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

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

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:

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

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. (**15 Marks**) When an aircraft is being tracked by radar the position of the aircraft is determined by distance (calculated from the return time of pulse) and the sweep angle of the radar. To give a meaningful radar display this information needs to be converted to cartesian coordinates and velocity and acceleration (both vectors) need to be calculated. Write a program to perform this operation (use centered finite differences (second-order correct)) and test your program with the data shown in the table below.

T, s	200	202	204	206	208	210
θ, rad	0.75	0.72	0.70	0.68	0.67	0.66
R, m	5120	5370	5560	5800	6030	6240

3. (**15 Marks**) Compare the Central Difference and Richardson Extrapolation methods for finding the value of the first derivative of $f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$ at $x = 0.5$. Explore the effect of using different values for h.

4. (**15 Marks**) The depths of a river H are measured at equally spaced distances across a channel as tabulated below. The rivers cross-sectional area can be determined by integration as in:

$$A_c = \int_0^x H(x)dx$$

   Write a C++ program that uses Romberg integration to perform the integration to a stopping criterion of 1%.

x, m	0	2	4	6	8	10	12	14	16
H, m	0	1.9	2	2	2.4	2.6	2.25	1.12	0

5. (**15 Marks**) The deflection of a boat mast by the wind can be modelled as:

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

   Where *f* = wind force, *E* = modulus of elasticity, *L* = mast 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 $f(x) = e^{-x} - x$. Start with initial estimates of $x_{-1} = 0$ and $x_0 = 1.0$.