
eta_D(ii) = calc_eta(qD(ii), theta_b, h(ii));

% effectiveness (epsilon)
ep_A(ii) = calc_epsilon(qA(ii), theta_b, h(ii));
ep_B(ii) = calc_epsilon(qB(ii), theta_b, h(ii));
ep_D(ii) = calc_epsilon(qD(ii), theta_b, h(ii));
```

Your task is to open **`calc_eta.m`** and **`calc_epsilon.m`** and complete these functions.

Now, if everything was done correctly, upon running **`ex2.m`**, you should have something similar to Fig. 1 below.
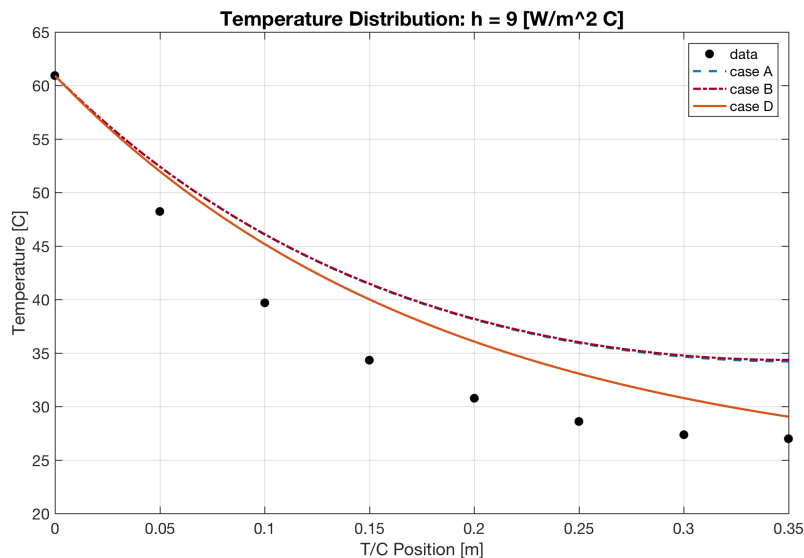
Figure 1: *Natural Convection: h = 9 (W/m^2C)*

In addition, your code should have printed several lines to the command line. To gauge whether your code is correct, you should have:

```
Effectiveness [—]:

  Station    CaseA    CaseB    CaseD
  ───────    ──────   ──────   ──────
  '1'        70.264   70.181   73.333
  '3'        40.095   40.091   40.166
  '5'        34.764   34.763   34.785
```

3.   Export Figures & Deliverables

Now, lets export our figures to PNGs. To do so, scroll down to the following line:

```
    % print figs
    %print(['figs/' str_print{1,ii}], '—dpng','—r200');
```

Uncomment the second line above. That is, you should see:

```
    % print figs
    print(['figs/' str_print{1,ii}], '—dpng','—r200');
```

Now, once you re-run your script, this will export several PNGs and save them into the `./figs` folder.

> **Deliverable 1.** *Export Figures:*   In a word processor (*MS Word, Pages, Open Office, or equivalent*) insert the three PNG images that you just produced (*i.e.,* `h_09.png`, `h_30.png`, `h_40.png`). Then, ***write 1-2 sentences*** describing the similarities/differences between each figure.

> **Deliverable 2.** *Cmd Output:*   Next, copy/paste the output from the MATLAB Command Window (*i.e.,* the tables showing Heat rate, error, efficiency, and effectiveness)

Once these deliverables are completed, print out your document and hand it in at the beginning of class.

4.   Radiation (Optional)

This section is optional and will not be graded as part of the pre-lab assignment. However, since we will be using it throughout the experiment, it has been included.

In this experiment, we are primarily interested in studying the effects of conduction and convection for an *extended surface*. That is, heat is transferred through the extended rod via conduction, and is presumed to be dissipated via convection. However, we know that radiation always plays a large role in this too. The question we now seek to answer is ***how large*** of a role does radiation play? Particularly in relation

to *convection*. In the sections that follow, we will estimate and compare the heat dissipated via convection and radiation for each of the experiment setups studied above.

4.1. Background: Radiation

First, we take a look at the experiment through the lens of an infrared camera, as seen below:
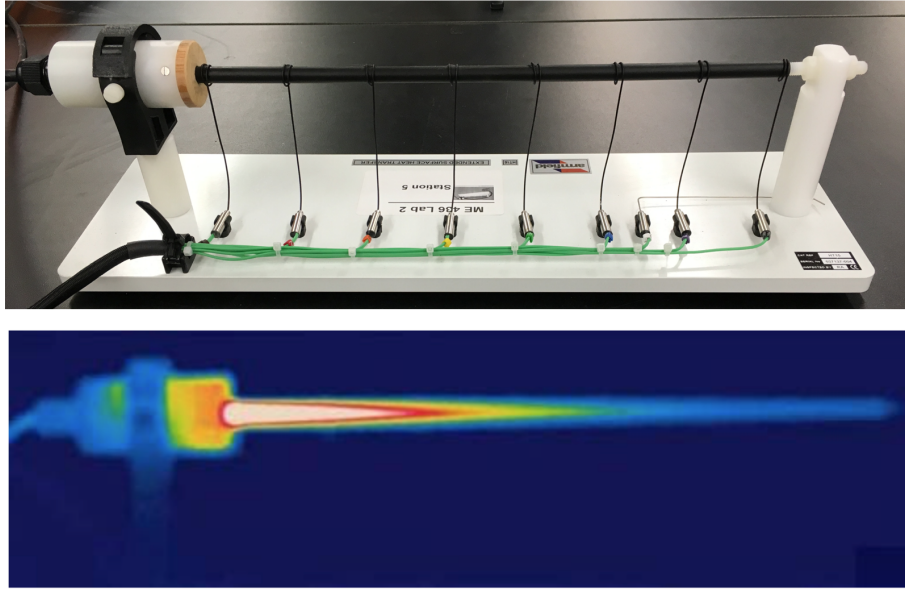
Figure 2: *FlIR Camera output*

This image shows us that we are dissipating a considerable amount of heat near the LHS of the apparatus; particularly along the first three thermocouples. In order to quantify this heat dissipation, lets make a few '*back of the envelope*' estimates.

First, lets assume that radiation may be reasonably computed using the following equation:

$$q_{rad} = \epsilon \sigma A_s (T_{avg}^4 - T_\infty^4) \ . \tag{1}$$

In which, $\sigma = 5 \times 10^{-5} \, (\mathrm{W \cdot m^{-2} \cdot K^{-4}})$, is the Stefan-Boltzmann constant, ϵ is the emissivity, A_s, the surface area, and T_{avg}, is the average material temperature, and T_∞ refers to the surroundings (all units are standard metric).

Now, since the *heat rate*, q, is essentially the integral of our temperature distribution over a given area, lets make *broad stroke* estimates for A_s and T_{avg}: lets use an average temperature T_{avg}, collected over an area covering only the first

$1/3$ **of the rod length.** This is easily done using the '*boxed averaging*' tool included with the FLIR Camera Toolset. Concretely, we assume A_s is computed by:

$$A_s = \pi D(^1/_3 L) \ .$$

When using the FLIR camera, isolating only this section gives us more confidence in our T_{avg} value, rather than averaging over the length of the rod. In addition, this allows us to use the simplified equation above, which only uses an averaged temperature, T_{avg}.
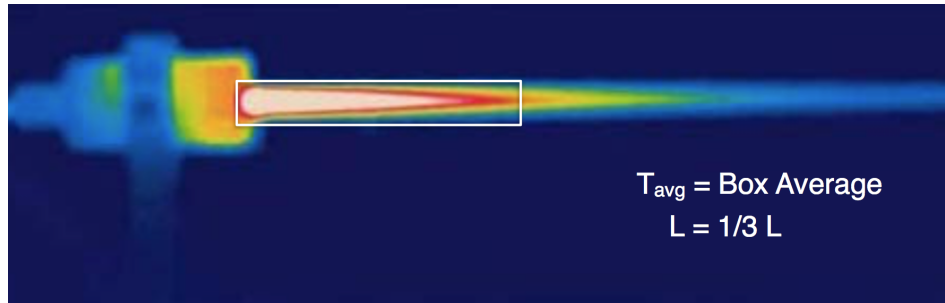
Figure 3: *FlIR Tools: Boxed Averaging*

If we were to sketch this out on paper, our estimate would look similar to the image below.
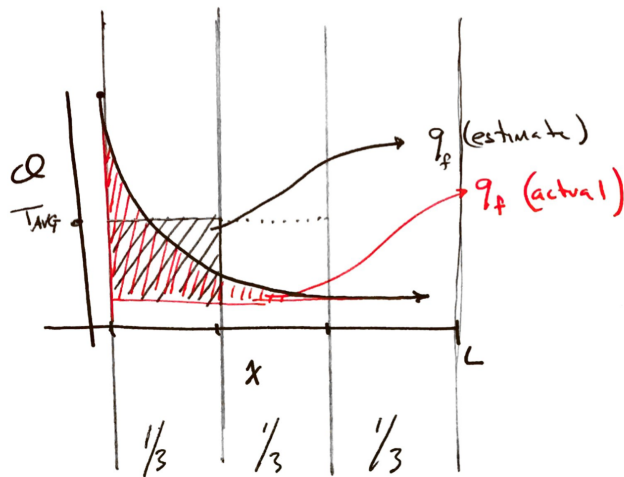
Figure 4: *Estimated heat rate*

As you can see here, the *actual* temperature distribution is an exponential curve. While we could integrate this numerically (since we have the FLIR data), at the moment

we're only interested in a quick *estimate*. Hence, the above approximation should be sufficient.

4.2. Background: Convection

Now, since we have assumed that all other losses are negligible (*i.e.,* conduction, uncertainty in system &, *etc.*), we can use the same arguments made above to estimate the losses due to convection. That is, lets assume that the following equation is sufficient:

$$q_{conv} = hA_s(T_{avg} - T_\infty) \,. \tag{2}$$

In which $A_s = \pi D(^1/_3 L)$, h is the convection coefficient, T_{avg} is again our *box-averaged* temperature.

4.3. Implementation

Now, let's implement the above in MATLAB. When completed, you should have a bar graph that resembles:
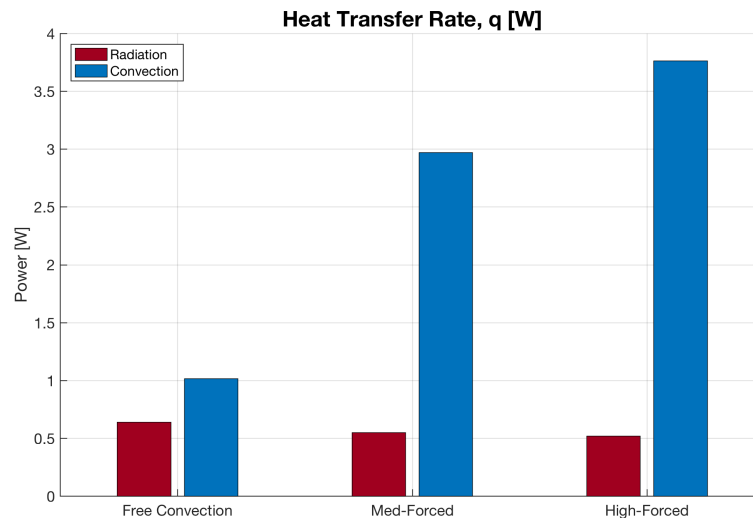
Figure 5: *Estimated heat rate for each station*

First, open the file **ex2_rad.m**. As before, make sure the script can find your data. Be sure that your paths are correct:

```
%% SETUP
% set path to folder containing excel sheets
pDIR = './data';
```

```
% grab all excel files in directory
xl_files = dir( [pDIR '/*.xlsx']);
NX = length(xl_files);
```

Next, we set several properties:

```
% Fin Properties
h = [9, 30, 40];                % [W/m^2 C]
D = 0.01;                       % (Diameter) [m]

% Surface area — Note: we assume 1/3 of the length!
L = 0.33 * 0.35;
As = pi * 0.01 * L;

% boltzmann
sigma = 5.67e-8;              % [W/(m^2 K^4)]
```

All of the above should be sufficient for our work. However, the next few lines may need to be changed:

```
% calibrated ep from FLIR camera (0.78 — 0.82)
ep = 0.82;

% Set Tinf — as measured in lab
Tinf = 23.5 + 273;     % [K]
```

When performing the experiment, you will need to adjust the ambient air temperature, T_{inf} as well as the emissivity (information on this will be provided at a later point – for now, the default value will work fine).

Now, we begin the loop over all data sets.

```
% loop over stations
for ii = 1:3
...
```

The only lines that you need to be concerned with are the following:

```
% radiation & convection estimations
q_rad(ii) = calc_rad(T);
q_conv(ii)= calc_conv(T,h(ii));
```

To get this working, you need to pen `calc_rad.m` & `calc_conv.m` and insert Eq. 1 & Eq. 2, respectively. If executed correctly you should obtain Fig. 5 above, as well as the following command line output:

```
 Heat Transfer Rate [W]:

    Station    CONV     RAD     PCT_RAD

    _____   _____   _____   _____
    'Free'    1.015   0.639   38.6
    'Med'     2.971   0.55    15.6
    'High'    3.764   0.52    12.1
```

4.4. Conclusions

Tabulating the above results, we can now clearly see the relative impact of neglecting radiation for each station. The apparent trend is that as the convection coefficient increases, radiation plays a lesser role

Station	Conv. [W]	Rad. [W]	Pct Rad.
Free	1.0	0.7	***38.6*** %
Med	3.0	0.6	***15.6*** %
High	3.8	0.5	***12.1*** %

Deliverable 3. *In-class assignment* Based upon the results above is it acceptable to neglect radiation for any of the above cases? Justify your answer.