PY2105 - Assignment 2

Deadline: 10 October 2024

Problem 1 (**7 marks**):
a) The small angle approximation ($\sin(\theta) \approx \theta$) is widely used in physics. Write a program to calculate when the small angle approximation breaks down (e.g. when the relative absolute true error is more than 1%).

b) If one were to include the 2nd term of the series, what would be the error at the angle where the normal approximation was giving an error of 1%? Write a program to solve this question.

c) Write a program that can calculate at what angle the new approximation gives an error of 1%?

TIP: The $\sin(x)$ function of C++ works in radians and not degrees.

Problem 2 (**4 marks**):
Write a program that uses the central differentiation method to find the derivative at $x = 2$ of the function

$$f(x) = x^2 - x + 1$$

using a step size of $\Delta x = 0.25$. What is the absolute true error of your approximation? How does your error change if you use $\Delta x = 0.125$? Does the scaling of the error of your program reflect the scaling discussed in the lecture?

Problem 3 (**9 marks**):
a) Use a program (e.g. gnuplot or python) to plot the following function

$$f(x) = \frac{300x}{1 + e^x} \tag{1}$$

from a=0 to b=10. The exponential function in gnuplot is called "exp(x)".
b) Draw a flow diagram on how you will solve the integration with a computer code numerically.
c) Write a program code to numerically solve a definite integral via the trapezium rule. Allow variables to set the number of segments n, the lower and upper bound a and b, as well as for the step size $h = (b - a)/n$.
d) Use $n = 2$ and estimate the error from the true value of the integral of function (1). Why is the error for $n = 2$ so large (use a graphical representation)? Show how the error reduces as you increase n to 4, 8, 16, 32, and 64.

Please submit all solutions electronically via Canvas (scan or take a photo of the hand-written parts and submit the source code of all programs)!