

Homework is due to Canvas by 11:00pm PDT on the due date.

To submit, see <https://canvas.uw.edu/courses/1038268/assignments/3220198>

Problem 1

Prove that the ODE

$$u'(t) = \cos(t^2 + u(t)^2), \quad \text{for } t \geq 0$$

has a unique solution for all time from any initial value $u(0) = \eta$.

Problem 2

- (a) Use Duhamel's Principle to solve the ODE

$$u'(t) = \lambda(u(t) - \cos(t)) - \sin(t) \tag{1}$$

with initial condition $u(t_0) = \eta$, for general real numbers λ and η .

- (b) Let $\lambda = -0.5$. On a single graph, use Python to plot $u(t)$ (on the time interval $0 \leq t \leq 4\pi$) for different choices of initial condition $u(0) = \eta = -1, -0.8, -0.6, \dots, 1.6, 1.8, 2$.
- (c) Do the same for $\lambda = -5$.

Problem 3

Consider the third-order ODE

$$\begin{aligned} v'''(t) - v''(t) - 2v'(t) &= 0 \\ v(0) = 12, \quad v'(0) = -8, \quad v''(0) &= 14. \end{aligned}$$

- (a) Solve this equation by using the fact that a linear ODE of this form has solutions of the form $v(t) = e^{rt}$ for certain values of r . Plugging this Ansatz into the equation shows that r must be a root of a cubic equation. There are three distinct roots and hence three linearly independent solutions of this form. Find the proper linear combination of these to find the solution that also satisfies the three initial conditions.
- (b) Solve this equation in a different way: rewrite it as a first-order system of three equations of the form $u'(t) = Au(t)$ where $u(t) \in \mathbb{R}^3$ and A is a 3×3 matrix, with suitable initial conditions $u(0) = \eta \in \mathbb{R}^3$. Then compute the matrix exponential based on the eigenvalues and eigenvectors of A in order to find $u(t) = \exp(At)\eta$. (See Section D.3 in the text.) Confirm that the solution agrees with what you got before.
- (c) Remind yourself what the “companion matrix” for a polynomial is, and say why this is relevant to relating the two solution techniques above. See Section D.2.1 in the text for a discussion of a similar technique for solving linear difference equations that we will use to analyze certain numerical methods.

- (d) Determine the best possible Lipschitz constant for this system in the max-norm $\|\cdot\|_\infty$ and the 1-norm $\|\cdot\|_1$. (See Appendix A.3.)
- (e) Solve the first-order system you derived above using the Python function `odeint` from the `scipy.integrate` module, with output times `t = linspace(0,2,51)`. Plot this solution as points on top of the true solution you computed above to show that you have done this correctly.

Problem 4

Compute the leading term in the local truncation error of the following methods:

- (a) the trapezoidal method (5.22),
- (b) the 2-step Adams-Bashforth method,
- (c) the Runge-Kutta method (5.32).

Problem 5

Determine the coefficients $\beta_0, \beta_1, \beta_2$ for the third order, 2-step Adams-Moulton method. Do this in two different ways:

- (a) Using the expression for the local truncation error in Section 5.9.1,
- (b) Using the relation

$$u(t_{n+2}) = u(t_{n+1}) + \int_{t_{n+1}}^{t_{n+2}} f(u(s)) ds.$$

Interpolate a quadratic polynomial $p(t)$ through the three values $f(U^n)$, $f(U^{n+1})$ and $f(U^{n+2})$ and then integrate this polynomial exactly to obtain the formula. The coefficients of the polynomial will depend on the three values $f(U^{n+j})$. It's easiest to use the "Newton form" of the interpolating polynomial and consider the three times $t_n = -k$, $t_{n+1} = 0$, and $t_{n+2} = k$ so that $p(t)$ has the form

$$p(t) = A + B(t+k) + C(t+k)t$$

where A , B , and C are the appropriate divided differences based on the data. Then integrate from 0 to k . (The method has the same coefficients at any time, so this is valid.)