EOSC 213 - Computational Methods for Geological Engineering

Homework 02

Due: 20th February, 2026 at 11:59 pm

Academic Integrity Policy: This assignment is to be completed individually. You may discuss ideas, concepts, and approaches with classmates, but all written work and Python codes must be your own. **The use of generative AI tools including chatbots (such as ChatGPT, Claude, GitHub Copilot etc.), solution manuals, or online sources that provide direct answers or worked solutions is strictly prohibited**. Searching for or copying complete solutions from websites, forums, or other students is not allowed. Any external resources used beyond the course materials must be explicitly cited. Submissions that are substantially similar, or that show evidence of unauthorized assistance, will be treated as violations of academic integrity and handled according to university policy.

Submission Instructions: For the theoretical (written) problems in this homework, you must submit handwritten solutions. Write your work neatly and clearly state any assumptions made. Scan or take clear photos of all pages and **combine them into a single PDF file** before submission. Ensure that **all pages are legible, correctly ordered, and properly oriented**. For the programming component, submit the **Jupyter Notebook (.ipynb)** exactly as specified in the programming instructions provided here (submit both .ipynb notebook and its exported .html versions). Submissions that are incomplete, incorrectly formatted, or split across multiple files may not be graded.

Problem 1: ODE Detective – Identifying Structure, Order, and Linearity

(a) Separable or not?

For each of the following ODEs, determine whether it is **separable** or **not separable**. If you claim the equation is *separable*, briefly show how it can be rearranged into the form,

$$g(y)\,dy = h(t)\,dt,$$

and state the expressions $g(y)$ and $h(t)$ in each of those cases. No need to give an explanation if it is *not separable*.

(1) $\frac{dy}{dt} = (t^2 + 1)\,y^3$

(2) $\frac{dy}{dt} = \sin t/(1 + y^2)$

(3) $(t + 1)\frac{dy}{dt} = y^2/(1 + y)$

(4) $\frac{dy}{dt} = y + t^2$

(5) $\frac{dy}{dt} = t\,e^{ty}$

(6) $\frac{dy}{dt} = (y - t)/(y + t)$

(b) Order and linearity

An ODE is **linear in** y if it can be written as,

$$a_n(t)\,y^{(n)} + a_{n-1}(t)\,y^{(n-1)} + \cdots + a_1(t)\,y' + a_0(t)\,y = g(t),$$

where the coefficients $a_k(t)$ depend only on t, and y and its derivatives $y^{(k)}$ (where the superscript (k) denotes the k^{th} derivative) appear only to the **first power** (not multiplied together and not inside nonlinear functions).

For each of the following differential equations, state (i) the **order** of the equation, and (b) whether it is **linear** or **nonlinear** in y.

(1) $y'' + t\,y' - y = e^t$

(2) $(y')^2 + y = t$

(3) $y''' + y\,y'' = 0$

(4) $\frac{d}{dt}(t\,y) + y = \sin t$

(5) $y'' + (y')^3 = 0$

(6) $t^2y'' + ty' + y = 0$

Note: No solving is required in Problem 1. Your task is purely classification and justification.

Problem 2: ODE Solver – Analytical Solutions of Initial Value Problems

In this problem, you will solve a collection of ODEs *analytically*. Each equation is accompanied by an initial condition. Your task is to identify appropriate solution methods (such as separation of variables, partial fraction decomposition, or the integrating factor method), carry out the necessary algebra and integrations, and determine the solution ($y(t)$) that satisfies the given initial condition. Show all essential steps clearly and justify any transformations you use. For this problem, assume $t \geq 0$. You should clearly state any additional restrictions on t that might arise during the solution process.

(1) $\frac{dy}{dt} = \big((y+1)(y+3)\big)/\big(t(y+2)\big), \quad y(1) = 0.$

(2) $\frac{dy}{dt} = (y/t) + (y/t)^2, \quad y(1) = 1.$ Are there values of t in $t \geq 0$ where the final solution $y(t)$ is undefined?

(3) $\frac{dy}{dt} = t^4 + 3t^2 + 1 - 2ty - \frac{y}{t}, \quad y(1) = 0.$ Are there values of t in $t \geq 0$ where the final solution $y(t)$ is undefined?

(4) $\frac{dy}{dt} = t - \frac{y}{t} - y\tan t, \quad y(\frac{\pi}{6}) = 0.$ For this problem, you won't be able to compute the final integral in terms of elementary functions (like polynomials, logarithms, exponentials, or trigonometric functions), so feel free to leave the final solution of $y(t)$ in the integral form as a function of t. Are there values of t in $t \geq 0$ where the final solution $y(t)$ is undefined?

Problem 3: Forward Euler with Polynomially-Growing Decay

Consider the initial value problem,

$$\frac{dy}{dt} = -t^2 y, \qquad y(0) = y_0,$$

whose exact solution is,

$$y(t) = y_0 e^{-t^3/3}.$$

This is similar to the exponential decay model discussed in class but with $\lambda(t) = t^2$ (an increasing function of t; not a constant). Let $h > 0$ be a fixed time step and define the time grid $t_n = nh$, where $n = 0, 1, \ldots, N-1$. Forward Euler algorithm attempts to produce approximations y_n at time points t_n which, depending on the accuracy of the algorithm, may or may not be approximately equal to the exact solution $y(t_n)$ at time points t_n.

(a) Forward Euler iteration

Derive the Forward Euler update formula and express y_n explicitly in terms of a product.

(b) Loss of monotone decay

Show that for any fixed $h > 0$, there exists an index k such that the Euler factor

$$1 - h(kh)^2$$

becomes negative. Explain the consequence for the numerical solution and contrast it with the exact solution.

(c) Exact step-to-step amplification

Compute the exact amplification factor,

$$\frac{y(t_{n+1})}{y(t_n)}$$

and compare it with the Forward Euler amplification factor at step n. Comment on how they differ as n increases.

(d) Local truncation error (single-step error)

Assume the exact value $y(t_n)$ is known. The *local truncation error* incurred on the step from t_n to t_{n+1} is defined as the difference between the exact solution and a single Forward Euler step starting from the exact value at t_n,

$$\tau_{n+1} := y(t_{n+1}) - \underbrace{\big(y(t_n) + h\,f(t_n, y(t_n))\big)}_{y(t_{n+1}) \text{ computed via the Forward Euler step}}$$

(where $f(t, y) = -t^2 y$ for this problem). Compute τ_{n+1} explicitly and also determine its leading-order behavior in terms of h and n. You may use the expansion $\exp(-x) = 1 - x + \frac{x^2}{2} + \mathcal{O}(x^3)$.

(e) Error propagation (global error)

Define the *global error* at time t_n as,

$$e_n := y(t_n) - y_n,$$

that is, the difference between the exact solution and the numerical approximation at the same time step. Derive a recursion for e_{n+1} in terms of e_n and τ_{n+1}, and explain why numerical errors may grow even though the exact solution is decaying.

Remark (local vs. global error): *The local truncation error measures the error introduced by a* single time step *of the numerical method, assuming the exact solution is known at the previous time step. In contrast, the global error measures the difference between the numerical solution and the exact solution at a given time, after errors from all previous steps have accumulated.*

(f) Interpretation

Briefly explain what this example reveals about the limitations of the Forward Euler method for long-time integration of stable ODEs with time-dependent coefficients.

Problem 4: Stability Bound for the Explicit Midpoint Method

In this problem, you will derive the stability bound (step-size restriction) for the *explicit midpoint method* applied to the exponential decay model (as discussed in the lectures),

$$x'(t) = -\lambda x(t), \qquad \lambda > 0.$$

Recall the explicit midpoint method (also known as the 2-stage Runge–Kutta method or RK2),

$$x_{n+\frac{1}{2}} = x_n + \frac{h}{2} f(t_n, x_n), \qquad x_{n+1} = x_n + h\, f\left(t_n + \frac{h}{2}, x_{n+\frac{1}{2}}\right),$$

where $f(t, x) = -\lambda x$.

(a) Derive the amplification factor

Derive the explicit midpoint update for this ODE in the form,

$$x_{n+1} = g_{\mathrm{EM}}\, x_n,$$

and show that the amplification factor is,

$$g_{\mathrm{EM}}(z) = 1 - z + \frac{z^2}{2}, \qquad \text{where } z := h\lambda.$$

(b) Solve the stability inequality

For a decaying solution, we require the method to be *stable* in the sense that,

$$|g_{\mathrm{EM}}(z)| \leq 1.$$

Solve the inequality,

$$\left|1 - z + \frac{z^2}{2}\right| \leq 1$$

explicitly for real $z \geq 0$, and hence obtain the stability bound on the step size h in terms of λ.

(c) Interpret the result

Using your result from parts (a)–(b) for the explicit midpoint method and the known stability bounds on h for Forward Euler as applied to exponential decay model (discussed in the lectures), compare the two methods in terms of their stability ranges for h and determine whether the maximum stable step size $h_{\max}$ is the same or different. Briefly discuss how their different amplification factors influence the qualitative behavior of the numerical solution within the stable regime.

Problem 5: Backward Euler – Analytical Updates and Stability

In this problem, you will analyze the *Backward Euler* method applied to initial value problems (IVPs). Unlike Forward Euler, Backward Euler evaluates the right-hand side at the future time step,

$$y_{n+1} = y_n + h\, f(t_{n+1}, y_{n+1}).$$

For the ODEs in this problem, the Backward Euler update can be solved *exactly* for y_{n+1} using algebra (no iterative solvers such as Newton's method are needed).

(a) Variable coefficient linear decay

Consider,

$$y'(t) = -t\, y(t), \qquad y(0) = y_0.$$

1. Derive the Backward Euler update formula for y_{n+1} in terms of y_n, h, and t_{n+1}.
2. Write y_n as a product involving the time grid points t_k.
3. Briefly comment on whether Backward Euler can change the sign of the solution when $y_0 > 0$.

(b) A nonlinear ODE with an explicit Backward Euler update

Consider the nonlinear IVP,

$$y'(t) = -y(t)^2, \qquad y(0) = y_0 > 0.$$

1. Derive the Backward Euler update formula for y_{n+1} in terms of y_n, h, and t_{n+1}.
2. Show that this equation is a quadratic in y_{n+1} and solve it explicitly for y_{n+1} in terms of y_n and h.
3. Select the physically meaningful branch (the one that keeps $y_{n+1} > 0$) of the quadratic equation and explain your choice briefly.

Problem 6: A Linear System of ODEs – Coupling and Stability

In this problem, you will analyze a coupled system of ordinary differential equations. Such systems arise frequently in geophysics, for example in coupled reservoirs, energy balance models, or interacting chemical species. Consider the linear system,

$$\begin{cases} x'(t) = -\alpha x(t) + \beta y(t), \\ y'(t) = -\beta x(t) - \alpha y(t), \end{cases} \qquad \alpha > 0,\ \beta > 0,$$

with initial condition $(x(0), y(0)) = (x_0, y_0)$.

(a) Matrix form

Write the system in matrix form,

$$\mathbf{u}'(t) = A\,\mathbf{u}(t), \qquad \mathbf{u}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix},$$

and explicitly state the matrix A.

(b) Eigenvalues and qualitative behavior

Compute the eigenvalues of A. Based on the eigenvalues, describe the qualitative behavior of the exact solution (decay, oscillation, or both).

(c) Exact solution structure

Without computing the full closed-form solution, show that the quantity,

$$r(t)^2 := x(t)^2 + y(t)^2$$

satisfies a scalar ODE. Solve this ODE and describe how the magnitude of the solution evolves in time.
Note: A scalar ODE is an ODE involving a single unknown function of one variable (as we discussed in class until lecture 06), as opposed to a system of ODEs that may involve multiple coupled unknown functions.

(d) Forward Euler applied to the system

Apply the Forward Euler method with step size $h > 0$ to the system and write the update in matrix form,

$$\mathbf{u}_{n+1} = B_{\mathrm{FE}}\,\mathbf{u}_n.$$

Determine the matrix B_{FE}.

(e) Stability of the Forward Euler discretization

Using the eigenvalues of A, determine the condition on h under which the Forward Euler method is stable. Briefly explain how this condition relates to the stability analysis for scalar ODEs.

Problem 7: Learning Dynamics of Coupled ODEs via the Lotka–Volterra Model (Jupyter Notebook)

In this problem, you will investigate the numerical solution of a coupled, nonlinear system of ODEs using the time-stepping methods introduced in class. Unlike earlier problems that focused on analytical derivations, this problem is designed as a computational mini-project to build intuition through simulation and visualization.

Work entirely within the **provided Jupyter notebook**. You will implement Forward Euler and the Explicit Midpoint method for the Lotka–Volterra predator–prey model. You will explore how step size and method choice affect numerical stability, accuracy, and long-time behavior, and you will visualize the dynamics using time-series plots and phase portraits.

Throughout the notebook, you will be asked to interpret numerical results and compare methods rather than perform new theoretical derivations. The goal is to understand how numerical discretization influences the behavior of nonlinear, coupled systems and to connect the analytical concepts from earlier in the course with practical computational modeling. Happy Pythoning 🐍!