Lab 6:
Graphical Methods

6.1 Introduction

Lab this week is going to introduce graphical solution and presentation techniques as well as surface plots.

6.2 Resources

The additional resources required for this assignment include:

- Books: Palm
- Pratt Pundit Pages: `MATLAB:Plotting`, `MATLAB:Plotting Surfaces`
- Lab Manual Appendices: None

6.3 Getting Started

1. Log into one of the PCs in the lab using your NET ID. Be sure it is set to log on to acpub.
2. Start a browser and point it to `http://pundit.pratt.duke.edu/wiki/Lab:B209`. Check out the `Using the PCs to Run MATLAB Remotely` section for how to get connected and make sure the connection is working.
3. Once connected to a machine you believe will also display graphics, switch into your `EGR53` directory and create a `lab6` directory inside it:

   ```
   cd EGR53
   mkdir lab6
   ```

4. Switch to your ~`/EGR53/lab6` directory:

   ```
   cd lab6
   ```

5. Copy all relevant files from Dr. G's public `lab6` directory:

   ```
   cp -i ~mrg/public/EGR53/lab6/* .
   ```

   Do not forget the space and the "." at the end.
6. Open MATLAB by typing `matlab &` at the prompt that appears in your terminal window. It will take MATLAB a few seconds to start up.

6.4 Assignment

6.4.1 Based on Palm Problem 5.1, p. 340

This problem, which comes from economics, is directed at solving the breakeven point for where a particular product will be profitable. It uses a highly simplified economic model. You will be solving the problem graphically. Note that the plot will have the sales quantity Q along the x axis and both the sum of the fixed and variable costs as well as the money earned from sales along the y axis. You will need to first generate an anonymous function for cost as a function of Q, then an anonymous function for money earned from sales as a function of Q. Make a plot using those anonymous functions that contains two curves - the costs for producing Q items and the sales earned from selling Q items. The intersection of these two curves will be the breakeven point. Your plot should start with a quantity of 0 and have a maximum value that is at least twice the breakeven point. You may need to experiment with the program a few times to make sure you are plotting far enough out.

Once you have the plot working, you can use a combination of `zoom` and/or `ginput` to find the approximate breakeven point. Once you have determined both the Q and dollar amount associated with the breakeven point, you will add code to your script such that it will automatically place a marker on it. To mark it, plot a square. For example, to plot the intersection of the lines $y = 2x$ and $y = x/3 + 2.5$, you could use the following:

```
x = linspace(0, 3, 100);
y1 = @(x) 2*x;
y2 = @(x) x/3+2.5;
plot(x, y1(x), 'k-', x, y2(x), 'k--');
```

then graphically find the approximate intersection point. Once found, you could then plot a square at that point on the same graph as follows:

```
xp = 1.5;
hold on
plot(xp, y1(xp), 'ks', 'MarkerSize', 20);
hold off
```

Note that the addition of the arguments `'MarkerSize'` and `20` at the end of the `plot` command tells MATLAB what size to make the individual points for that plot. Also, you could use either `y1` or `y2` for plotting the intersection point since, by definition, the two functions are the same there.

Be sure to label and title your graph. You should also use a legend to note the costs, sales, and breakeven point. In the body of your lab report, you will include the quantity required to make the breakeven point and the total cost / total revenue for that location. You must also calculate and report the amount of money lost if only half that amount is made and sold and the profit earned if twice the breakeven amount is made and sold. Those calculations should be in your script as well.

6.4.2 Based on Palm Problem 5.3, p. 341

There are three real roots. To find them, you should plot the polynomial and then zoom in to see the locations where the polynomial crosses the x axis. The way you will present these graphically is to plot the whole function on the left side of a figure, then put zoomed-in versions on the right side to show the actual roots. You can manually zoom to determine the axis limits to use for a particular subplot, then use the `axes` command to set those values. The following code may be helpful:

```
subplot(1,2,1)
% Plot entire function

subplot(3,2,2)
% Plot function, then set axis values to zoom in to first root
axis([ ?? ])
title('First root: ??');

subplot(3,2,4)
% Plot function, then set axis values to zoom in to second root
axis([ ?? ])
title('Second root: ??');

subplot(3,2,6)
% Plot function, then set axis values to zoom in to third root
axis([ ?? ])
title('Third root: ??');
```

You should make sure to add proper labels. Your NET ID only needs to go in the title of the main, left graph. You should indicate the three values for the roots that you have found. The x axis of your main plot should have a domain from -6 to 6 while your zoomed-in plots should have a maximum domain of 0.01, centered on the root, and a range from -0.01 to 0.01. The values you report for roots should have three digits after the decimal point. You may use the `grid on` command if that helps. Also note that because of the precision you are asked for, you will need to plot 10000 points - this is a case where you are using more than the usual number of points because you will be zooming in.

Note on Printing Contour / Surface / Mesh Plots

When using the `print` command for `mesh/meshc` plots, `surf/surfc` plots, and `contour` plots use the `-depsc` adverb instead of just `-deps` so that the figures will print in grayscale. That is to say, your MATLAB scripts should have lines similar to

```
print -depsc MyFileName
```

in them for several of the plots below. This is so that when the printer gets these files, it will assign different shades of gray to each color. You do *not* need to pay money to print this out on a color printer.

6.4.3 Palm Problem 5.56, p. 356

You will be making three figures for this problem - one where $n = [1]$, one where $n = [1, 3, 5]$, and one where $n = [1, 3, 5, ..., 99]$. The goals are to see the effect of using more terms in the solution, to get practice using `meshc` and `countour` plots, and to visualize the temperature distribution on a plate with one side held at a higher temperature than the other three. Each figure will be split into two subplots - put the `meshc` plot on the left and the `contour` plot on the right. **Note:** instead of using a spacing of 0.2, use a spacing of 0.1 to get a more refined plot. Be sure to properly label your axes and to title each graph. The title should include how many terms were used in the calculation and what kind of plot is being presented. In the body of the lab report, state what you believe is the effect of using more terms.

6.4.4 Palm Problem 5.61, p. 357

The plot for this should be similar to each of the plots from the problem above - except you will be using `surfc` instead of `meshc` for the left-hand plot. Be sure to include your code and your graph in the appendix. Use 20 points in each direction for your `meshgrid` and use 20 contour lines in your `contour` plot. In the body of the lab report, answer the questions given in the problem. Note that a saddle point is typically indicated by contour lines that intersect each other.

6.4.5 Based on Palm Problem 4.25, pp. 248-249

For this problem, you are going to generate a surface of the costs associated with placing a distribution center at a particular location and then use that to determine the best place to put the center. Each student in the class will have a different distribution of customers - both from each other and from the book. The `MakeCustomers` script you copied into your account for this assignment needs to be altered such that your NET ID is used to start the random number generator. The script will then create a file called `DataTable.dat` that contains the customer number, x-location, y-location, and sales volume for your customers. Note that the x and y locations for each customer will be between 1 and 29.

Your script will be generating three graphs. The first will show where each of your customers is. This will be similar to the graph on page 249 of the book *except* that instead of labeled dots you will be putting the customer numbers only on the graph. The second will be a mesh plot - with contours - of the cost surface. The third will be a shaded surface plot - as viewed from above - of the cost surface.

To start, you may first want to write code that loads the customer information from your data file and splits it into intelligently-named matrices. At this point, you should also determine how many customers you have. Do *not* hard-code this into your program - when we grade your program, we may use a different data set entirely.

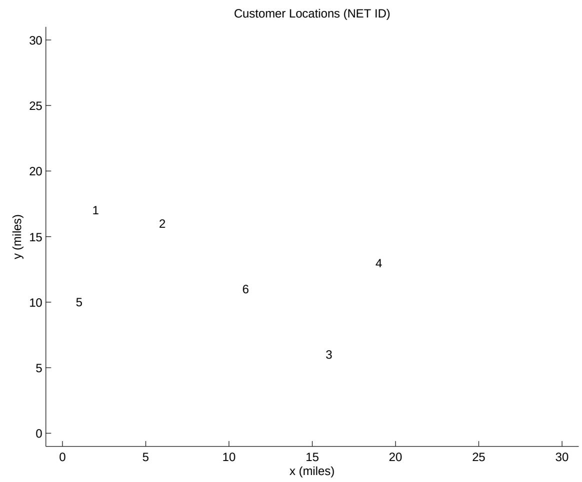
Next, write code that generates a grid of x and y locations for each mile - for this problem, that is the resolution you will be using both graphically and numerically. After that, generate an equation for the distance between each one of those 961 points and the first customer. If you make a plot of that equation, you should note a minimum at the coordinates of your first customer. To create a cost function for the first customer, take the distances you just generated and apply the cost function given in the book to them. This graph now represents the cost function for the first customer, and if you were to find the minimum, you (should) find that the best place to put the distribution center would be on top of that customer.

However, there are a total of between six and eight customers to think about, so you will need to combine the cost functions for all of them before determining the optimal location of the distribution center. Generate a matrix for the *total cost* at each location by successively adding cost functions for each customer. Then, use the `min` and `find` commands to your advantage to determine the minimum operating cost as well as the best location for the distribution center. You will include this information in your lab report as well as the three graphs.

For the first graph, which puts the customer number at an appropriate location in a figure window, you will want to first start with a clean figure:

```
figure(1); clf
```

then see the Pundit page on Plotting - specifically about Putting Text on a Plot - for how to create the customer map. Here is an example:



You may want to use the `axis` command to make sure your x and y axes span the entire domain and range from 0 to 30.
The second graph should be in its own figure window, so start with:

```
figure(2); clf
```

and then make a mesh plot with contours of the total cost function. Finally, for the third graph, start with

```
figure(3); clf
```

and create a surface plot, using interpolated shading, as viewed from above. Some code for the third plot is given below, assuming `x` is your x variable, `y` is your y variable, and `TotalCost` is the matrix containing the total cost:

```
surf(x, y, TotalCost);
shading interp % change shading and remove grid lines
view(2)        % change camera location to straight-above
colorbar       % add a colorbar to interpret shadings
```

In the body of the lab report, you should present a data table similar to that on p. 249 but using the data for your customers. You should also present the best (lowest-cost) location for the distribution center and the coordinates. There is an exceedingly small, but still non-zero, chance that more than one location will have the same cost - if your data set presents you with multiple best-options, indicate them all.

Note that you do *not* need to include the `MakeCustomers` program file in your lab report. Be sure, however, that you used *your* NET ID to generate your data file.