3801ICT Numerical Algorithms – Assignment (100 Marks)

This assignment must be done individually. The programming language to be used is C++ but you may use Python to generate graphs for your reports. The submission time and date is 11pm on Sunday, 17th May, 2020 and the submission method will be communicated during semester.

For each question you are required to produce a C++ program and a supporting document which describes your algorithm (including any preliminary problem analysis) and the testing and associated results used to verify your program.

1. (**14 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 value 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:

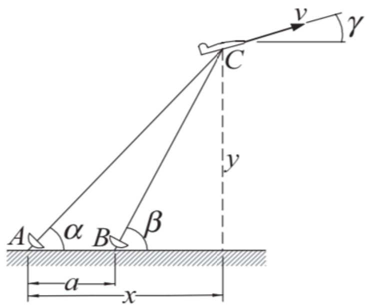
$$h_{opt} = \sqrt[3]{\frac{3\varepsilon}{M}}$$

Given:

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

Write a C++ program that uses a centered-difference approximation to estimate the first derivative of this function with varying values of h and varying precision to demonstrate the validity of the analysis above and the impact of both round-off and truncation errors.

2. (**12 Marks**) Write a C++ program to solve the following problem:



Radars A and B, distance a = 500 m apart, track plane C by recording angles α and β at one-second intervals. Three successive readings are

	9	10	11
A (degrees)	54.80	54.06	53.34
B (degrees)	65.59	64.59	63.62

Calculate the speed v of the plane and the climb angle γ at t = 10 s. The (x, y) coordinates of the plane are:

$$x = a\frac{\tan\beta}{\tan\beta - \tan\alpha}$$

$$y = a\frac{\tan\alpha\ tan\beta}{\tan\beta - \tan\alpha}$$

3. (**10 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

4. (**10 Marks**) Given the following formula for a falling body:

$$v = \frac{gm}{c}(1 - e^{-\left(\frac{c}{m}\right)t})$$

where g = 9.8 m/s^2 and linear drag c = 10 kg/s. Write a C++ program that uses Romberg integration which determines how far the body falls, to an approximate error of 1%, in the first 8 seconds when m = 80kg.

5. (**10 Marks**) Write a C++ program to solve the following problem: Assuming drag is proportional to velocity squared the velocity of a falling object is:

$$\frac{dv}{dt} = g - \frac{c_d}{m}v^2$$

Where v is the velocity (m/s), g is gravitational acceleration (m/s^2), c_d (0.225 kg/m) is drag coefficient and m (= 90kg) is mass. Solve for distance fallen and speed. Given an initial height of 1 km and no vertical velocity find the time to ground. Use Euler's method and a fourth-order RK method and compare the results.

6. (**10 Marks**) Write a C++ program to solve the following problem: The speed v of a rocket in vertical flight near the surface of earth can be approximated by

$$v = u \ln \frac{M_0}{M_0 - \dot{m}t} - gt$$

where

u = 2 510 m/s = velocity of exhaust relative to the rocket
M_0 = 2.8 × 10^6 kg = mass of rocket at lift off
$\dot{m}$ = 13.3 × 10^3 kg/s = rate of fuel consumption
g = 9.81 m/s^2 = gravitational acceleration
t = time measured from lift off

Determine the time when the rocket reaches the speed of sound (335 m/s).

7. (**10 Marks**) The trajectory of a thrown object can be computed as:

$$y = (\tan \theta_0)x - \frac{g}{2v_0^2 cos^2\theta_0} x^2 + y_0$$

In C++ find the maximum height attained given initial height y_0 = 1m, initial velocity v_0 = 25 m/s and initial angle $\theta_0 = 50$. The approximate error in your answer must be less than 1%. Use a value of 9.81 m/s^2 for g.

8. (**12 Marks**) Write a C++ program that uses least-squares regression to compare two models that produced the results shown in the following table. You should identify which model is more correct and why.

Time (Seconds)	Measured	Model 1	Model 2
1	10.00	8.953	11.240
2	16.30	16.405	18.570
3	23.00	22.607	23.729
4	27.50	27.769	27.556
5	31.00	32.065	30.509
6	35.60	35.641	32.855
7	39.00	38.617	34.766
8	41.50	41.095	36.351
9	42.90	43.156	37.687
10	45.00	44.872	38.829
11	46.00	46.301	39.816
12	45.50	47.490	40.678
13	46.00	48.479	41.437
14	49.00	49.303	42.110
15	50.00	49.988	42.712

9. (**12 Marks**) Write a C++ program that uses Newton's interpolating polynomial to determine y at x = 3.5 to the best possible accuracy. Compute the finite divided difference and order your points to attain optimal accuracy and convergence.

x	0	1	2.5	3	4.5	5	6
y	2	5.4375	7.3516	7.5625	8.4453	9.1875	12