Many students have limited or no experience with MatLab. This sheet is designed to get students started with MatLab and specifically for Problem 10 in the **Lin HW**.

1. The first step is downloading MatLab from
   https://edoras.sdsu.edu/ download/matlab.html.
   This step can take a fair amount of time, so plan accordingly.

2. Next establish a named folder on your computer where you are going to store all your Math 337 MatLab files.

   - Appropriate MatLab functions and scripts are available on the
     https://jmahaffy.sdsu.edu/courses/f18/math337/Lectures.html.
   - Some appropriate functions for Problem 10 are
     fishdat.mat, sumsq_vonBert.m, vonBert_plot.m, powerfit.m, allo_plot.m, Hg_plot.m, sumsq_nonlin.m, sumsq_cubic.m, sumsq_Hg.m, sumsq_wtfish.m, wtfish_plot.m.
   - Download these files into your folder (*right click* and *save link as*).

3. The first part of Problem 10 is finding the best fitting parameters to the von Bertalanffy equation:

$$L(t) = L_\infty \left(1 - e^{-bt}\right).$$

   - Begin by double clicking on the program for finding the sum of square errors for this model – The notes use the MatLab function `sumsq_vonBert.m`.
   - This will invoke MatLab, taking you to the directory where this MatLab function is stored.
   - I usually *undock* the **Editor** in MatLab to leave a large **Command** window.
   - It is important that line below the menu points to the directory that has your files.
   - The program is renamed to `sum_vonB` and appears as follows in the MatLab Editor window:

```
function J = sum_vonB(p,tdata,ldata)
% von Bertalanffy eqn
model = p(1)*(1 - exp(-p(2)*tdata));  % Model eqn with
    parameters
error = model - ldata;          % Error between model and data
J = error*error';               % Sum of square errors
end
```

   - In the **Command** window, enter the data as vectors in MatLab:
     `tfish = [1:10]`
     `lfish = [0.77 1.3 1.59 1.72 1.8 1.83 1.85 1.86 1.87 1.87]`
   - These data are used with the MatLab function `fminsearch` and the sum of square errors function `sum_vonB` with the following command:
     `[p1,J,flag] = fminsearch(@sum_vonB,[1.9,0.2],[],tfish,lfish)`

where the arguments in `fminsearch` are the function `sum_vonB` proceeded by the @ sign, the initial guess of the parameters ([1.9,0.2]), null for the options ([]), and the data `tfish,lfish`.

- Assuming the flag = 1 (meaning the program converged), then `p1` gives the best parameters and `J` gives the least sum of square errors.

4. An alternate way to program in MatLab is creating `script` files.

5. The following script file does the first parts of Problem 10, including the best fitting von Bertalanffy equation and the allometric model along with the associated graphs of the models and data:

```
% von Bertalanffy Problem 18

clear;clc;

td = [1 2 3 4 5 6 7 8 9 10];
ld = [0.77 1.3 1.59 1.72 1.8 1.83 1.85 1.86 1.87 1.87];
[p1,J] = fminsearch(@sum_vonB,[1.88,0.5],[],td,ld) % Find
   parameters and SSE
tt = linspace(0,15,200);          % Create domain for function
   on the interval [0,15] with 200 points
ll = p1(1)*(1 - exp(-p1(2)*tt));  % Vector function of model
plot(td,ld,'bo');                 % Graph data
hold on
plot(tt,ll,'r-');grid;            % Graph model
title('von Bertalanffy Model of Marlin','FontSize',16,'FontName'
   ,'Times New Roman');
xlabel('$t$ (yrs)','FontSize',16,'interpreter','latex');
ylabel('Length (m)','FontSize',16,'interpreter','latex');
hold off

print -depsc marlin_len_gr.eps    % Save as .eps file

figure(102)

Ld = [1 1.11 1.16 1.21 1.28 1.38 1.47 1.53 1.68 1.77];
Wd = [7 12 15 19 21 31 32 34 52 65];
lnLd = log(Ld);             % Find log of length data
lnWd = log(Wd);             % Find log of weight data
coef = polyfit(lnLd,lnWd,1)  % Find power law coefficients
a = coef(1);
k = exp(coef(2));
Ll = linspace(0,2,200);     % Domain of length for model
Ww = k*Ll.^a;               % Vector Power law function
plot(Ld,Wd,'ro');           % Graph Weight vs Length data
hold on
plot(Ll,Ww,'m-');grid;      % Graph Weight vs Length model
```

```
34 xlim([0,2]);
35 ylim([0,100]);
36 title('Allometric Model of Marlin','FontSize',16,'FontName','
      Times New Roman');
37 xlabel('Length (m)','FontSize',16,'interpreter','latex');
38 ylabel('Weight (kg)','FontSize',16,'interpreter','latex');
39 hold off
40
41 print -depsc marlin_allo_gr.eps  % Can save in other formats
```

Studying the MatLab programs and commands above will give you much of the needed MatLab for this course.