3801ICT Numerical Algorithms – Assignment Part 1

Note:

a) This assignment must be done individually.
b) The programming language to be used is C++ but you may use Python to generate graphs for your reports.
c) For each question requiring a C++ program you must document the algorithm and show any test cases you used. Only submit a single Word document containing the documentation for all questions.
d) The submission time and date are as specified in the Course Profile and the submission method will be communicated during semester.

1. (**20 Marks**) A centered difference approximation of the first derivative can be written as:

$$\underbrace{f'(x_i)}_{\text{True value}} = \underbrace{\frac{f(x_{i+1}) - f(x_{i-1})}{2h}}_{\text{Finite-difference approximation}} \underbrace{- \frac{f^{(3)}(\xi)}{6} h^2}_{\text{Truncation error}}$$

However, as we are using a computer, the function values in the numerator of the finite-difference approximation include round-off errors as follows:

$$f(x_{i-1}) = \tilde{f}(x_{i-1}) + e_{i-1}$$
$$f(x_{i+1}) = \tilde{f}(x_{i+1}) + e_{i+1}$$

Substituting these values we get:

$$\underbrace{f'(x_i)}_{\text{True value}} = \underbrace{\frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h}}_{\text{Finite-difference approximation}} \underbrace{+ \frac{e_{i+1} - e_{i-1}}{2h}}_{\text{Round-off error}} \underbrace{- \frac{f^{(3)}(\xi)}{6} h^2}_{\text{Truncation error}}$$

Assuming that the absolute value of each component of the round-off error has an upper bound of ε, the maximum possible value of the difference $e_{i+1} - e_{i-1}$ will be 2ε. Further, assume that the third derivative has a maximum absolute value of M. An upper bound on the absolute value of the total error can therefore be represented as

$$Total\ error = \left| f'(x_i) - \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} \right| \leq \frac{\varepsilon}{h} + \frac{h^2 M}{6}$$

An optimal step size can be determined by differentiating this equation, setting the result equal to zero and solving to give:

$$h_{opt} = \sqrt[3]{\frac{3\varepsilon}{M}}$$

Given:

$$x = 0.5, f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.15x + 1.2$$

use a centered-difference approximation to estimate the first derivative of this function with varying values of h to demonstrate the validity of the analysis above and the impact of both round-off and truncation errors.

2. (**20 Marks**) Use the multiple-application Simpson's rule to evaluate the distance travelled where the velocity v as a function of time is as follows

$$
\begin{array}{ll}
v = 11t^2 - 5t & 0 \le t \le 10 \\
v = 1100 - 5t & 10 \le t \le 20 \\
v = 50t + 2(t-20)^2 & 20 \le t \le 30
\end{array}
$$

In addition use numerical differentiation to develop graphs of acceleration (dv/dt) and rate of change of acceleration for t = 0 to t = 30.

3. (**15 Marks**) Compare the Central Difference and Richardson Extrapolation methods for finding the value of the first derivative of $f(x) = -0.2x^4 - 0.30x^3 - 1.5x^2 - 0.45x + 2.6$ at $x = 0.5$. Explore the effect of using different values for h.

4. (**15 Marks**) A variable V is determined by:

$$V = \int_{t_1}^{t_2} P(t)d(t)dt$$

where:

$$P(t) = 9 + 4cos^2(0.4t)$$
$$d(t) = 5e^{-0.5t} + 2e^{0.15t}$$

Evaluate this integral between $t_1 = 2\ and\ t_2 = 8$ using Romberg Integration with a tolerance of 0.1%

5. (**15 Marks**) The deflection of a rod can be modelled as:

$$\frac{d^2y}{dz^2} = \frac{f}{2EI}(L-z)^2$$

Where *f* = force, *E* = modulus of elasticity, *L* = rod length, and *I* = moment of inertia. Programmatically in C++ calculate the deflection if *y* = 0 and *dy/dz* = 0 at *z* = 0. Use *f* = 60, *L* = 30, *E* = 1.25 x 10^8 and *I* = 0.05.

6. (**15 Marks**) Using a C++ program, compare the performance of the Bisection, Newton-Raphson and Secant methods in estimating the root of $f(x) = e^{-x} - x + x^2$. Start with initial estimates of $x_{-1} = 0$ and $x_0 = 1.0$.