This assignment is due at the **start** of your lecture on Friday, 3 October 2014.

For the questions that require you to write a MatLab program, hand-in the program and its output as well as any written answers requested in the question. Your program and its output, as well as your written answers, will be marked. Your program should conform to the usual CS standards for comments, good programming style, etc. When first learning to program in MatLab, students often produce long, messy output. Try to format the output from your program so that it is easy for your TA to read and to understand your results. To this end, you might find it helpful to read "A short description of fprintf" on the course webpage. Marks will be awarded for well-formatted, easy-to-read output.

1. [6 marks]

   What are the approximate absolute and relative errors in approximating e (the base of the natural logarithms) by the following values?

   (a) 2.7
   (b) 2.7183
   (c) 2.718281828

   For the purposes of this question, you can assume that the "true" value of e is 2.718281828459046.

2. [10 marks]

   In a floating-point number system with parameters $\beta = 10$, $p = 3$, $L = -20$ and $U = +20$ that uses the *round-to-nearest* rounding rule and allows gradual underflow with subnormal numbers, what is the result of each of the following floating-point arithmetic operations?

   (a) $1.23 \cdot 10^0 + 5.14 \cdot 10^{-2}$
   (b) $1.58 \cdot 10^1 - 5.41 \cdot 10^{-1}$
   (c) $1.23 \cdot 10^1 + 5.14 \cdot 10^{-4}$
   (d) $2.66 \cdot 10^4 - 5.42 \cdot 10^6$
   (e) $3.76 \cdot 10^{12} - 7.69 \cdot 10^5$
   (f) $1.87 \cdot 10^1 + 4.31 \cdot 10^2$
   (g) $1.67 \cdot 10^{10} \times 5.43 \cdot 10^{-15}$
   (h) $-4.67 \cdot 10^{10}/(1.84 \cdot 10^{-15})$
   (i) $3.86 \cdot 10^{-10} \times 1.23 \cdot 10^{-12}$
   (j) $2.94 \cdot 10^{-10} \times 6.23 \cdot 10^{-15}$

Write each answer as a normalized 3-digit floating-point number if possible. If that is not possible, write your answer as a subnormal 3-digit floating-point number if that is the most accurate representation. If that is not possible either, then write your answer as +Inf, −Inf or NAN, whichever best represents your answer.

3. [5 marks]

   Consider the function $f(x) = x^{1/3}$ for real positive x.

   Is this function well-conditioned or ill-conditioned in a relative error sense with respect to small relative changes in the value of the input argument x?

   Justify your answer.

4. [10 marks: 5 marks for each part]

   Assume throughout this question that x is a floating-point number in your computer and that it contains no errors. Also assume that $-1 < x < 1$ and that there are no overflows or underflows in any of the computations below.

   It is easy to show that, in exact arithmetic,

$$\frac{1}{1-x} - \frac{1}{1+x} = \frac{2x}{(1-x)(1+x)}$$

   Which of the two mathematically equivalent expressions

$$\frac{1}{1-x} - \frac{1}{1+x} \qquad \text{or} \qquad \frac{2x}{(1-x)(1+x)}$$

   can be evaluated more accurately in floating-point arithmetic?

   (a) Show that one of the expressions always gives a very accurate answer in a relative error sense.

   (b) Give an example which illustrates that the other expression can give an answer that is much less accurate in a relative error sense.

5. [15 marks: 5 marks for each part]

   (a) Write a MatLab function exp1 to approximate e^x by summing the series

$$1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

   from left to right until the accumulated sum stops changing.
   Test your program by computing exp1(x) for $x = -25, -24, -23, \ldots, +25$ (i.e., x = −25 : +25 in MatLab).

For each value of x, compute the relative error

$$\frac{\exp1(x) - \exp(x)}{\exp(x)}$$

where $\exp(x)$ is the MatLab function that approximates e^x, and print both x and the relative error.

For the purpose of this question, assume $\exp(x) = e^x$.

Format your output neatly.

(b) For what values of x does your function produce accurate approximations to e^x and for what values of x does your function produce poor approximations to e^x?

Explain why your function performs well in the cases where it produces accurate approximations to e^x and also explain why your function performs poorly in the cases where it produces poor approximations to e^x.

Don't just say that it performs poorly because there is rounding error. There is rounding error in your computations for all values of x (except possibly $x = 0$). However, in some cases the rounding errors are insignificant and you obtain a good approximation to e^x, while in other cases the rounding errors are significant and you obtain a poor approximation to e^x. Explain why.

(c) Make a small change to your function exp1 so that it produces accurate approximations to e^x for all $x = -25, -24, -23, \ldots, +25$. Call your new function exp2. (If you find it helpful, you can call exp1 from within exp2.)

For each value of x, compute the relative error

$$\frac{\exp2(x) - \exp(x)}{\exp(x)}$$

where $\exp(x)$ is the MatLab function that approximates e^x, and print both x and the relative error.

Format your output neatly.

Hint: note $e^x = 1/e^{-x}$.