**M6803 Computational Methods in Engineering**

**AY 2022/2023 Assignment 1** Due Date: **Friday 9 Sep 2020**

For some questions, you will need to do extra reading beyond what is covered in the class. Please submit your solutions in soft copy through the course site, using the same link where you obtained this question paper. Please submit the source code of your programs in separate files.

1. Explain the representation of negative integers in terms of (i) a sign bit, (ii) ones complement and (iii) twos complement. Write the binary number of the decimal integer -161 using the three different ways.

2. Write a program that converts a given floating point binary number with a 24-bit normalized mantissa and an 8-bit exponent to its decimal (i.e. base 10) equivalent. For the mantissa, use the representation that has a hidden bit, and for the exponent use a bias of 127 instead of a sign bit. Of course, you need to take care of negative numbers in the mantissa also. Use your program to answer the following questions:

1. Mantissa: 11110010 11000101 01101010, exponent: 01011000. What is the base-10 number?
2. What is the largest number (in base 10) the system can represent?
3. What is the smallest non-zero positive base-10 number the system can represent?
4. What is the smallest difference between two such numbers? Give your answer in base 10.
5. How many significant base-10 digits can we trust using such a representation?

Your program should do the computations from first principle, and not use functions already available in MATLAB or Python, if you are using either language.

3. The sine function can be computed as the sum of an infinite power series

sin(*x*) = *x* |*x*| < ∞, *x* is in radians.

Compute the value of sin(*x*) for these three *x* values: 0.5, 5, 50. Note that this is an infinite series and you can only terminate the computation at a term that is very small (i.e. below a certain tolerance which you can set). Discuss your observations in doing these computations. Do you see any problem with the direct evaluation of this series on a computer? Why? Suggest any remedy you may have in overcoming the problem(s).

You can evaluate the series by writing a MATLAB or Python program if you wish. But because both languages can do their computation to a very high precision, the manifestation of the problem might not be obvious.

How about the evaluation of the binomial expansion below? Any similar problem?

, -1 < *x* < 1.

4. One common method for comparing the equality of two numbers is the *epsilon* test:

if *abs*(*a* – *b*) < *ε* then return *a* and *b* are equal

where *abs* is the absolute function that returns the positive value of a number, and *ε* is a very small positive number below which the value is considered insignificant.

Explain the rationale of this test. Is it good enough? If not, explain why and suggest an improvement.

Write a program to solve the quadratic equation *ax*2 + *bx* + *c* = 0, in which you should apply the above test appropriately to determine if the equation has real or complex roots.

5. Using the Taylor series, write (i) the first order approximation and (ii) the second order approximation to the function f(x) = x2 + sin(2x). Given the value of f(1), which you can evaluate accurately, find the first order and the second order Taylor series approximation value of f(1.2). (Again, *x* is in radians.)

<><><> The End <><><>