CS 257: Numerical Methods
Spring 2006

Homework, Set 3 **Due Thursday February 9, 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. 5 #3) Convert the following numbers (by hand) to normalized floating point values with eight bit mantissas: 0.4, 0.5, 1.5.

(2) (NMM Ch. 5 #4) Translate algorithm 5.1 on page 198 (below) into a MATLAB function that computes the bit pattern of a floating point mantissa. Store the bit pattern in a string vector. The remainder r can be stored in a scalar that is overwritten for each k. Why does the built-in `dec2bin` function not work for this algorithm? Test your function by converting the values used in the preceding exercise.

$r_0 = x$
for $k = 1, 2, \cdots, m$ **do**
  if $r_{k-1} \geq 2^{-k}$ **then**
    $b_k = 1$
    $r_k = r_{k-1} - 2^{-k}$
  else
    $b_k = 0$
  end if
end for

(3) (NMM Ch. 5 #8) Manually evaluate the two roots from Example 5.3 using five-digit arithmetic in the classic quadratic formula. Repeat the calculations using six-digit arithmetic. What are the relative errors in the two roots for each case?

Example 5.3: $x^2 - 54.32x + 0.1 = 0$

(4) (NMM Ch. 5 #11, #13) Begining with the MATLAB code on page 210 (below), write a function m-file to compute ϵ_m. The code on page 210 will give an ϵ_m value that is slightly different from the built-in variable `eps`. Make sure your function returns a value exactly equal to `eps`. What modification is necessary? Explain. Now modify this m-file so that it takes a single input argument x_0. Use x_0 as the reference value for computing a relative ϵ_m. In other words, find $\hat{\epsilon}_m$ such that $x_0 + \delta = x_0$ whenever $\delta < \hat{\epsilon}_m$. Evaluate $\hat{\epsilon}_m$ for the two sequences of x_0 values $x_0 = \{1, 10, 100, 1000\}$ and $x_0 = \{1, 2, 4, 8\}$. Explain why the value of $\hat{\epsilon}_m$ increases in different proportions for the two series of x_0 values.

```
epsilon = 1;
it = 0;
maxit = 100;
while it < maxit
  epsilon = epsilon/2;
  b = 1 + epsilon;
  if b == 1
     break;
  end
  it = it + 1;
end
```

(5) (NMM Ch. 5 #14) Let $c(x) = 1 - x^2/2! + x^4/4!$ and $s(x) = x - x^3/3! + x^5/5!$ be approximations to $\cos(x)$ and $\sin(x)$, respectively. What are relative and absolue errors in using these formulas for $x = -2, -1, 0, 1, 2$ degrees?

(6) (NMM Ch. 5 #21) Use the sinser function in Listing 5.5 (below) and a fixed value of the convergence tolerance, tol = 5e-9, to explore the sensitivity of the convergence rate to the value of x in $\sin(x)$. In particular, make a plot of the number of terms to reach convergence n, versus x for 10 points in the interval $0 \leq x \leq 8\pi$. You will need to modify the sinser function so that it returns a value of n. you will also need to specify a minimum of 50 terms in the series to see a trend in n versus x.

```
function ssum = sinser(x,tol,n)
% sinser  Evaluate the series representation of the sine function
%
% Synopsis: ssum = sinser(x)
%           ssum = sinser(x,tol)
%           ssum = sinser(x,tol,n)
%
% Input:    x   = argument of the sine function, i.e., compute sin(x)
%           tol = (optional) tolerance on accumulated sum. Default:  tol = 5e-9
%                 Series is terminated when abs(T_k/S_k) < delta.   T_k is the
%                 kth term and S_k is the sum after the kth term is added.
%           n   = (optional) maximum number of terms. Default: n = 15
%
% Output:   ssum = value of series sum after nterms or tolerance is met

if nargin < 2,  tol = 5e-9;  end
if nargin < 3,  n = 15;      end

term = x;  ssum = term;         %  Initialize series
fprintf('Series approximation to sin(%f)\n\n  k      term         ssum\n',x);
fprintf('%3d  %11.3e  %12.8f\n',1,term,ssum);

for k=3:2:(2*n-1)
  term = -term * x*x/(k*(k-1));                    %  Next term in the series
  ssum = ssum + term;
  fprintf('%3d  %11.3e  %12.8f\n',k,term,ssum);
  if abs(term/ssum)<tol, break;  end               %  True at convergence
end
fprintf('\nTruncation error after %d terms is %g\n\n',(k+1)/2,abs(ssum-sin(x)));
```

(7) (NMM Ch. 5 #22) Modify the above sinser function so that the termination criterion is

$$\frac{T_k}{T_{k-2}} < \delta$$

where T_k is the current term being added to the sum S_k. Is there a significant difference between this convergence criterion and the original criterion? Compare the number of terms necessary for convergence when the same value of δ is used for your function and the original sinser. Which convergence criterion do you recommend?

(8)

- Log into the CS257 project Wiki (http://walleye.cs.uiuc.edu/wiki/).
- Make a comment (any comment) on your User page.

  (For example, see http://walleye.cs.uiuc.edu/wiki/index.php/User:Luke). Any comment will suffice.