Computing and Numerical Methods 2
Coursework

Academic Year 2024/2025

Deadline: 24th January 2025, 11pm GMT

Instructions

This coursework is divided into two parts. You are required to follow the instructions for each part detailed below.

Part I:
This part consists of four exercises and involves programming in Matlab. You are required to submit all your MATLAB codes (in a zipped folder) and a report. You should address each question in a report written in the following format **font: calibri, font size: 12pt, line spacing: single**. Each part addressed must be clearly labelled (*i.e.* indicating in subheadings the specific question Qx(y) you are addressing, where x refers to the question number and y to the specific part of Qx).
For each part, **it is indicated what your response should include (either plots only, discussion only or discussion + relevant plots) and how long it can be** (in terms of a maximum number of lines). If you prefer to present the plots for different questions within a single figure, please ensure that the figure is properly labelled with a clear legend and title to distinguish each plot. Additionally, make sure to refer to the figure appropriately wherever it is discussed in your work. The **Part I report should not exceed 5 pages.** A single equation counts as one line (also if it involves matrices/vectors that span multiple lines). Note that it is possible to respond to Part I well below the 5-page limit, so do not feel pressured to submit 5 pages if your response is shorter: clear, concise (but complete) statements are generally better than statements that are unnecessarily long-winded! Make sure that your response adheres to these guidelines and that all figures are formatted so that they are clear, legible and labelled correctly. Hand-written responses will not be considered.

Having completed the tutorial sheets, you will already have the core parts of the Matlab code required to complete Part I. While marks will be awarded for each question, questions requiring analysis carry a heavier weight. Any discussions/analysis can be made solely based on the course material and observations made regarding the questions within this coursework. Make sure that your answers to these parts are clear, concise and to-the-point.

Part II:
This part requires you to write a program in C++. While the structure of the program is not prescribed, marks will be awarded for well-structured code which uses good programming practices and is clear to read. You should ensure your program adheres to any requirements on the input and output format of parameters and generated data.

The report should be written in 12pt font. The **Part II report should not exceed five A4 pages**; any material beyond the 5-page limit will not be marked. Make sure that your answers to these parts are clear, concise and to-the-point. Plots should be legible and any figure text should be no smaller than 12pt font.

Part I - Numerical Methods

The objective of this part is to understand and apply numerical methods for solving second-order ordinary differential equations, specifically modelling the motion of a floating object using the buoyancy model.

Consider a floating object in a liquid of density ρ_l. The object has uniform mass density ρ, constant cross-sectional area A and is of height L.

According to the principle of buoyancy, the force on the object is governed by two primary forces:

1. Weight of the object: Acts downward;
2. Buoyancy force: Equal to the weight of the liquid displaced by the object, acting upward.

The governing equation for the displacement $y(t)$ from the equilibrium position can be derived as:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} + \omega^2 y(t) = 0, \tag{1}$$

where

$$\omega^2 = \frac{\rho_l g}{\rho L}. \tag{2}$$

The object is initially displaced from its equilibrium position $\bar{y}$ and then released, allowing it to oscillate in the liquid. We have the initial conditions:

- $y(0) = y_0$ (initial displacement)
- $\frac{\mathrm{d}y}{\mathrm{d}t}(0) = y_0'$ (initial velocity)

Use the following parameters for the part:

- $\rho = 500$ kg/m^3;
- $\rho_l = 1000$ kg/m^3;
- $g = 9.81$ m/s^2;
- $L = 1$ m;
- Initial displacement $y_0 = 0.1$ m;
- Initial velocity $y_0' = 0$ m/s.

Continues on next page ...

Tasks

Q1) (a) Derive the analytical solution of the differential equation for $y(t)$ with the given initial conditions. Provide a detailed derivation for the given values of ρ, ρ_1, L and g. [2.5%]

(b) Provide a plot of the displacement of the floating object y vs time t for the range of $t = 0$ to $t = 10$ s to represent your solution, for the given values of ρ, ρ_1, L and g. **(Provide a single plot only)** [2.5%]

(c) Write the second order ODE as a system of first order ODEs, as $\dot{\mathbf{Y}} = \mathbf{AY}$, where A is a 2×2 matrix, and $\mathbf{Y} = [y, y']^\top$, where y' is the derivative of y respect to time. [2.5%]

(d) Find the eigenvalues of matrix A, and discuss their connection to the parameters in the exact solution of the system. [2.5%]

Q2) Consider the given values of ρ, ρ_1, L and g.

(a) Write a code in MATLAB to solve the system of ODEs using explicit Euler with step size $\Delta t = 0.1$. Provide a plot of the trajectory of the relative motion (i.e. a plot of position y vs time t for the range of $t = 0$ to $t = 10$) to represent your solution. **(Provide a single plot only)** [2.5%]

(b) Solve the system of ODEs using 4th order Runge-Kutta with step size $\Delta t = 0.1$. Provide a plot of the trajectory of the relative motion (i.e. a plot of position y vs time t for the range of $t = 0$ to $t = 10$) to represent your solution. **(Provide a single plot only)** [2.5%]

(c) Again, asssuming $\Delta t = 0.1$, obtain a numerical solution using MATLAB's built-in function `ode45`. **(Provide a single plot only)** [2.5%]

(d) Compare what you obtained using explicit Euler, 4th order Runge-Kutta, and the `ode45` function with the exact solution of the system obtained in Q1(b). **(Max 5 lines + a relevant plot)** [2.5%]

(e) Provide a plot of the velocity v of the mass versus its position y. Explain what you obtain. **(Max 5 lines + a relevant plot)** [2.5%]

(f) Compare the obtained solutions in terms of

i. how reliable the solutions are (do they represent the exact solution well?)

ii. computational complexity.

(Max 15 lines) [2.5%]

Q3) In this question we are interested in gaining some insights related to numerical stability, using the theory you have learnt in the course.

(a) Calculate the error between the analytical solution and each numerical solution at each time step. Plot the error over time for each method and discuss which method provides the most accurate results and why. **(Max 5 lines + relevant plot(s))** [3%]

(b) Suppose we are interested in solving the system of ODEs using explicit Euler (with fixed step size) over some time span $t \in [0, T]$. Experiment with at least 3 different step sizes (e.g., $\Delta t = 0.5$, $\Delta t = 0.05$ and $\Delta t = 0.005$) and observe the impact on the error. Comment on your observations – are there any selections for Δt that result in a numerically stable scheme? Note that you may find it useful to vary T and/or zoom in on your plots. **(Max 5 lines + relevant plot(s))** [3%]

(c) Suppose we are interested in solving the system of ODEs using 4th order Runge-Kutta (with fixed step size) over some time span $t \in [0, T]$. Experiment with at least 3 different step sizes (e.g., $\Delta t = 0.5$, $\Delta t = 0.05$ and $\Delta t = 0.005$) and observe the impact on the error. Comment on your observations - are there any selections for Δt that result in a numerically stable scheme? Note that you may find it useful to vary T and/or zoom in on your plots. **(Max 5 lines + relevant plot(s))** [3%]

(d) Investigate the effect of initial condition $y(0)$ and $y'(0)$. For which values of $y'(0)$ the object would sink? **(Max 15 lines + relevant plot(s))** [6%]

Q4) (a) Discuss how the buoyancy model differs from the spring-mass model in terms of physical interpretation and mathematical formulation. Explain any challenges you encountered while implementing the numerical methods and how you addressed them. [3%]

(b) Fluid resistance due to viscosity and drag plays a significant role in the motion of a floating object. To incorporate this effect, a damping term is added to the governing equation, resulting in:

$$\frac{d^2y}{dt^2} + c\frac{dy}{dt} + \omega^2 y(t) = 0$$

where c is the damping coefficient, which depends on the object's shape and the fluid properties. This term accounts for energy loss due to fluid resistance, leading to a gradual reduction in the oscillation amplitude over time.

i. **Explain** how the inclusion of the damping term changes the dynamics of the system compared to the undamped case.

ii. **Explain** how considering damping would affect the numerical stability of the system.

iii. Solve the resulting system of ordinary differential equations (ODEs) using the 4th-order Runge-Kutta method with the damping coefficient $c = 0.2$, and a step size $\Delta t = 0.1$.

iv. Generate a plot of the relative motion (position y vs time t) for $t \in [0, 10]$, clearly showing the damping effect on the oscillations.
(Max 10 lines + relevant plot(s)) [7%]

Continues on next page ...

Part II: C++ Programming

The objective of this part of the assignment is to use C++ to solve a numerical problem, demonstrating good code design and robust implementation.

Consider an extension of the system in Part I as follows. Suppose that the object is now a hollow vessel with a hole at the bottom into which water gradually enters. To simplify our problem, we assume that the vessel has mass m and is of infinite length, and its walls are of negligible thickness (that is, the interior and exterior cross-sectional areas are equal). We also assume that water enters the hole in the vessel at a mass flow-rate proportional to the difference in the water level inside and outside the vessel, with a constant of proportionality K. The flow-rate is assumed to be independent of the motion of the vessel.

The volume of water inside the vessel is time-dependent and is governed by the following first-order differential equation:

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{K}{\rho_l A}\left(\bar{y} + y(t) - h(t)\right)$$

where the equilibrium position of the bottom of the empty vessel is $\bar{y} = \frac{m}{\rho_l A}$.

In addition to weight and buoyancy forces, the vessel motion is further subjected to a drag force in the direction opposite to its motion, resulting in the following terms:

$$\begin{aligned} \text{Weight}: &\quad F_w = mg + \rho_l g A h(t) \\ \text{Buoyancy}: &\quad F_b = -\rho_l g A(\bar{y} + y(t) - h(t)) \\ \text{Drag}: &\quad F_d = -\frac{1}{2}\rho A C_d \left|\frac{\mathrm{d}y}{\mathrm{d}t}\right| \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right) \end{aligned}$$

where the total acceleration is given by

$$(m + \rho_l A h(t))\frac{\mathrm{d}^2}{\mathrm{d}t^2}(\bar{y} + y(t)) = F_w + F_b + F_d$$

and the downward direction is considered as positive.

Substituting expressions for the forces and the equilibrium position, and rearranging, we find that the motion of the vessel is governed by the following second-order differential equation:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = \frac{\rho_l A}{m + \rho_l A h(t)}\left[2gh(t) - gy(t) - \frac{1}{2}C_d\left|\frac{\mathrm{d}y}{\mathrm{d}t}\right|\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)\right]$$

Continues on next page ...

Tasks

Q5) Write an object-oriented C++ code to numerically simulate the motion of N vessels in a fluid, from time $t = 0$ to time $t = T$. Your code should:

- read a file called `parameters.txt` (filename all lowercase), located in the current working directory, which contains the following data:
  - The first line should contain: an integer indicating the time integration scheme to use (0 = explicit Euler (FE), 1 = 4th-order Runge-Kutta (RK4)); the values of T, Δt, g and ρ and C_d;
  - Each subsequent line should describe one vessel in the system, containing its mass m_i, cross-sectional area A_i, flow-rate coefficient K_i, initial displacement from its equilibrium position $y_i(0)$, its initial velocity $\dot{y}_i(0)$ and the initial water level $h_i(0)$.

  These values must be specified in the file exactly as described above, with values on a single line separated by one space. Lines beginning with a # symbol should be ignored. An example input file is provided with this assignment;
- time-integrate the system using one of: forward Euler or 4th-order Runge-Kutta;
- write the vessel displacements, velocities and water levels to file `output.txt` (all lowercase) in the current working directory at each time step. Each line should contain the following values in order, expressed using 6 significant figures of precision, using 10 character column widths, padding with spaces: $t \quad y_1(t) \quad \dot{y}_1(t) \quad h_1(t) \quad \ldots \quad y_N(t) \quad \dot{y}_N(t) \quad h_N(t)$
- not require interaction with the user through the terminal during execution;
- not use third-party libraries or code beyond the standard C++ header files.

Marks will be awarded for the following:

- Writing code which aspires to meet all the requirements above;
- Appropriate use of classes, functions, data structures; minimising code duplication; no memory leaks;
- Code organisation and formatting; sensible choice of variable/function names; use of comments. [20%]

Q6) Write a short report as described below (max 3 pages). You may use any software to generate the plots. The code used to produce the plots should be included in an appendix (not counted in the 3 pages).

(a) Use the provided in the table below to assess the correctness of your code. For each case,
  i. identify and state the maximum time-step which gives a converged solution for FE and RK4.
  ii. state, in table form, the final displacements, velocities and water levels of the vessels.
  iii. plot the displacement of the vessel base from the $\bar{y}$ as a function of time, for FE and RK4 [10%]

Case	System Parameters	Vessel parameters					
		m	A	K	y_0	$\dot{y}_0$	h_0
Single vessel, no drag	$T = 1$, $C_d = 0.0$	1.0	-0.1	0.0	-0.1	0.0	0.0
Single vessel, low drag	$T = 1$, $C_d = 0.1$	5.0	0.5	0.0	0.1	-0.5	0.0
Sinking vessel	$T = 2$, $C_d = 0.0$	100.0	0.1	1000.0	0.0	0.0	0.0
Three vessels	$T = 10$, $C_d = 0.01$	1	0.1	20	-0.1	0.0	0.0
		10.0	0.5	10.0	-0.1	0.0	0.0
		50.0	10.0	1.0	-0.1	0.0	0.0

(b) Use your code to numerically investigate the relationship between individual vessel parameters and the frequency of oscillation of the vessel.
  i. Which parameters affect the frequency of oscillation?
  ii. For those parameters, plot the oscillation frequency as a function of parameter value. [5%]

(c) Explain why it is advantageous to solve this problem, as well as more complex numerical problems, using C++ instead of an interpreted language. [5%]

(d) Describe your implementation approach: How did you use the STL and object-oriented programming paradigms? Which aspects of the problem did you encapsulate in using classes and why? How did you ensure ensure your code was reasonably CPU- and memory-efficient? [10%]

Submission

You should submit the **four files** below to the **four separate submission boxes** on Blackboard before the deadline. Make sure that these files have the format and names as specified below.

1. `Matlab.zip`: The code component of Part I must be submitted as a **ZIP** archive. It should contain the necessary MATLAB codes files. While there is no mark for the MATLAB codes, they should produce the same results and figures you presented in the report. Your report will not be assessed if you fail to submit the supporting codes.
2. `Part1.pdf`: The report corresponding to Part I of this coursework must be submitted as a **PDF** (*i.e.* not as a word file).
3. `Code.zip`: The code component of Part II must be submitted as a **ZIP** archive. It should contain the necessary C++ source code files only. For this, the `xarchiver` program, available from the *Accessories* menu on the remote Linux environment can be used. Please unpack the archive file again before submitting to check all the necessary files have been included and your code compiles.
4. `Part2.pdf`: The report component of Part II must be submitted a **PDF** (*i.e.* not as a word file).

You may make unlimited submissions in each of the submission boxes. Your last attempt before the deadline will be marked in each case.

End of assignment.