Homework 4

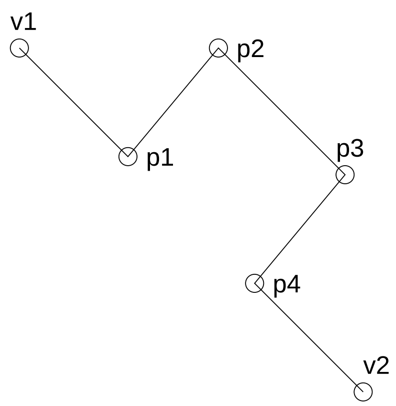
cs249
Fall 2001

Allen B. Downey
Computer Science Department

Due:
The reading for this assignment is

4.1 Quadric Koch curve

The generator for the quadric Koch curve is



where all the angles are right and all the line segments are the same length.

1. Download a copy of `snow.m` from

   `http://rocky.wellesley.edu/cs249/code/snow.m`

   Install it in your script folder and run it. Read the code and make sure you understand how it works (with the exception of `draw_line`; we don't understand that yet).

2. In the top-level function, set the value of `n` to 0 and run the program again. Set it to the other values 1...4 and run the program again. You should understand the relationship between the number of levels of recursion and the figure that is generated.

   If you run the program with more levels of recursion, it takes longer but doesn't get much more interesting.

3. Before making any other changes to the program, make a new copy of it called `koch`. Rename the top-level function `koch` and replace the word "snow" everywhere it appears with the word "koch". Change the title command so that it says, "Quadric Koch curve by *Your Name*".

4. In the top-level function, modify the program so that it draws a square rather than a triangle. Test the program again.

5. In the helper function `draw_koch` (formerly `draw_snow`), change the definitions of `p3` and `p4` as shown in the figure, and change the recursive calls to draw the 5 line segments in the figure.

6. Set the recursion level to 1 and test the program. If it works, increase the recursion level one at a time, and see what the figure looks like. If the program doesn't work, use the tips below to help you debug it.

7. When the program works to your satisfaction, add comments to the code in the style we have discusses and print a copy. Also print a copy of the figure. Make sure your name is on both.

Here are some debugging tips that might make life easier:

- Start with a working program. Make small changes at a time, and then run the program again. If there is an error, you have a pretty good idea where it is!
- If the program runs into an error, reduce the level of recursion to the simplest level that causes the error. This will simplify the debugging effort.
- If you know what variable is causing the error, try displaying the value of that variable when it gets assigned, or when it gets passed as an argument.
- Inside a function, it is often useful to add a line right at the beginning that displays the values of the arguments.
- Work methodically. Don't make random changes to the program. Don't flail. Think!

4.2 Your own generator

The curve in the figure above is called a generator. Practically any similar generator yields a Koch curve that is visually interesting.

- Draw a sketch of a generator curve that you think is interesting (and that you can implement in MATLAB). For your own entertainment, try to figure out on paper what the Koch curve will look like for various levels of recursion.
- Make a copy of `koch` called `koch2` and modify it to use your generator instead of the quadric generator. You may also want to change the shape of the top-level figure (the thing that we changed from a triangle to a square).
- Type `help subplot` and read the documentation of the `subplot` command.
- Split the top-level function into two functions. The new top-level function, `koch2` generates a figure that is divided into four subplots, and then invokes `koch_helper` to fill in the subplots.

```
function koch2 ()
    clf; axis equal
    subplot(2,2,1)
    koch_helper (1)

    subplot (2,2,2)
```

```
    koch_helper (2)

    subplot (2,2,3)
    koch_helper (3)

    subplot (2,2,4)
    koch_helper (4)

function koch_helper (n)
% draw a Koch curve with n levels of recursion
```

- Again, add comments to your code and print a copy. Draw your generator curve on this paper (and label the points in accordance with the code). Print a copy of the figure, showing the result of your generator curve at all four levels of recursion.