Assignment 7 ENGR 220

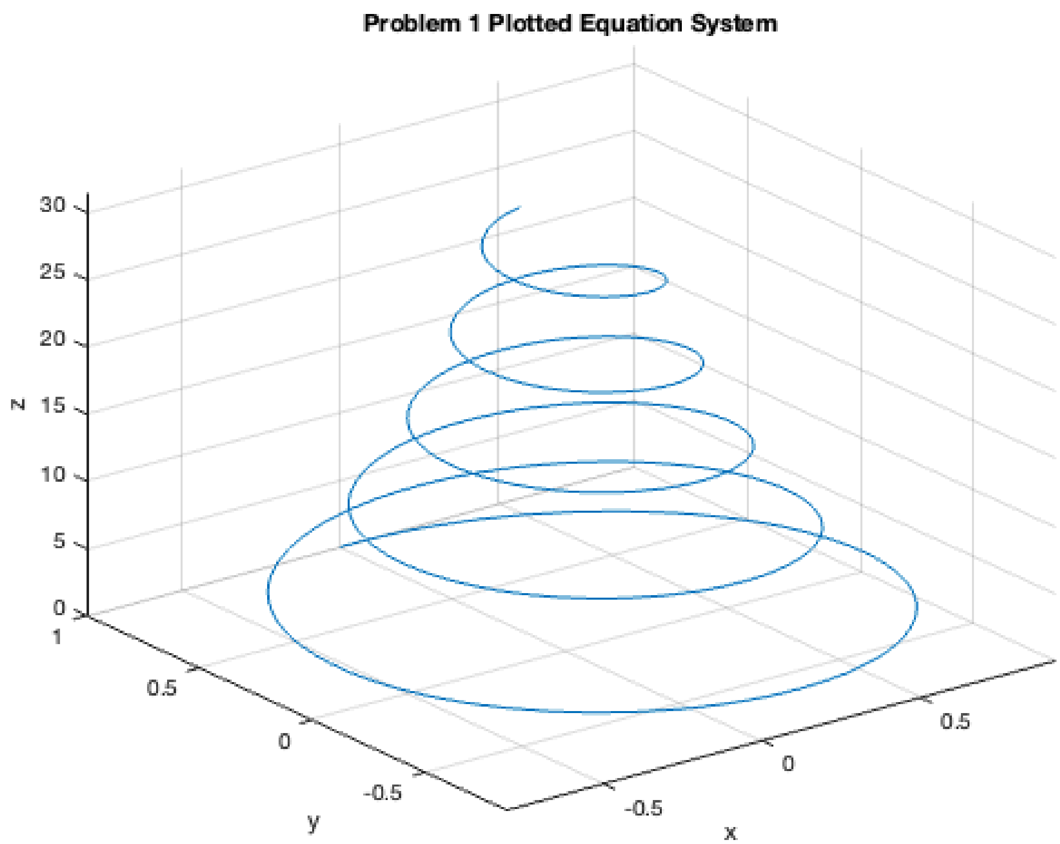
Table of Contents

Ethan Vosburg

Problem 1

Allow the variable "t" to vary over the range 0 to 10π with increments of $\pi/50$. Plot the following equations on an x-y-z orthogonal axis system.

```
clear;
clc;

figure(10);

fplot3(@(t) exp(-0.05*t)*sin(t), @(t) exp(-0.05*t)*cos(t), @(t) t, [0 10*pi]);

% Create title
title({'Problem 1 Plotted Equation System'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
zlabel({'z'});
```



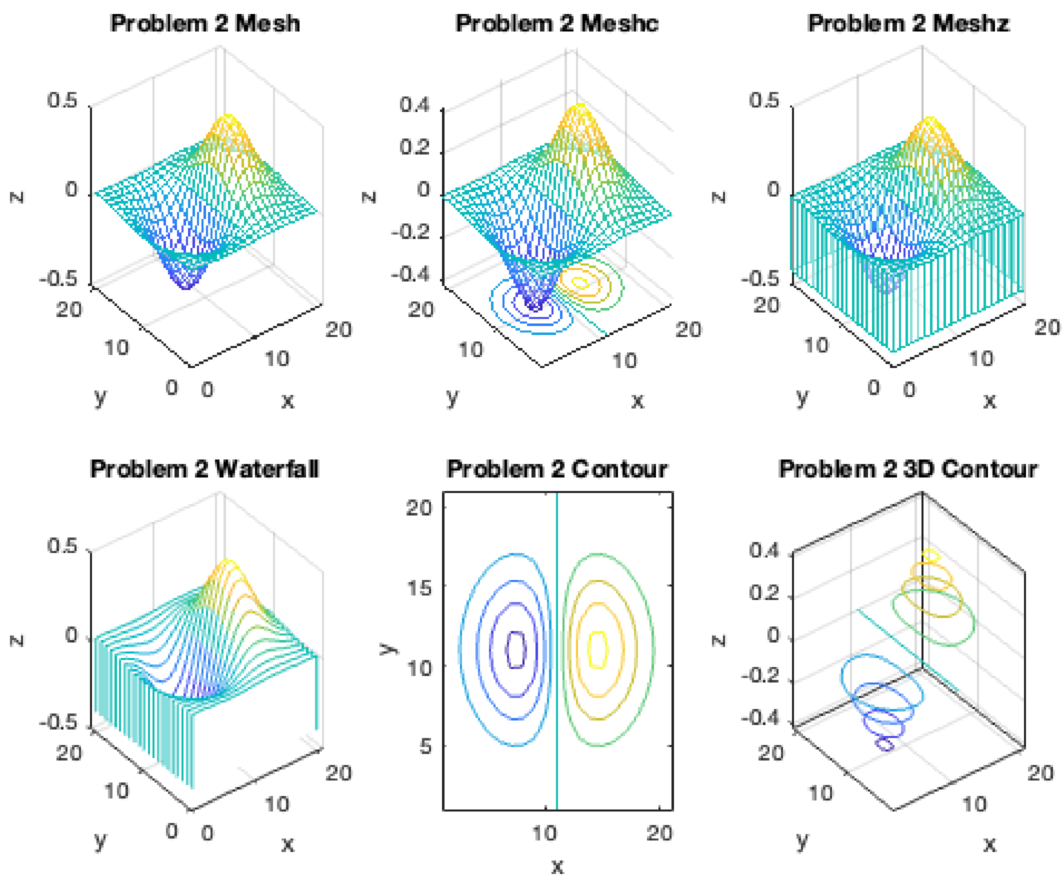
Problem 2

Let x and y vary between -2 and 2 with increments of 0.2. For the function:

```
figure(20);
[X,Y] = meshgrid(-2:.2:2); % Define mesh to modify for each plot
Z = X .* exp(-(X.^2 + Y.^2)); % Apply transformation to mesh

subplot(2,3,1)

% Mesh Plot
meshFig = mesh(Z,'FaceAlpha','0.5');

% Create title
title({'Problem 2 Mesh'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
```

```
zlabel({'z'});


subplot(2,3,2)

% Mesh Plot with Contour
meshcFig = meshc(Z,'FaceAlpha','0.5');

% Create title
title({'Problem 2 Meshc'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
zlabel({'z'});


subplot(2,3,3)

% Mesh Plot with Curtain
meshzFig = meshz(Z,'FaceAlpha','0.5');

% Create title
title({'Problem 2 Meshz'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
zlabel({'z'});


subplot(2,3,4)

% Waterfall Plot
waterfallFig = waterfall(Z);

% Create title
title({'Problem 2 Waterfall'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
```

```
zlabel({'z'});


subplot(2,3,5)

% Contour Plot
contourFig = contour(Z);

% Create title
title({'Problem 2 Contour'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
zlabel({'z'});


subplot(2,3,6)

% 3D contour plot
contour3Fig = contour3(Z);

% Create title
title({'Problem 2 3D Contour'});

% Create ylabel
ylabel({'y'});

% Create xlabel
xlabel({'x'});

% Create zlabel
zlabel({'z'});
```



Problem 3

Quenching is the process of immersing a hot metal object in a bath for a specified time to obtain certain properties (such as hardness). A copper sphere 25 mm in diameter, initially at 300°C, is immersed in a bath at 0°C. The following are the measured data of temperature vs. time (time is the independent variable on the horizontal axis):

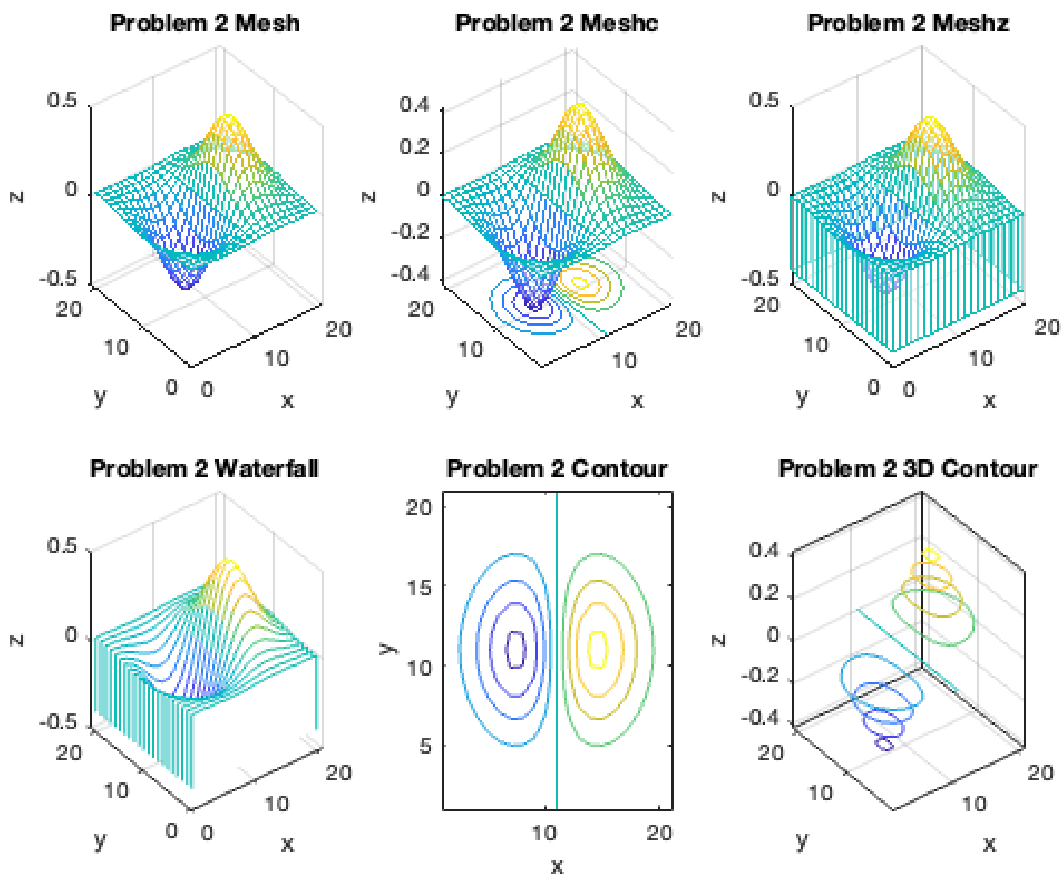
```
problem3Data = [[0, 300]; [1, 150]; [2, 75]; [3, 35]; [4, 12]; [5, 5]; [6,
 2]];

problem3Data(:, 3) = log(problem3Data(:, 1));
problem3Data(:, 4) = log(problem3Data(:, 2));
```

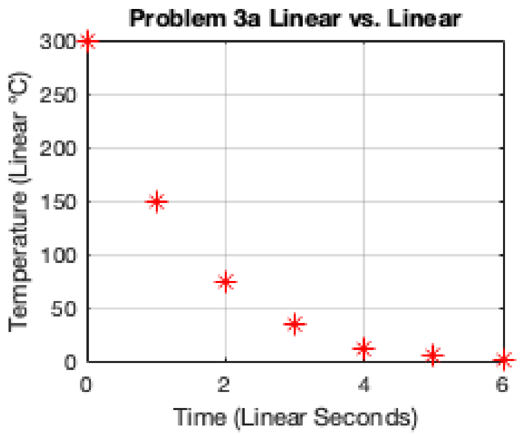
Problem 3a

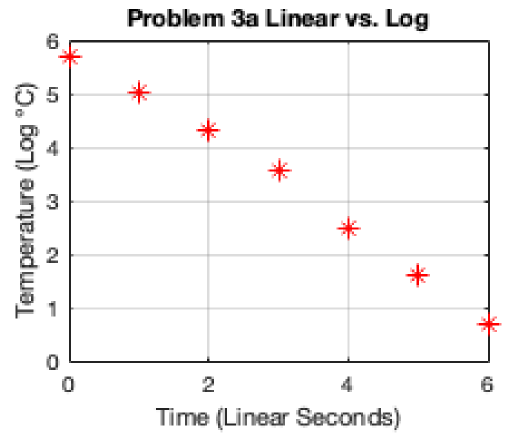
Find a functional description of these data. Do this by creating four (linear-linear, semilogx, semilogy, log-log) subplots and inspecting said subplots for the most “linear”.

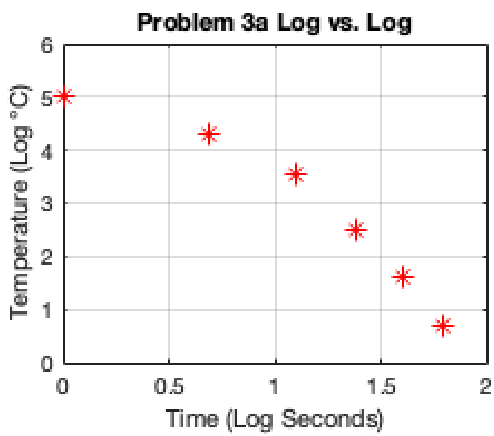
```
figure(30);

% Plotting Linear vs Linear
problem3aLinLin = subplot(2,2,1);
plot(problem3Data(:, 1),problem3Data(:, 2),'*r' ,'Parent', problem3aLinLin);
```

```
% Define Plot Characteristics
xlabel(problem3aLinLin, "Time (Linear Seconds)");
ylabel(problem3aLinLin, "Temperature (Linear °C)");
title(problem3aLinLin, "Problem 3a Linear vs. Linear");
grid(problem3aLinLin, 'on');


% Plotting Linear vs Log
problem3aLinLog = subplot(2,2,2);
plot(problem3Data(:, 1),problem3Data(:, 4),'*r' , 'Parent', problem3aLinLog);

% Define Plot Characteristics
xlabel(problem3aLinLog, "Time (Linear Seconds)");
ylabel(problem3aLinLog, "Temperature (Log °C)");
title(problem3aLinLog, "Problem 3a Linear vs. Log");
grid(problem3aLinLog, 'on');


% Plotting Log vs Log
problem3aLogLog = subplot('Position', [0.35,.1,0.34,0.34] );
plot(problem3Data(:, 3),problem3Data(:, 4),'*r' , 'Parent', problem3aLogLog);

% Define Plot Characteristics
xlabel(problem3aLogLog, "Time (Log Seconds)");
ylabel(problem3aLogLog, "Temperature (Log °C)");
title(problem3aLogLog, "Problem 3a Log vs. Log");
grid(problem3aLogLog, 'on');
```


Problem 2 Mesh
Problem 2 Meshc
Problem 2 Meshz
Problem 2 Waterfall
Problem 2 Contour
Problem 2 3D Contour
x
y
z





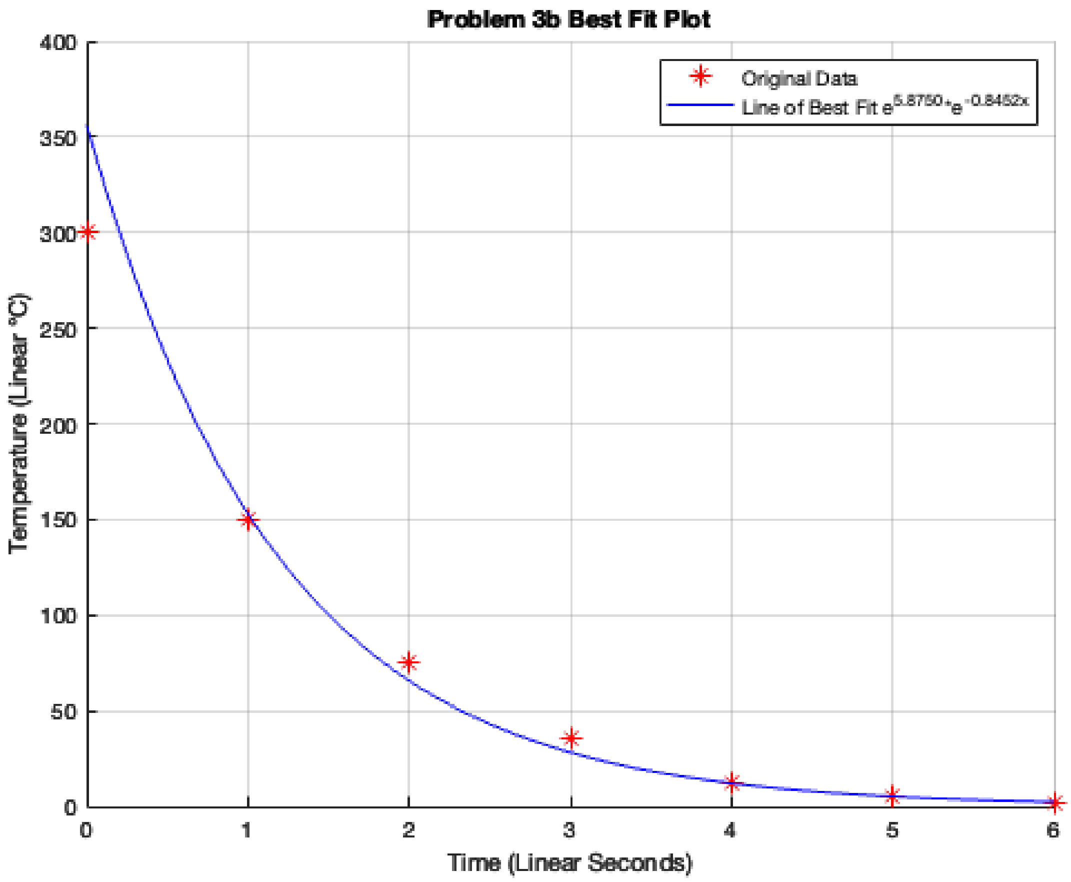


Problem 3b

Plot the function and the data on the same plot.

```
figure(31);

% Plotting most Linear vs. Log function as it has the most linear
 characteristic
problem3bFig = axes();
problem3bFit = polyfit(problem3Data(:, 1),problem3Data(:, 4),1);
hold(problem3bFig, 'on')

% Plotting Data and Function
plot(problem3Data(:, 1),problem3Data(:, 2),'*r' ,'Parent', problem3bFig);
fplot(@(x) exp(problem3bFit(2))*exp(problem3bFit(1)*x),[min(problem3Data(:,
 1)), max(problem3Data(:, 1))] ,'b' ,'Parent', problem3bFig);

% Define Plot Characteristics

xlabel(problem3bFig, "Time (Linear Seconds)");
ylabel(problem3bFig, "Temperature (Linear °C)");
title(problem3bFig, "Problem 3b Best Fit Plot");
```

```
legend(problem3bFig, "Original Data", "Line of Best Fit
 e^{5.8750}*e^{-0.8452x}");
grid(problem3bFig, 'on');
```



Problem 3c

Create a table of the given data using fprintf. Your output table should have two columns: one column with the “Time” heading and one with the “Temp” heading, including units, of course. Your table should be properly formatted for output in the published (pdf) file.

```
% Set up data for table
problem3cPrintData(1, :) = [problem3Data(:, 1)'];
problem3cPrintData(2, :) = [problem3Data(:, 4)'];

% Print Table
fprintf('%-9s %15s\r\n',' Time(s)',' Temperature(°C)');
fprintf('%9s %15s\r\n','---------','-----------------');
fprintf('% -09.2f % -15.2f\r\n', problem3cPrintData);
```

```
 Time(s)   Temperature(°C)
--------- -----------------
 0.00      5.70
 1.00      5.01
```

```
2.00      4.32
3.00      3.56
4.00      2.48
5.00      1.61
6.00      0.69
```

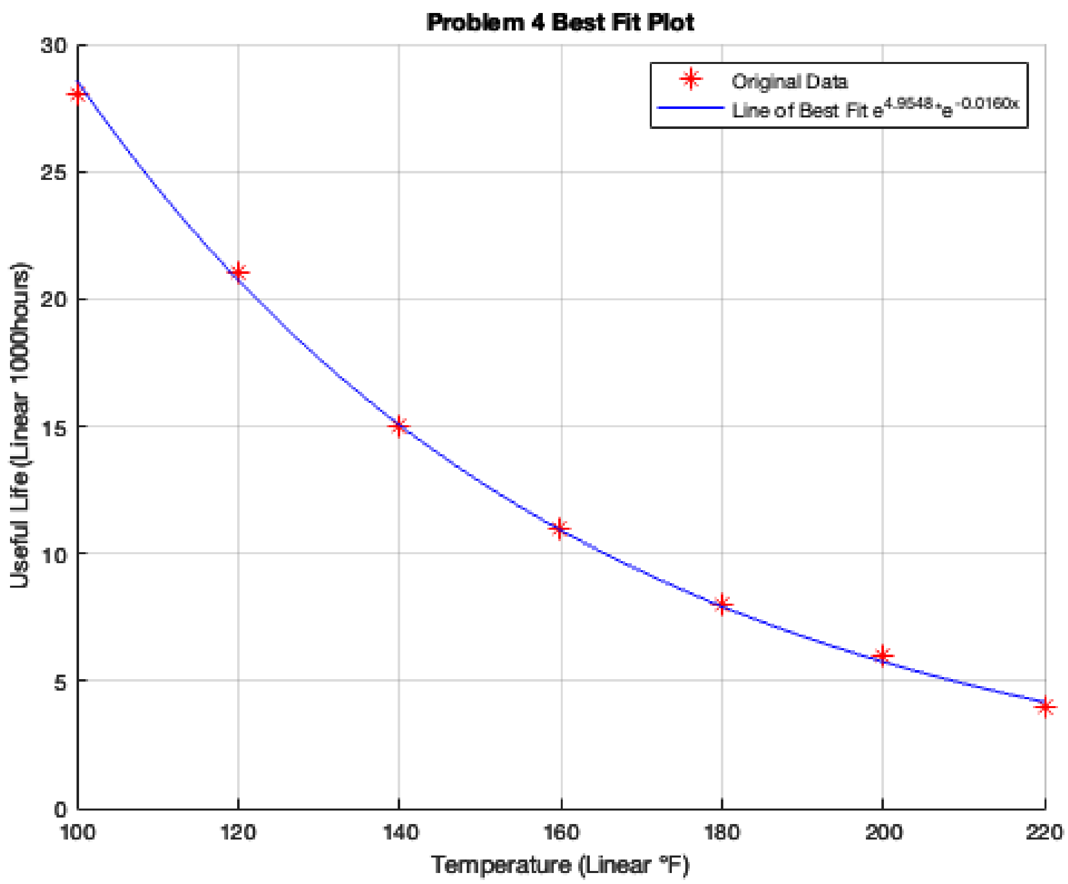
Problem 4

The useful life of a machine bearing depends on its operating temperature (see data). Obtain a functional description of these data (use the same process as in Problem 3). Plot the function and the data on the same plot (Temperature is the independent variable on the horizontal axis). Use your plot and function to estimate the bearing's life if it operates at 150°F.

```
% Set up problem 4 Data
problem4Data(:, 1) = (100:20:220);
problem4Data(:, 2) = [28, 21, 15, 11, 8, 6, 4];
problem4Data(:, 3) = log(problem4Data(:, 2));


figure(40);

% Set up axes for plotting
problem4fig = axes();
hold(problem4fig, 'on');
grid(problem4fig, 'on');
problem4Fit = polyfit(problem4Data(:, 1), problem4Data(:, 3), 1);
problem4Function = @(x) exp(problem4Fit(2))*exp(problem4Fit(1)*x);

% Plot data and best fit line
plot(problem4Data(:, 1), problem4Data(:, 2),'*r' ,'Parent', problem4fig);
fplot(problem4Function,[min(problem4Data(:, 1)), max(problem4Data(:, 1))] ,'b'
 ,'Parent', problem4fig);

% Define Plot Characteristics

xlabel(problem4fig, "Temperature (Linear °F)");
ylabel(problem4fig, "Useful Life (Linear 1000hours)");
title(problem4fig, "Problem 4 Best Fit Plot");
legend(problem4fig, "Original Data", "Line of Best Fit
 e^{4.9548}*e^{-0.0160x}");

% Create a table of the given data using fprintf. Your output table should
 have two columns: one column with the
% “Temperature” heading and one with the “Bearing Life” heading, including
 units, of course.  Include the estimated
% bearing life at 150°F in the table. Your table should be properly formatted
 for output in the published (pdf) file.

% create a matrix for fprintf data
problem4PrintData = zeros(2,8);

% Load matrix with data
problem4PrintData(1, :) = [problem4Data(1:3, 1)' 150 problem4Data(4:end, 1)'];
```

```
problem4PrintData(2, :) = exp(polyval(problem4Fit, problem4PrintData(1, :)));

% Print solution table
fprintf('%-9s %15s\r\n',' Time(s)',' Temperature(°F)');
fprintf('%9s %15s\r\n','---------','-----------------');
fprintf('% -09.2f % -15.2f\r\n', problem4PrintData);
```

```
 Time(s)   Temperature(°F)
--------- -----------------
 100.00    28.58
 120.00    20.74
 140.00    15.06
 150.00    12.83
 160.00    10.93
 180.00    7.93
 200.00    5.76
 220.00    4.18
```



Published with MATLAB® R2022b