Fall 2016 Math 541

Homework – MatLab Programming Due Tues. 9/20/16

1. (**Each part 1 pt**) If I give you the array

```
X = linspace(0,1,5);
```

   (a) How many points are in the array?

   (b) What is the spacing between the points?

   (c) What code would you write to double the number of points in the array?

   (d) What code would you write to have equally spaced points in the interval $[0, 2]$, but with the same spacing between points as the original array?

   (e) What code would I write to turn `X` into a column vector?

   (f) What array would the code `X.^2` produce?

   (g) What array would the code `X(end:-1:1)` produce?

   (h) Using vectorization, what code would I write to efficiently plot 2003 equally spaced points of the function $\sin(x^3 + 2x)$ over the interval $[-3.7, 4.2]$ in the color blue with a linewidth of 2? Note, your answer should be two lines. One to define an array of points, say `X`, and one to make the plot.

2. (**3 pts**) Using a for loop based approach, write a program which finds

$$\sum_{j=1}^{n} (j^3 + 4j^2),$$

for any n.

The beginning of your program should look something like

```
function tot = sumfun(nstop)

tot = 0;
for jj=1:nstop
     tot = tot + ;% you fill in the rest...
end
```

Give the answers for $n = 10$, 43, and 72.

3. (**3 pts**) Extend the program for generating Fibonacci numbers to satisfy the recursion relationship:

$$p_n = ap_{n-1} + bp_{n-2}, \; p_0 = s_0, \; p_1 = s_1.$$

You program should take as input the values a, b, s_0, s_1, and n, and it should return p_n. Thus, you would want to start your program with something like

```
function pn = gen_fib(a,b,s0,s1,n)

p0 = s0;
p1 = s1;

for jj = 2:n
    % You fill in the rest
end
```

For $a = 3.2$, $b = -2$, $s_0 = 3$, $s_1 = 0$, what is p_{10}? What is p_{50}?

4. (**4 pts**) Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

   1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

   By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms. Note, the use of the Matlab command `mod` is going to be critical.

5. (**4 pts**) If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6, and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000. Note, aside from using a for loop, you need to make use of a logic structure like

```
if(mod() || mod()) % You fill in the blanks here.

end
```

6. (**3 pts**) Using Matlab and the Maclaurin series for $\sin(x)$, write a code, which computes $\sin(1)$. Explain how you choose a stopping criteria, and determine the maximum accuracy you are able to achieve.

7. (**5 pts**) Create a Matlab function of the Maclaurin series for $\cos(x)$, which depends on `x` and a tolerance, `tol`. Explain how you choose a stopping criteria, and determine the maximum accuracy you are able to achieve. Create another MatLab function, which plots the function for $-L_x \leq x \leq L_x$ where L_x is a user specified input. Overlay a plot of the MatLab defined function `cos(x)`, using dashed lines.

8. (**10 pts**) An important differential equation in mathematical physics is Airy's equation which is given by

$$y'' - xy = 0.$$

   Two solutions to this equation can be found via the power series solutions

$$y_1(x) = 1 + \sum_{m=1}^{\infty} \frac{x^{3m}}{(2\cdot 3)(5\cdot 6)\cdots((3m-1)\cdot 3m)}$$

   and

$$y_2(x) = x + \sum_{m=1}^{\infty} \frac{x^{3m+1}}{(3\cdot 4)(6\cdot 7)\cdots(3m\cdot(3m+1))}$$

Write a while based code which ultimately plots the function for $-L_x \leq x \leq 0$ and $0 \leq x \leq L_x$ where L_x is a user specified input. So, first, you would write a code which found the solutions to Airy's equation, such as

```
function [y1x,y2x] = airy_maker(x,tol)
% You fill in the rest
```

Then, you would want a code which called your `airy_comp` which might look something like

```
function airy_plotter(Lx,Npts)
  tol = ; % You choose a tolerance
  posvals = linspace(0,Lx,Npts);
  negvals = linspace(-Lx,0,Npts);
  % You fill in the rest
```

Explain how you choose a stopping criteria and the accuracy you are able to achieve with your program. How large can you make L_x before your series solutions are unreliable? Provide plots that justify your answer. Describe the difference between the behavior of the solutions for $x < 0$ and $x > 0$. (Hint: Refer to the class notes on Bessel functions to help design these programs.)