

Homework 4, due Thu Nov 14

1. **Low-Storage Runge–Kutta:** Consider the ODE

$$\frac{dV}{dt} = \eta V,$$

where η is a coefficient, and $V(0) = V_0$ is the initial condition.

- (a) For large computations of practical use, Runge–Kutta methods known as *low-storage*-Runge–Kutta (LSRK) are very often used. One example is given by the following 3rd-order formula

$$V^{n+1} = V^{**} + 8k_3/15,$$

where

$$\begin{aligned} k_1 &= \delta t \eta V^n; & V^* &= V^n + k_1/3, \\ k_2 &= \delta t \eta V^* - 5k_1/9; & V^{**} &= V^* + 15k_2/16, \\ k_3 &= \delta t \eta V^{**} - 153k_2/128, \end{aligned}$$

Show why this scheme requires less storage than the regular 3rd-order Runge–Kutta:

$$V^{n+1} = V^n + \frac{1}{6}(k_1 + 4k_2 + k_3),$$

where

$$k_1 = \delta t \eta V^n; k_2 = \delta t \eta (V^n + k_1/2); k_3 = \delta t \eta (V^n - k_1 + 2k_2).$$

Take $V_0 = 1, \eta = -1, t = [0, 20]$, and implement the LSRK3-scheme and compare the relative errors of Euler and LSRK3 using the two time steps $\delta t = 2.0$ and $\delta t = 0.05$ s.

- (b) Now let's consider the RK2 scheme we derived in class (or any RK2 scheme you have to hand). Write down a single RK2 step for the equation

$$\frac{dV}{dt} = i\kappa V,$$

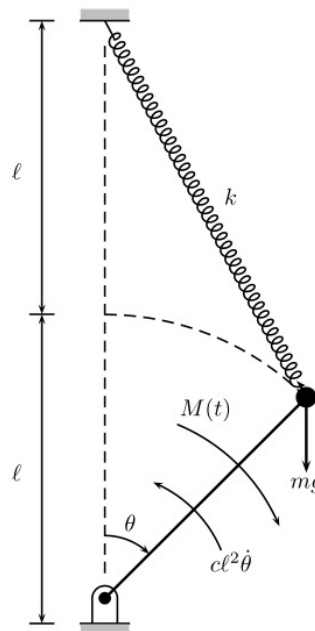
where κ is a real valued constant that represents the frequency of the oscillation and $i = \sqrt{-1}$ is the imaginary number. Derive, from the expression that you have just written, the amplification factor σ and $|\sigma|^2$.

For the next two questions we will be using MATLAB's `ode45` function. It is MATLAB's standard solver for ordinary differential equations. The function implements a Runge–Kutta method with a variable time step to solve

$$y' = f(t, y), y'(t_0) = y_0, t \in [t_0, t_f]$$

By calling `[t,y]=ode45(odefun,tspan,y0);`, where `odefun` is the function $f(t, y)$, `tspan` is a 2 element vector holding t_0, t_f , and `y0` is the initial value. See MATLAB's documentation for further information.

2. **Inverted pendulum:** Consider the problem of simulating forced oscillations of an inverted pendulum shown in the figure below:



The setup consists of a weightless rigid rod of length l that has a mass m attached to its end. Attached to the mass is a spring with stiffness constant k and an unstretched length of γl . The spring has length l when the pendulum is in the vertical position. External moments acting on the system include a driving moment $M(t)$, moment due to the gravitational force, and a viscous damping moment $cl^2\dot{\theta}$. The nonlinear second order ODE governing the equation of motion of the system can be written using dimensionless variables as follows:

$$\frac{d^2\theta}{d\tau^2} = -\alpha \frac{d\theta}{d\tau} + \sin(\theta) + P(\tau) - \beta \sin(\theta) \left(1 - \frac{\gamma}{\lambda}\right), \quad (1)$$

where

$$\begin{aligned}\tau &= \text{dimensionless time,} \\ \alpha &= \frac{c}{m} \sqrt{\frac{l}{g}}, \text{ viscous damping factor,} \\ \beta &= \frac{2k}{m} \left(\frac{g}{l}\right), \text{ spring stiffness factor,} \\ \lambda &= \sqrt{5 - 4 \cos(\theta)}, \\ \gamma &= (\text{unstretched spring length})/l, \\ P(\tau) &= \frac{M(\tau)}{mgl}, \text{ dimensionless driving moment.}\end{aligned}$$

$\theta_{\text{exact}}(\tau) = \theta_0 \sin(\omega\tau)$ is an exact solution of the ODE satisfying initial conditions $\theta(0) = 0$, $\frac{d\theta}{d\tau}(0) = \theta_0\omega$, where θ_0 and ω , are constants. For this solution, the corresponding dimensionless driving moment $P(\tau)$ takes the following form:

$$\begin{aligned}P(\tau) &= \frac{d^2\theta_{\text{exact}}}{d\tau^2} + \alpha \frac{d\theta_{\text{exact}}}{d\tau} - \sin(\theta_{\text{exact}}) + \beta \sin(\theta_{\text{exact}}) \left[1 - \frac{\gamma}{\sqrt{5 - 4 \cos(\theta_{\text{exact}})}} \right] \\ &= -\omega^2\theta_0 \sin(\omega\tau) + \alpha\omega\theta_0 \cos(\omega\tau) - \sin(\theta_0 \sin(\omega\tau)) \\ &\quad + \beta \sin(\theta_0 \sin(\omega\tau)) \left[1 - \frac{\gamma}{\sqrt{5 - 4 \cos(\theta_0 \sin(\omega\tau))}} \right].\end{aligned}$$

- (a) Using the parameters $\theta_0 = \pi/8, \omega = 0.5, \alpha = 0.1, \beta = 2$, and $\gamma = 0.5$, solve equation (1) for $\theta(\tau)$, $0 \leq \tau \leq 250$ using Matlab's `ode45` command. Experiment with several different values of input tolerance for the `ode45` command. From what input tolerance (10^7 approximately) onward do you see very little variation in the solution? You may use the error measure $\|\theta_{\text{numerical}}(\tau) - \theta_{\text{exact}}(\tau)\|_2 / \|\theta_{\text{exact}}(\tau)\|_2$ to answer this question
- (b) We would like to apply the RK4 method to equation (1) and solve for $\theta(\tau)$. In order to compute a stable step size for the RK4 method, first linearise equation (1) by assuming small oscillations ($\sin \theta \approx \theta$). Then, in the resulting second order linear ODE, use the substitutions $z_1 = \theta, z_2 = d\theta/d\tau$ to derive a system of first order ODEs of the form $d\mathbf{z}/d\tau = \mathbf{A}\mathbf{z} + \mathbf{b}$, where $\mathbf{z} = [z_1 \ z_2]^T$. What are the entries of matrix \mathbf{A} and vector \mathbf{b} ? Using the same parameters from the previous part, compute a suitable step size for the RK4 method by computing the eigenvalues of matrix \mathbf{A} . You may slightly adjust the step size (conservatively) such that the end time ($\tau = 250$) is a multiple of it. Construct a RK4 solver and solve for $\theta(\tau)$ using the computed stable step size. Do you observe a stable solution? In order to see this, make two figures: one containing a plot of the error $|\theta_{\text{numerical}}(\tau) - \theta_{\text{exact}}(\tau)|$ for $0 \leq \tau \leq 250$ and the other containing a plot of

$$\frac{\theta(\tau)}{\theta_0} \quad \text{w.r.t.} \quad \frac{1}{(\theta_0\omega)} \frac{d\theta}{d\tau}.$$

If the solution is stable, you should see approximately a unit circle in the second figure, since the exact solution $\theta_{\text{exact}}(\tau) = \theta_0 \sin(\omega\tau)$ satisfies

$$\theta_{\text{exact}}(\tau)^2 + \frac{1}{\omega^2} \left(\frac{d\theta_{\text{exact}}}{d\tau} \right)^2 = \theta_0^2.$$

If you do not observe a stable solution, reduce the step size (say by halving it) and check when you start observing a stable solution. Include an error plot and a phase plot as explained above for this step size. How does this step size compare to the estimate that was derived using linearisation? Compare to the step size taken by `ode45`.

3. **Chemical reactions:** Chemical reactions often give rise to stiff systems of coupled rate equations. Consider the following rate equations:

$$\begin{aligned} \frac{dC_1}{dt} &= -k_1 C_1 + k_2 C_2 C_3, \\ \frac{dC_2}{dt} &= k_1 C_1 - k_2 C_2 C_3 - 2k_3 C_2^2, \\ \frac{dC_3}{dt} &= 2k_3 C_2^2, \end{aligned}$$

where C_1, C_2 and C_3 are the concentrations and k_1, k_2 and k_3 are reaction rate constants given as:

$$k_1 = 0.04 \quad k_2 = 10.0 \quad k_3 = 1.5 \times 10^3.$$

Initially, $C_1(0) = 0.9$, $C_2(0) = 0.1$, and $C_3(0) = 0$.

- What is the analytical steady state solution? Note that these equations should conserve mass; that is, $C_1 + C_2 + C_3 = 1$.
- To ascertain if the problem is stiff or not we consider linearising the system about $t = 0$. Write down the 3×3 Jacobian matrix that results from this and compute its eigenvalues. Is the problem stiff?
- Solve the given system to a steady state solution ($t = 3000$ represents steady state in this problem) using:
 - MATLAB's `ode45` solver.
 - MATLAB's stiff solver `ode23s`.

Make two plots; one for each solver containing a log-log plot of the concentrations C_i vs time. Compare and comment on the computational times.

4. **Advection equation:** Consider the advection equation $u_t + cu_x = 0, c > 0$. We will study the effect of different spatial discretisation schemes on the numerical solution by studying the associated modified wavenumber.
- As in modified wavenumber analysis, assume a solution of the form $u(x, t) = \phi(t)e^{ikx}$. Substitute $u(x, t) = \phi(t)e^{ikx}$ in the advection equation, solve the resulting ODE analytically and show that the solution at x_j is:

$$u(x_j, t) \equiv u_j(t) = \phi(0)e^{ikx_j} e^{-ickt}. \quad (2)$$

- (b) Consider a semi-discretisation of the advection equation:

$$\frac{\partial u_j}{\partial t} = -c \frac{\partial u_j}{\partial x}, j = 1, 2, \dots, \quad (3)$$

where $\partial/\partial x$ is the spatial discretisation operator. Substitute $u(x, t) = \phi(t)e^{ikx}$ in the semi-discrete equation (3), and show that the analytical solution of the semi-discrete equation at x_j is:

$$u_j(t) = \phi(0)e^{ikx_j}e^{-ick't}, \quad (4)$$

where k' is the modified wavenumber associated with the spatial discretisation scheme. The difference between k and k' will reveal how the choice of $\partial/\partial x$ affects the phase and amplitude of the computed solution.

- (c) For $\partial/\partial x$, consider the central difference, first order upwind, and second order upwind schemes. For each scheme, show that the corresponding modified wavenumber satisfies:

$$k' = k + C_1 h^p + iC_2 h^q, \quad (5)$$

where C_1 and C_2 are the coefficients of the leading error real and imaginary terms, respectively. Identify C_1 , C_2 , p , and q in each case.

- (d) For each choice of $\partial/\partial x$ above, replace k' in equation (4) by the corresponding expression in (5), and comment on whether the modified wavenumber k' contributes to both phase and amplitude errors.

- (e) Let us confirm our analysis by applying the methods to the following problem:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0, 0 \leq x \leq 10, \quad (6)$$

$$u(0, t) = 0, \quad (7)$$

$$u(x, 0) = \begin{cases} \cos^2(\pi x) - \cos(\pi x), & 0 \leq x \leq 2 \\ 0, & 2 < x \leq 10. \end{cases} \quad (8)$$

Let $c = 0.8$. The exact solution is:

$$u(x, t) = \begin{cases} \cos^2(\pi(x - ct)) - \cos(\pi(x - ct)), & 0 \leq (x - ct) \leq 2 \\ 0, & 2 < \text{otherwise.} \end{cases} \quad (9)$$

Using $h = 10/150$ and $\delta t = 1/240$, solve the problem for $0 \leq t \leq 8$ using explicit Euler time stepping and central difference/first-order upwind/second-order upwind for spatial discretisation.

Note that the explicit Euler with central difference is unstable for this problem. Nevertheless, we can see the effect of the spatial discretisation for short time periods before the instability shows up.

Make three figures, one for each method: each figure should contain two subplots showing the numerical and exact solutions for $t = 4$ and $t = 8$. Comment on what you observe.

- (f) Can we reduce the numerical dispersion observed above by grid refinement? Check this by repeating part (e) with $h = 10/600$ and $\delta t = 1/960$.