In-class test

There will be an in-class test next week (TBD), 4pm Wed 22nd Feb.

- It is a test on MATLAB (not scientific computing)
- Open book: use anything on Blackboard or the MATLAB help.
- There will be questions on
  - Defining and using anonymous functions;
  - Writing scripts in function files.
  - Flow-of-control (`if`, `for`, `while`...)
- Input and (especially) output.
- Working with vectors and matrices.
- Graphics.

Include your answers in a single script file, except, if needed, for any file function. Each question would be in a separate "section" of the script, meaning that there is a line between the relevant sections of code that start with a double percent. (See solution to sample questions, or ask during the test).

Q1. Define an anonymous function to represent $f(x) = \sin(x^2)$. Create a plot of it that resembles, as close as possible, the following , the image in 1.

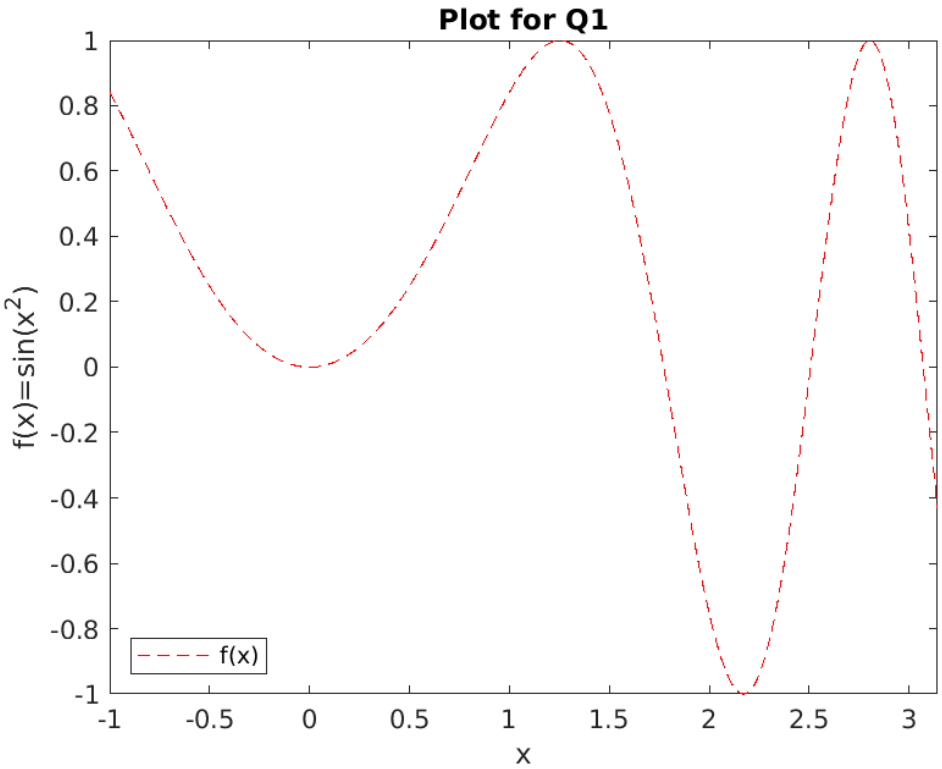

Figure: Figure for Q1

.

Q2. Write a (file) function called *make_normal()* that takes a single vector, x, as its input, and returns a vector y that is a scalar multiple of x but has the property that all its entries sum to 1. For example, if

$$x = \begin{bmatrix} 1 \\ 5 \\ 0 \\ -2 \\ 6 \end{bmatrix}, \quad \text{then it should return} \quad y = \begin{bmatrix} 0.1 \\ 0.5 \\ 0 \\ -0.2 \\ 0.6 \end{bmatrix}.$$

Your solution script file should demonstrate that your function works by applying it to the vector x above.

Q3. Find out what the `diff()` function does when applied to a vector. Write a script file that creates a **row** vector, x, such that `diff(x)` returns the vector $[0.1,\ 0.02,\ -3,\ 40,\ -500]$.

Q4. Write script that prompts the user to enter a number m. Then it sets n to be the maximum of 5 and $\lfloor m \rfloor$. Next `for`-loops are used to make an $n \times n$ matrix with the property that

$$a_{i,j} = \begin{cases} i - j & n \text{ is even} \\ i + 2j & n \text{ is odd} \end{cases}$$

Q5. Show how to use the zeros function and the colon operator to make the matrix using only 3 lines of MATLAB (and no `for` loops or `if` statements).

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Q6. Consider the following lines of MATLAB code:

```
x = 0.1:0.01:1;
y = exp( -1./x);
for i=1:10:length(x)
    fprintf('i=%d, x=%f, y=%f\n', i, x(i), y(i));
end
```

Modify the `fprintf` line so that the output resembles, as closely as possible, the following:

```
i=  1, x= 0.1, y= 4.540e-05
i= 11, x= 0.2, y= 6.738e-03
i= 21, x= 0.3, y= 3.567e-02
i= 31, x= 0.4, y= 8.208e-02
i= 41, x= 0.5, y= 1.353e-01
i= 51, x= 0.6, y= 1.889e-01
i= 61, x= 0.7, y= 2.397e-01
i= 71, x= 0.8, y= 2.865e-01
i= 81, x= 0.9, y= 3.292e-01
i= 91, x= 1.0, y= 3.679e-01
```