

# 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, be sure to download the starter code from Blackboard (Bb), and *extract* (unzip) its contents to the directory in which you wish to complete the exercise<sup>1</sup>. Before attempting, make sure that you have completed all of the *prerequisites* below.

---

<sup>1</sup>Note: when using the Heat Transfer Lab computers, you must extract your code to the local hard drive: C:/temp/, NOT to the server (ie., Desktop, Group Folder, &, etc.)

*Files included in this exercise:*

Once you have unzipped the contents of the starter package, you should see 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 (at this time).

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. The essential parts of this assignment are highlighted in red.

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. Then, for each dataset you will need to enter three critical pieces of information:

1. Filename,
2. Convection coefficient,  $h$ ,
3. Title to describe the data once plotted

```

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;
    [TA, qA] ← caseA();
    [TB, qB] ← caseB()*;
    [TD, qD] ← caseD()*;

    Set: performance;
    η ← calc_eta()*;
    ε ← calc_epsilon()*;
end
Plot: TA,B,D vs. Tm;
Print: q & ε, η to screen;

```

**Algorithm 1:** Extended Surface Model

Examples are provided to you using the sample dataset:

```

% set 1
fname{1,1} = 's1_h9.xlsx';
h(1) = 9;    % [W/m^2 C]
titles{1,1} = 'Temperature Distribution: h = 9 [W/m^2 C]';

% set 2
fname{1,2} = 's3_h30.xlsx';
h(2) = 30;   % [W/m^2 C]
titles{1,2} = 'Temperature Distribution: h = 30 [W/m^2 C]';

% set 3
fname{1,3} = 's5_h40.xlsx';
h(3) = 40;   % [W/m^2 C]
titles{1,3} = 'Temperature Distribution: h = 40 [W/m^2 C]';

```

Here, it is easiest to simply replace the Excel sheet titles with your own data (be sure that your data has been moved to the ./data folder).

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

## 1. Input Properties

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

```
%% ===== Part 1: Input Properties =====  
% Now, using the procedures document for reference, fill in the necessary  
% information below
```

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. Here you must replace the incorrect values or provide the correct equation (as appropriate):

```
% Fin Properties **FIX ME**  
D = 1; %??? % (Diameter) [m]  
k = 1; %??? % (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 = 22.5; % [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.

Now, when you're ready to continue, comment out the following line:

```
% remove break  
break_msg; dbstack; return;
```

## 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.

### 2.1. Plotting: Cases A, B, D

However, once we get down to the following lines:

```
% 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 (BCs) 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:

```
% 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 for 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 completed as an example.

To complete this portion of the exercise, open the files *caseB.m* & *caseD.m* and insert the correct equation(s) to allow the function to execute properly.

**Note:**

- The function is provided with the following input quantities:  $x$ ,  $\theta_b$ ,  $m$ ,  $M$ . Hence, you **do not** need to define them, just *use* them. **See caseA.m as an example.**
- In addition, the **global** values are available for you to use as well, you do not need to define them. **See caseA.m as an example.**
- The function requires a return value of **temperature**,  $T$ , **not** the *Excess Temperature*,  $\theta/\theta_b$ . Hence, you will need to do a little algebra to solve for  $T$ .

## 2.2. Efficiency & Effectiveness

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

```

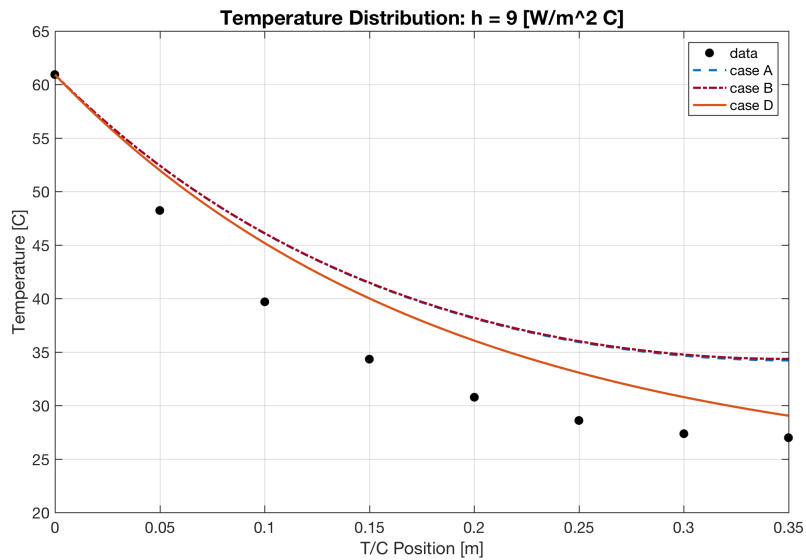
% efficiency (eta)
eta_A(ii) = calc_eta(qA(ii), theta_b, h(ii));
eta_B(ii) = calc_eta(qB(ii), theta_b, h(ii));
eta_D(ii) = calc_eta(qD(ii), theta_b, h(ii));

% 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:  $h = 9 \text{ (W/m}^2\text{C)}$

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

In the `./figs` folder, you should see several images that have been exported automatically. These should be inserted into your report, rather than a screenshot. In addition, the tables have been exported as `.csv` files in the `./output` folder. You can simply copy/paste these into a nice Excel table for proper presentation.

**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.

---

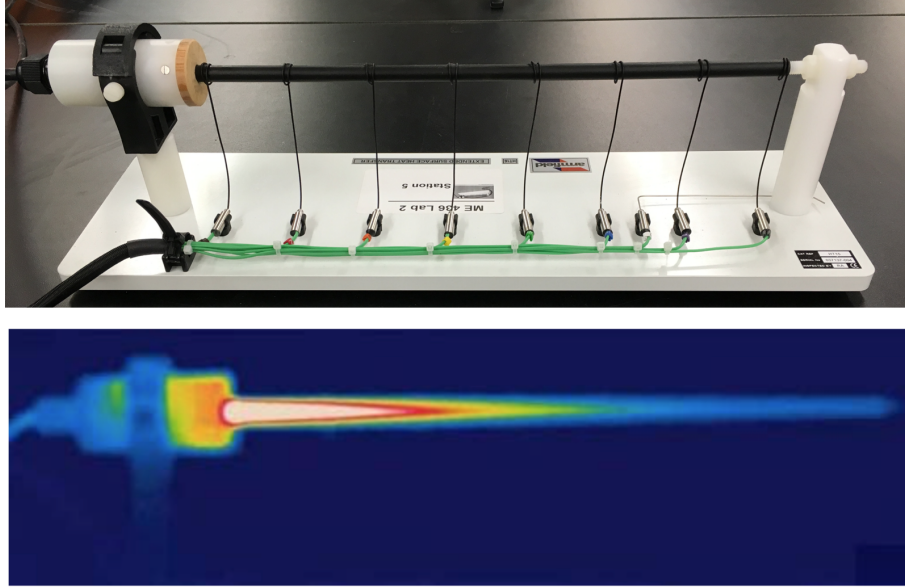
### 4. Radiation (Optional)

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 (and for your report), 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, let's make a few 'back of the envelope' estimates.

First, let's assume that radiation may be reasonably computed using the following equation:

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

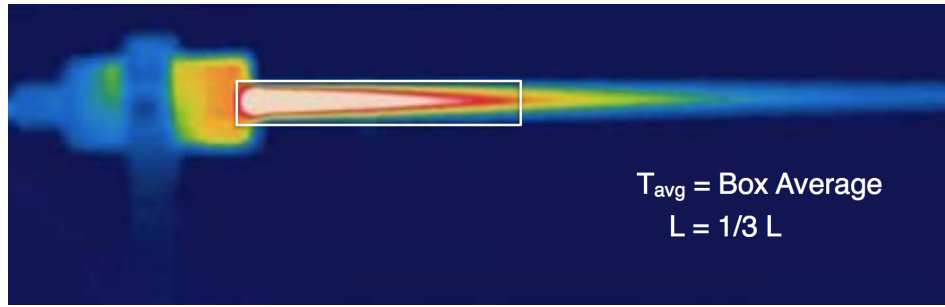
In which,  $\sigma = 5 \times 10^{-8} \text{ (W} \cdot \text{m}^{-2} \cdot \text{K}^{-4}\text{)}$ , is the Stefan-Boltzmann constant,  $\epsilon$  is the emissivity,  $A_s$ , the surface area, and  $T_{avg}$ , is the average material temperature, and  $T_{\infty}$  refers to the surroundings (all units are standard metric).

Now, since the *heat rate*,  $q$ , is essentially the integral of our temperature distribution over a given area, let's make *broad stroke* estimates for  $A_s$  and  $T_{avg}$ : let's 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:

$$A_s = \pi D (1/3 L) .$$

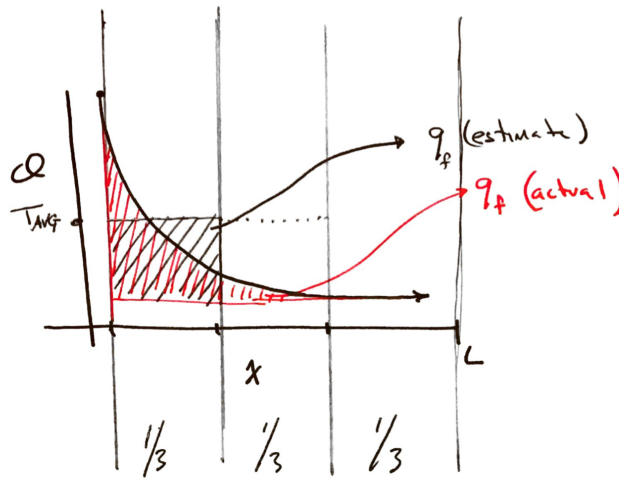
When using the FLIR camera, isolating only this section gives us more confidence in our  $T_{avg}$  value, rather than averaging over the length of the rod. In addition, this allows us to use the simplified equation above, which only uses an averaged temperature,  $T_{avg}$ .





**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

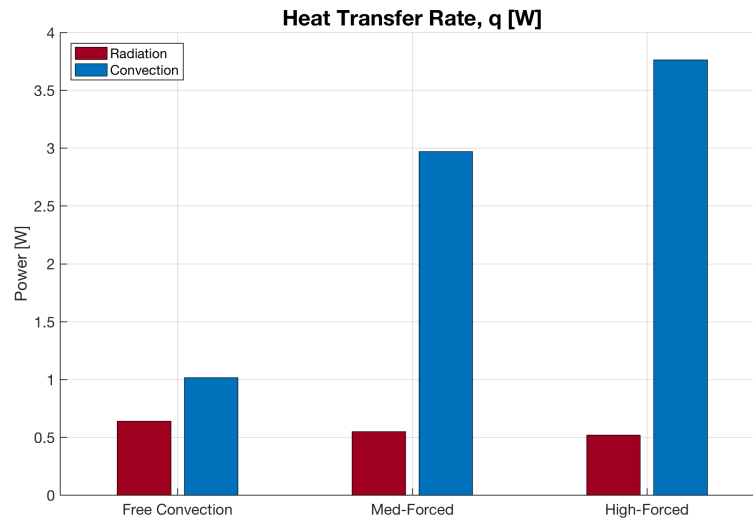
Now, since we have assumed that all other losses are negligible (*i.e.*, conduction, uncertainty in system &, *etc.*), we can use the same arguments made above to estimate the losses due to convection. That is, let's assume that the following equation is sufficient:

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

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 as well as the properties.

Note, you will need to make adjustments to emissivity as more information comes available to you. For instance, using the FLIR camera to extract a accurate value vs. using an estimation from the textbook.

```
% 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:length(fname)
    ...
```

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 open `calc_rad.m` & `calc_conv.m` and insert Eq. 1 & Eq. 2 (above), 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.
<i>Free</i>	1.0	0.7	<b>38.6 %</b>
<i>Med</i>	3.0	0.6	<b>15.6 %</b>
<i>High</i>	3.8	0.5	<b>12.1 %</b>

**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.