Problem Set 9

Due MARCH 19th, 2021, at 5.00 PM PST

Important:

- Write your name as well as your SUNet ID on your assignment. Please number your problems.
- Submit both results and your code. You will not be graded on programming style, but clearly written code with comments may help you get more partial credit if your answer is incorrect. You can utilize MATLAB's Publish feature to produce a PDF for submission.
- Give complete answers. Do not just give the final answer; instead show steps you went through to get there and explain what you are doing. Do not leave out critical intermediate steps.
- This assignment must be submitted electronically through Gradescope by Friday, March 19, 5:00 pm PST.
- Recommended Reading: Ch. 16.1, 16.2, 16.3, 16.5 (initial value ODEs)

1. (25 pts) DAMPED HARMONIC OSCILLATOR

Consider solving the initial value problem for the displacement $u(t)$ of a damped harmonic oscillator,

$$\ddot{u} + 2\zeta\dot{u} + u = 0, \quad u(0) = u_0, \quad \dot{u}(0) = v_0, \tag{1}$$

using forward Euler with constant time step Δt. Determine the maximum stable time step, $\Delta t_{\max}$, assuming $0 < \zeta < 1$.

2. (35 pts) SHOOTING METHOD

Consider this boundary value problem for $u(x)$:

$$\frac{d}{dx}\left[m(x)\frac{du}{dx}\right] - k^2 m(x)u(x) = 0, \quad 0 < x < \infty, \tag{2}$$

where $k = 3$ and

$$m(x) = \begin{cases} 0.5 + 0.4\sin(12\pi x), & 0 < x < 1, \\ 1, & 1 < x < \infty, \end{cases} \tag{3}$$

subject to boundary conditions

$$\begin{aligned} u(0) &= 1, & (4)\\ u(x) &\to 0 \text{ as } x \to \infty. & (5) \end{aligned}$$

Both $u(x)$ and $m(x)du/dx$ are continuous, even across the discontinuity in $m(x)$ at $x = 1$.

(a) (10 pt) Show that $u(x) = Ae^{-kx}$ for any constant A is a solution to (2) on $1 < x < \infty$ that satisfies the boundary condition (5).

(b) (25 pt) Set up a shooting problem to determine $u(x)$ by expressing (2) in the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, t), \tag{6}$$

for some vector $\mathbf{y}(t)$ and independent variable t. Solve this as an initial value problem using an initial value ODE solver (i.e., your own implementation of one of the methods discussed in class). Write the initial condition $\mathbf{y}(0)$ in terms of one known value and one unknown value α. Write down an appropriate nonlinear function $g(\alpha)$, involving your result from part (a), such that the solution $\alpha = \alpha^*$ to $g(\alpha) = 0$ solves the original boundary value problem. Use a nonlinear solver (e.g., bisection, secant, Newton, or your hybrid method from HW1) to find α^* and solve the boundary value problem. *Hint: Shoot over the interval* $0 < x < 1$ *only.* Plot $u(x)$ for $0 < x < 2$ (not just $0 < x < 1$).

3. (40 pts) ADAPTIVE TIME STEPPING

The oscillations of a gas-filled bubble in water are described by the Rayleigh equation,

$$R\ddot{R} + \frac{3}{2}\dot{R}^2 = \frac{p_w}{\rho_w}\left[\left(\frac{R_{eq}}{R}\right)^{3\gamma} - 1\right], \tag{7}$$

where t is time, $R(t)$ is the bubble radius, γ is the adiabatic exponent of the gas, p_w is the ambient pressure in the water, ρ_w is the water density, and R_{eq} is the equilibrium bubble radius at pressure p_w. Our objective is to solve for $R(t)$ for $0 < t < t_{\max}$ with initial conditions $R(0) = R_0$, $\dot{R}(0) = \dot{R}_0$. The nonlinearity of this problem makes it very challenging to solve with constant time steps, motivating adaptive time-stepping in which the time step Δt is selected to maintain local error slightly smaller than a tolerance tol. When solving this problem, use SI units and the following parameter values: $\gamma = 1.4$, $p_w = 10^5$ Pa, $\rho_w = 10^3$ kg/m^3, $R_{eq} = 1$ m, $t_{\max} = 1$ s, $R_0 = 0.1$ m, $\dot{R}_0 = 0$, and tol $= 10^{-6}$.

When solving an ODE system of the form $d\mathbf{y}/dt = \mathbf{f}(\mathbf{y}, t)$, the local error in the solution vector $\mathbf{y}$ after time step n is the difference between the numerical solution $\mathbf{y}_{n+1}$ and the exact solution to the ODE when solved over just one time step, both starting from the initial condition $\mathbf{y}_n$ at time $t = t_n$. (This makes local error different from global error; unlike global error, local error does not depend on the global error of the previous time steps compounded with the growth factor.) We will use the three-stage explicit Runge-Kutta method

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{y}_n, t_n), \tag{8}$$

$$\mathbf{k}_2 = f\left(\mathbf{y}_n + \frac{\Delta t}{2}\mathbf{k}_1, t_n + \frac{\Delta t}{2}\right), \tag{9}$$

$$\mathbf{k}_3 = f(\mathbf{y}_n + \Delta t(-\mathbf{k}_1 + 2\mathbf{k}_2), t_n + \Delta t), \tag{10}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{2}(\mathbf{k}_1 + \mathbf{k}_3), \tag{11}$$

$$\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{6}(\mathbf{k}_1 + 4\mathbf{k}_2 + \mathbf{k}_3), \tag{12}$$

where $\mathbf{y}_{n+1}$ is a 2nd order accurate update and $\hat{\mathbf{y}}_{n+1}$ is a 3rd order accurate update. Their difference, $\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1}$, is an estimate of the local error of the 2nd order update.

(a) (5 pt) Rewrite (7) as a system of first-order differential equations. Solve it using the 4th order explicit Runge-Kutta method given in the notes with constant time step. Approximately how many time steps are required to obtain a solution that is visually identical to a more accurate solution obtained using smaller Δt and more time steps? Plot the solution $R(t)$.

(b) (5 pt) Show that for the three-stage method given above using time step Δt, the L_2 norm of the local error estimate $\ell(\Delta t) = ||\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1}||_2 \approx K\Delta t^3$ for some constant $K > 0$ (you don't need to find K, just explain why it exists).

(c) (5 pt) Show that the time step

$$\Delta t' = \Delta t \left(\frac{\mathrm{tol}}{\ell(\Delta t)} \right)^{1/3} \tag{13}$$

will provide a solution with local error approximately equal to tol.

(d) (25 pt) Implement the following adaptive time-stepping method: Take one time step using some Δt. Estimate the local error. If the local error is greater than the tolerance, reject that update (i.e., don't update the solution) and attempt the time step again, but using the smaller time step

$$\Delta t' = 0.9\Delta t \left(\frac{\mathrm{tol}}{\ell(\Delta t)} \right)^{1/3}. \tag{14}$$

This is the same as (13) except for a "safety factor" of 0.9 that decreases the number of rejected steps. If the local error is less than the tolerance, update the solution using the 3rd order accurate update $\hat{\mathbf{y}}_{n+1}$. Select the next time step using (14). Use $\Delta t = 10^{-7}$ s for the first time step to start this method from $t = 0$. Apply this method to solve (7). Plot $R(t)$ on the same graph on your previously calculated $R(t)$. Report the number of time steps used. Plot $\ell(t)$ and verify that the local error is less than the tolerance. Plot time step Δt vs. time step number, using a logarithmic axis for Δt. Explain why this method is more efficient than the method in part (a).

Background: The Rayleigh equation is derived by as the spherically symmetric solution to the conversation of mass and momentum of an incompressible fluid surrounding a spherical bubble having uniform pressure p_b. The gas inside the bubble is an ideal gas, expanding and contracting under adiabatic conditions (no heat exchange with the fluid), such that $p_b V_b^{\gamma} = p_\infty V_{eq}^{\gamma}$, where the bubble volume is $V_b = (4\pi/3)R^3$. When the bubble pressure equals the ambient pressure in the fluid, $p_b = p_\infty$, then the bubble attains its equilibrium volume $V_{eq} = (4\pi/3)R_{eq}^3$. The right side of (7) is $(p_b - p_\infty)/\rho_\infty$, which captures the imbalance of forces on the water when the bubble pressure differs from the ambient water pressure. That force imbalance drives expansion or contraction of the bubble, with inertia of the water allowing the bubble to overshoot or undershoot its equilibrium size. Watch this video at 2:30 or this other video at 4:31 to see bubble oscillations. This model is used applied to underwater explosions (from weapons or volcanoes), cavitation bubbles that damage ship propellers, medical ultrasonic treatments, light generation by bubble collapse (sonoluminescence), and pistol shrimp killing their prey.

P.S. Thanks for a fun quarter and happy computing in your career!