University of Toronto Mississauga

CSC 338 - Numerical Methods, Spring 2010

Assignment 1

Due date: Thursday Jan 28, 4:10pm, at the start of tutorial.
No late assignments will be accepted.

Note: The material you hand in should be legible (either typed or *neatly* hand-written), well-organized and easy to mark, including the use of good English. All computer programs should be handed in and should be well commented. In general, short, simple answers are worth more than long, complicated ones. Unless stated otherwise, all answers should be justified.

1. (47 points total) The following exercises from Chapter 1 in the text are to be done by hand (unless specified otherwise).

   (a) (6 points) Question **1.42** on page 41. (1 point for each part)

   (b) (6 points) Question **1.43** on page 41. (2 points for each part)

   (c) (4 points) Question **1.51** on page 42.

   (d) (6 points) Question **1.2** on page 42. (2 points for each part)

   (e) (8 points) Question **1.4** on page 42. (Recall that $sin(x) \approx x$ when $x \approx 0$.) Repeat the problem using the function $f(x) = x^2 - 1$ instead of $\sin(x)$.

   (f) (12 points) Question **1.9** on page 43. (2 points for each part)

   (g) (5 points) Question **1.10** on page 43. (2 points for part a, and 3 for part b)

2. (26 points total) The following computer problems are to be done using Matlab or Octave. For each question, hand in your program code, any requested output, and a transcript of a terminal session demonstrating that your programs work correctly. Be sure to indicate clearly which questions the programs and the transcripts refer to.

   (a) (8 points total) Compute the value of the expression $(2 + 2^{-n}) - 2$ for $n = 5$, 25, 75 and 250. For which n is the value of the expression error-free (2 points)? Find the largest value of n for which the value of the expression is error-free (2 points). What can you say about ϵ_{mach} on your computer (3 points)? Based on Table 1.1 on page 18 in the text, which standard floating-point system does your computer use (1 point)?

   (b) (10 points total) Consider the expression $(2^n + 10) - ((2^n + 5) + 5)$. Find all positive numbers n for which the value of this expression is non-zero. Using the rounding rules described in Section 1.3.4 of Heath, explain why the expression takes on these particular non-zero values (6 points). Why is it zero for all other values of n (4 points)? You may assume that Matlab uses round to nearest in base 2.

   (c) (8 points total) For $n = 10$ and 20, add the squares of all the numbers from 1 to 2^n. Do this in two ways: in ascending order, and in descending order. When are the two sums the same (2 points)? Find the largest value of n for which the two sums are the same (1 point). Can you explain why this particular value of n is the largest (5 points)? (Hint: $\sum_{m=1}^{N} m^2 \approx N^3/3$.)

3. (30 points total) In this question, you will implement the function $\exp(x)$ using the formula $\sum_{n=0}^{\infty} x^n/n!$. More specifically, you will use the following two algorithms:

```
Algorithm 1:
    y = 0
    for n from 0 to N
        tn = x^n/n!
        y = y+tn

Algorithm 2:
    tn = 1
    y = 1
    for n from 1 to N
        tn = tn*x/n
        y = y+tn
```

In both algorithms, the final value of `y` is an estimate of `exp(x)`. In the questions below, you will be implementing these algorithms in Matlab (or Octave). You should use the Matlab function `plot` to plot points, and the Matlab function `factorial` to compute `n!` in Algorithm 1.

(a) (2 points) Prove that Algorithm 1 is equivalent to Algorithm 2 (assuming infinte-precision arithmetic).

(b) (3 points) How many multiplications and divisions does each algorithm perform? (Hint: $\sum_{n=1}^{N} n = N(N+1)/2$.)

(c) (10 points) Using Algorithm 1 with `N=1000`, plot $\exp(x)$ v.s. x for 500 values of x uniformly spaced from -1 to 1. Repeat using Algorithm 2. (5 points) Describe any differences in the behaviour of the two algorithms. Explain these differences. (5 points) (If either algorithm takes more than five minutes to execute, try plotting only 50 points, instead of 500. If both algorithms take less than five seconds to execute, try plotting 5000 points.)

(d) (5 points) Repeat the previous question four times using values of x in the following four ranges, respectively: 0 to 4, 8 to 12, -4 to 0, and -12 to -8. Describe and explain any differences in the behaviour of the two algorithms.

(e) (5 points) Using only Algorithm 2, plot $\exp(x)$ v.s. x for each of the following six ranges of x: -20 to -16, -40 to -36, -100 to -96, 16 to 20, 36 to 40, 96 to 100. Describe and explain any differences in the behaviour of the algorithm in the six ranges. Why does the algorithm behave differently for positive and negative values of x? Are the errors you observe rounding errors or truncation errors (as described in Section 1.2.4 of Heath)?

(f) (5 points) Give an algorithm for $\exp(x)$ that is fast and accurate for all values of x from -100 to 100. (Hint: $\exp(-x) = 1/\exp(x)$). Implement your algorithm in Matlab and test it.

University of Toronto Mississauga
CSC 338 - Numerical Methods, Spring 2010

Cover sheet for Assignment 1

Complete this page and hand it in with your assignment.

Name: ________________________________

(Underline your last name)

Student number: ________________________________

I declare that the solutions to Assignment 1 that I have handed in are solely my own work, and they are in accordance with the University of Toronto Code of Behavior on Academic Matters.

Signature: ________________________________