CS 257: Numerical Methods
Spring 2006

Homework, Set 2 **Due Thursday February 2, 2006**

- **(-)** Start EACH PROBLEM on a SEPARATE piece of paper (This is important since we may assign each problem to a different grader).
- **(-)** Put your NETID and HW NUMBER on top of EACH PAGE clearly, e.g. "netid: zamani hw2".
- **(-)** Write descriptive solutions. Comment your code!
- **(-)** Include your curves/graphs (and other supporting materials) in your write-up.
- **(-)** Don't use handwritten code (unless you want to lose points), copy-paste your code into your write-up or attach a proper print of code to your papers.
- **(-)** Please write everything in a "portrait" style (not landscape).
- **(-)** Please number problems according to numbers presented in the homework write-up that appears on the course page, NOT according to the numbers in the textbook.
- **(-)** Please type your homework or hand-write it legibly (but yet attach a print of your codes to your handwritten stuff).

(1) (NMM Ch. 3 #12) Write two function m-files, `FtoC` and `CtoF`, to convert temperatures from degrees Farenheit to degrees Celsius and from degrees Celsius to degrees Farenheit. Each function should have one input parameter and one output parameter. Test your functions with

```
>> FtoC( CtoF(100) )
>> CtoF( FtoC(32) )
>> FtoC(0:10:100)
>> CtoF(0:10:100)
```

along with any other tests you think is appropriate.

(2) (NMM Ch. 3 #15) Write a function `tridiag(c,d,n)` where n is the dimension of a matrix and allow c and d to be vectors or scalars. The function should create the matrix

$$D = \begin{bmatrix} c_1 & d_1 & 0 & 0 & \cdots & 0 \\ d_1 & c_2 & d_2 & 0 & \cdots & 0 \\ 0 & d_2 & c_3 & d_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & d_{n-1} \\ 0 & 0 & 0 & \cdots & d_{n-1} & c_n \end{bmatrix}$$

if c and d are vectors, or create the matrix

$$D = \begin{bmatrix} c & d & 0 & 0 & \cdots & 0 \\ d & c & d & 0 & \cdots & 0 \\ 0 & d & c & d & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & d \\ 0 & 0 & 0 & \cdots & d & c \end{bmatrix}$$

if c and d are scalars. Note that D is symmetric, i.e., $D = D^T$. (*Hint:* Check the size of c and d to see if these input values are vectors or scalars.)

(3) (NMM Ch. 3 #17) Write the MATLAB statements that use a loop and the fprintf function to produce the following output:

(a)

```
1^2 = 1
2^2 = 4
3^2 = 9
4^2 = 16
5^2 = 25
```

(b)

```
(0.10)^2 = 0.0100
(0.20)^2 = 0.0400
(0.30)^2 = 0.0900
(0.40)^2 = 0.1600
(0.50)^2 = 0.2500
```

(4) (NMM Ch. 3 #18 - #20)

(I) Write a horner function that uses Horner's rule to evaluate a polynomial of arbitrary degree. (Cf. Example 3.7 on page 113). Use the function definition

```
function p = horner(b,x)
```

where b is the vector of coefficients that define the polynomial. The return value p is the value of the polynomial at x. Test your function with $b =$ [1 2 -1 0] at $x = 1, 2, 3, 4, 5$.

(II) Extend the above function so that it returns a vector of polynomial values if the input x is a vector.

(III) Extend the above function so that it can be called in two ways, viz.,

```
p = horner(b,x)
```

and

```
[p, pp] = horner(b,x)
```

where p is the value of the polynomial at x and pp (p prime, $p' = dp/dx$) is the value of the first derivative of $p(x)$ evaluated at x. Make sure that your function returns p and pp as vectors if x is a vector.

(5) (NMM Ch. 3 #35) Given the x vector

```
>> x = [21 22 23 24];
```

and B matrix

```
>> B = ones(3,3);
```

write the one-line vectorized copy operation that has the same effect as the following scalar loop:

```
k = 0;
for i=2:3
  for j=1:2
    k = k + 1;
    B(i,j) = x(k);
  end
end
```

(*Hint:* The built-in `reshape` function will be helpful.)

(6) (NMM Ch. 3 #36) The snippet of code on page 121 at the end of §3.5.2 computes

$$s_j = \begin{cases} \sqrt{y_j}, & \text{if } y_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

Write a vectorized (no loop) version of this code. (*Hint:* Preallocate s and use the `find` function.)

(7) (NMM Ch. 4 #3) Semiconductor devices are mass produced on wafers of Silicon that can hold more than a hundred devices each. As a wafer moves through production, random failures can occur during any one of many stages of processing. When the wafer is completely processed, each device on the wafer is tested, and the wafer is cut into individual devices, or *chips*. Chips passing the final test are attached to a lead frame and encapsulated in a package that is eventually assembled into a circuit board. The ratio of successful chips to the total number of chips is called the yield of the process. The `chip.dat` file contains data for a production run of a semiconductor device. The first column is the serial number of the chip, and the second column is the chip's speed limit in megahertz (MHZ). The speed limit is the highest clock speed at which a chip can reliably run. Entries in the second column equal to `NaN` indicate that the chip completely failed the speed test.

Use stepwise refinement to create a program (that may consist of multiple m-file functions) to determine the gross yield and the number of chips in the data set that qualified at 250, 300, 350, 400, and 450 MHZ.

Get `chip.dat` from here: http://www.cs.uiuc.edu/class/sp06/cs257/homework/chip.dat