Computing and Numerical Methods 2
Coursework

Academic Year 2022/2023

Deadline: 27th January 2023, 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 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). 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 three A4 pages**; any material beyond the 3-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.

Notation: Given a vector v, its derivative with respect to an independent variable t is denoted by $\dot{v}$. Namely

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \dot{v}\,.$$

In this assignment you are going to numerically solve a system of ordinary differential equations (ODE) that describes the motion of celestial bodies. Note that while the ODE is in the same format as those encountered in celestial mechanics, parameters have been selected to keep the computational complexity low (to avoid long run times) - thus do not be startled by the, perhaps surprising, values of masses and gravitation used in this coursework!

A system consists of N bodies. Their motion is governed by the gravitational attraction of the bodies with each other. Newton's law of gravitation states that for two bodies i and j,

$$\mathbf{F}_{ij} = G\frac{m_i m_j}{d_{ij}^2}\hat{\mathbf{d}}_{ij} \,,$$

where $\mathbf{F}_{ij}$ is the force acting on body i, due to the presence of body j, G is the gravitational constant and m_i and m_j are their masses. The distance d_{ij} between the two bodies, with positions $\mathbf{x}_i = (x_i, y_i)$ and $\mathbf{x}_j = (x_j, y_j)$, is given by

$$d_{ij} = \|\mathbf{x}_j - \mathbf{x}_i\|$$

and the unit vector pointing from body i towards body j is given by

$$\hat{\mathbf{d}}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{d_{ij}}.$$

Newton's second law allows us to express this relationship as a system of second-order differential equations for each interaction between two bodies:

$$\begin{aligned}\ddot{x}_i &= G\frac{m_j}{d_{ij}^3}(x_j - x_i),\\ \ddot{y}_i &= G\frac{m_j}{d_{ij}^3}(y_j - y_i).\end{aligned}$$

Each body (the i-th body) is influenced by all other bodies in the system and, thus, its overall dynamics in two dimensions is given by the system of equations

$$\begin{aligned}\ddot{x}_i &= G\sum_{j\neq i}\frac{m_j}{d_{ij}^3}(x_j - x_i),\\ \ddot{y}_i &= G\sum_{j\neq i}\frac{m_j}{d_{ij}^3}(y_j - y_i).\end{aligned} \tag{1}$$

Thus, the motion of celestial bodies can be obtained by solving equations (1) subject to initial conditions $x_i(0)$, $y_i(0)$, $\dot{x}_i(0)$, $\dot{y}_i(0)$.

When considering the motion of a single celestial body ($i = 2$) about a second body ($i = 1$) of significantly larger mass (e.g. a satellite orbiting Earth), it is often of interest to study the motion of the former relative to the latter. This relative motion evolves according to the ordinary differential equations

$$\begin{aligned}\ddot{x}_r &= -G\frac{m_1 + m_2}{d_{12}^3}x_r \,,\\ \ddot{y}_r &= -G\frac{m_1 + m_2}{d_{12}^3}y_r \,,\end{aligned} \tag{2}$$

where x_r and y_r denote the relative position of the lighter body with respect to the heavier body and, thus $x_r(0) = x_2(0) - x_1(0)$, $y_r(0) = y_2(0) - y_1(0)$, $\dot{x}_r(0) = \dot{x}_2(0) - \dot{x}_1(0)$,and $\dot{y}_r(0) = \dot{y}_2(0) - \dot{y}_1(0)$.

Part I - Numerical Methods

In the first part of this coursework we will consider relative motion.

Table 1: Parameters to be used in Part I.

m_1	m_2	G	$x_r(0)$	$y_r(0)$	$\dot{x}_r(0)$	$\dot{y}_r(0)$
625	1	1	1	0	0	25

Q1) Considering the ODE (2), define a vector v of appropriate dimension and write an initial value problem, involving *a system of first order ODEs*, that describes the relative motion of body 2 (the lighter body) with respect to body 1 (the heavier body). **(Max 3 lines)** [5%]

Q2) Suppose the parameters in (2) and initial conditions are as given in Table 1. We will now solve the corresponding system of ODEs (obtained in Q1) using different solvers, over the time span $t \in [0, T]$, with $T = 1$.

(a) Solve the system of ODEs using explicit Euler with step size $\Delta t = 0.001$ Provide a plot of the trajectory of the relative motion (*i.e.* a plot of y_r vs x_r for all $t \in [0, 1]$) 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 $\Delta t = 0.001$ Provide a plot of the trajectory of the relative motion (*i.e.* a plot of y_r vs x_r for all $t \in [0, 1]$) to represent your solution. **(Provide a single plot only)** [2.5%]

(c) 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) [10%]

Q3) In this question we are interested in gaining some insights related to numerical stability.

(a) 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 different step sizes Δt and 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))** [5%]

(b) 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 different step sizes Δt and 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))** [5%]

Q4) In Q3 we have explored numerical stability by "trial and error". In this question we will see if our observations are in accordance with what local stability analysis tells us.

(a) Compute the linearisation of (2) about a generic point $x_r^\star$, $y_r^\star$, $\dot{x}_r^\star$, $\dot{y}_r^\star$. The result should be in the form

$$\dot{v} \approx Av + \bar{b}t + \bar{c}\,,$$

where A is a matrix of appropriate dimension, whereas $\bar{b}$ and $\bar{c}$ are constant vectors of approriate dimension (potentially zero). **(Max 4 lines)** [5%]

(b) Let x_r^* and y_r^* vary over a uniform grid of spacing $\Delta_{x,y} = 1$, such that $1 \leq |x_r^*| \leq 5$ and $\leq 1|y_r^*| \leq 5$ Namely, create a square grid for the points x_r^* and y_r^* about which the linearisation is performed and for each grid point compute the eigenvalues of the matrix A. Plot the locations of the eigenvalues of A computed at each grid point in a single plot (on the complex plane). Comment on whether or not you observe any pattern. **(Provide a single plot + max 3 lines)** [2.5%]

(c) Are the observations made in Q3) consistent with what you have observed in Q4)(b)? Hint: Remember that linear analysis only gives us "local" information! **(Max 20 lines)** [12.5%]

Part II: C++ Programming

Q5) Write a C++ code to numerically simulate the motion of N bodies under gravitation using SI units (i.e. metres, seconds, kilograms). Solve the system, given by equation (1), from time 0 to time T. Your code should:

- read a file called `parameters.txt` (all lowercase), located in the current working directory, which contains the following:
  - The first line should contain the values of G, T and Δt;
  - Each subsequent line should describe a body in the system, containing its absolute initial position $x_i(0)$, $y_i(0)$, initial velocity $\dot{x}_i(0)$, $\dot{y}_i(0)$, and mass m_i.

  These values must be specified in the file exactly as described above, with values on a single line separated by one space, for example:
  ```
  1.0 5.0 0.001
  3.0 3.0 0.0 0.0 1.0
  ```
- integrate the system using the 4th-order Runge-Kutta scheme.
- write the body index i, time t, (x_i, y_i) positions of the bodies and their velocity $(\dot{x}_i, \dot{y}_i)$ of each body at each time step to the file `output.txt` (all lowercase) in the current working directory;
- 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:

- Code correctness;
- Code design (appropriate use of object-oriented paradigms, functions, suitable data structures);
- Code readability (layout, sensible choice of variable/function names, use of comments). [30%]

Q6) Write a short report as described below (max 3 pages). You may use any software to generate the plots.

(a) Test your code using the initial conditions provided in the table below. Use $G = 1$, $T = 1.0$ and $\Delta t = 0.01$. For each case, plot the trajectories of the bodies and state, in table form, the final positions and velocities of each body. For cases with $N \leq 2$, are your results consistent with your MATLAB code?

Case	Body Index i	$x_i(0)$	$y_i(0)$	$\dot{x}_i(0)$	$\dot{y}_i(0)$	m_i
One Body	1	0.0	0.0	1.0	1.0	1.0
Deflect	1	0.0	0.0	0.0	0.0	10.0
	2	1.0	-1.0	0.0	5.0	1.0
Orbit1	1	0.0	0.0	0.0	0.0	625.0
	2	1.0	0.0	0.0	25.0	1.0
Orbit2	1	0.0	0.0	0.0	0.0	500.0
	2	1.0	0.0	0.0	25.0	1.0
	3	-1.0	0.0	0.0	-25.0	1.0

[4%]

(b) In contrast to Q3(b), now consider the two-body system:

Body Index i	$x_i(0)$	$y_i(0)$	$\dot{x}_i(0)$	$\dot{y}_i(0)$	m_i
1	0.5	0.0	0.0	1.0	1.0
2	-0.5	0.0	0.0	0.87	1.0

with $G = 1$, $T = 10$. Investigate the convergence of the dynamics of this system with the time-step Δt. Plot the trajectories of the converged dynamics. State the choice of Δt which you consider provides a converged solution and explain why you believe the solution is converged. [4%]

(c) Look up appropriate physical values to simulate two orbits of the isolated Earth-Moon system. Include the `parameters.txt` file you created as well as a plot of the Moon's trajectory relative to the Earth. [4%]

(d) List the potential advantages and disadvantages of solving numerical problems, such as the one in this assignment, using C++ instead of an interpreted language. [4%]

(e) Describe how you exploited the STL and object-oriented programming paradigms in your code. [4%]

Submission

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

1. `Part1.pdf`: The report corresponding to Part I of this coursework must be submitted as a **PDF** (*i.e.* not as a word file).
2. `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.
3. `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.