Homework 1
MA/CS 375, Fall 2005
Due September 16

This homework will count as part of your grade so you must work independently. It is permissible to discuss it with your instructor, the TA, fellow students, and friends. However, the programs/scripts and report must be done only by the student doing the project. Please follow the guidelines in the syllabus when preparing your solutions.

1. How many numbers belong to the set $\mathbf{F}(2, 2, -1, 4)$? What is the value of ϵ_M for such set?

2. Use MATLAB to construct an upper (respectively lower) triangular matrix of dimension 8 with -4 on the diagonal and 1 on the upper (respectively lower) diagonal. Interchange the second and fifth rows and then the second and sixth columns. Compute the determinants of each of the matrices you have computed. Do you notice anything?

3. Check the linear independence of the following vectors in $\mathbb{R}^5$:

$$v_1 = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} -2 \\ 2 \\ 3 \\ 3 \\ 0 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad v_4 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 0 \end{pmatrix}, \quad v_5 = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}.$$

4. The Stirling numbers of the second kind are defined by the formula:

$$S_n^{(m)} = \frac{1}{m!} \sum_{k=0}^{m} (-1)^{(m-k)} \begin{pmatrix} m \\ k \end{pmatrix} k^n$$

.

   (a) Write a matlab function "stirling2.m" that accepts the arguments n and m and returns $S_n^{(m)}$. The function ought to have the form:

function S = stirling2(n,m)

(b) Use your function to compute $S_n^{(m)}$ for $m = 5, 10, 15$ and $n = 5, 10, 15, 20, 25$. The answers must be as accurate as possible. You must write a driver script that calls the function and prints your answers (with as many digits as feasible–use *format long g*) in a table of the form:

n \ m	5	10	15
5			
10			
15			
20			
25			

(NOTE: to test your script, use the fact that:
S(n=10, m=2) = 511, S(n=15,m=8) = 216627840), while for $n < m$, S = 0 and for $n = m$, S=1).

5. Consider the following algorithm to compute the volume of the unit sphere. Compute n random triples (x_k, y_k, z_k) with each coordinate uniformly distributed in the interval $[0, 1]$. Compute the number of these, p, lying inside the unit sphere. For n large we expect:

$$\frac{p}{n} \approx \frac{\frac{\text{Volume of Sphere}}{8}}{\text{Volume of Unit Cube}} = \frac{\text{Volume of Sphere}}{8},$$

since the points are constrained to lie in the octant $0 \leq x, y, z \leq 1$ and $1/8$ of the unit sphere lies in that octant. Carry out the computation for increasing n until the results apparently converge to two decimal digits. Using the fact that the volume is $\frac{4}{3}\pi$, use your results to approximate π. How accurate is your approximation?

6. The roots of the quadratic

$$ax^2 + bx + c = 0$$

can be found by the well known formula

$$x_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

We can also approximate the roots using fixed-point iterations, as follows:

(a) divide through by x to get the equivalent equation (assuming $x \neq 0$):

$$ax + b + \frac{c}{x} = 0$$

(b) Method 1, forward iteration: consider the sequence

$$\begin{aligned} x_1 &= x_{init} \\ x_{n+1} &= -\frac{b}{a} - \frac{c}{a}\frac{1}{x_n}\ ,\ n = 1, 2, 3, \ldots \end{aligned}$$

(c) Method 2, backward iteration: consider the sequence

$$\begin{aligned} x_1 &= x_{init} \\ x_n &= -\frac{b}{a} - \frac{c}{a}\frac{1}{x_{n+1}} \\ \text{i.e.} \quad x_{n+1} &= -\frac{c}{b + a x_n}\ ,\ n = 1, 2, 3, \ldots \end{aligned}$$

Write matlab functions forward.m and backward.m implementing the above iterations, that accept as inputs x_{init}, a, b, c, tol and produce an answer x^* that differs from one of the roots by no more than tol. (You can accomplish that, e.g., by terminating the iteration if $|x_{n+1} - x_n| \leq \text{tol}/2$). Demonstrate the performance of your function for the values:

$$x_{init} = 1\ ,\ a = 1\ ,\ b = c = -1\ ,\ \text{tol} = 10^{-10}$$

How many iterations are required to reach the desired accuracy? Note that the two functions will reach different roots, $x^*_{forward}$ and $x^*_{backward}$. Verify directly that these are within the desired tolerance from one of the roots of the quadratic. Which root is found by each method? Does the answer change if you vary x_{init}?

Now, use $x^*_{forward}$ as an initial value for backward.m with $\text{tol} = 10^{-10}$. What value do you end up with? Does the answer change if you also run backward.m again with $x_{init} = x^*_{forward}$ as before but with $\text{tol} = 10^{-13}$. Why?