AM205: Assignment 3

Question 1 [26 points]

(a) [7 points] (Written question, no code required) The 4-point Newton-Cotes formula on $[-1, 1]$ is

$$Q_3(f) = w_0 f(-1) + w_1 f(-1/3) + w_2 f(1/3) + w_3 f(1).$$

Use the fact that the formula should be exact for all polynomials of degree 3 to derive a linear system for the quadrature weights that involves a 4×4 (transposed) Vandermonde matrix. What are the values of the weights w_0, w_1, w_2 and w_3?

(b) [6 points] Write functions `Q_1h = comptrap(f,a,b,m)` and `Q_2h = compsimp(f,a,b,m)` that implement the composite trapezoid rule and composite Simpson rule, respectively. Here `f` is a "function handle," a and b define the interval of intergration, and m is the number of subintervals we use in $[a, b]$; the number of quadrature points used in `comptrap` and `compsimp` is $m + 1$ and $2m + 1$, respectively.

Consider the "error function,"[1]

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \text{d}t.$$

Use `comptrap` and `compsimp` to approximate $\text{erf}(1)$ with n quadrature points, for $n = 3 : 2 : 101$. Compute and plot the error from `comptrap` and `compsimp` with respect to the `erf` function that is available in the computational environment that you're using. What asymptotic convergence rate in terms of h do you observe in each case?

(c) [5 points] (Written question, no code required) Given that the cubic Legendre polynomial is $P_3(x) = \frac{1}{2}x(5x^2 - 3)$, derive the 3-point Gauss quadrature rule on the interval $[-1, 1]$ by evaluating the relevant integrals by hand. Demonstrate that this quadrature rule integrates all polynomials up to the expected degree exactly.

(d) [8 points] Given an n-point quadrature rule, $\{x_1, x_2, \ldots, x_n\}$ and $\{w_1, w_2, \ldots, w_n\}$, on $[-1, 1]$, give the formula for a corresponding n^2-point quadrature rule on a grid in $[0, 1]^2$.

Use your formulation to approximate the integral

$$\int_0^1 \int_0^1 e^{-x^2 \sin y} \text{d}x \text{d}y$$

based on the 5-point Gauss quadrature rule for the interval $[-1, 1]$ provided in `gauss_quad_5pts.txt`. Also, plot the points in $[0, 1]^2$ at which your quadrature rule samples the integrand.

[1] This function is widely used in probability and statistics.

Question 2 [26 points]

(a) [12 points] Write a Matlab function `D1 = cdiff1(n,a,b)` that returns the $n \times n$ differentiation matrix for the first derivative for n evenly spaced points in the interval $[a, b]$ (where a and b are grid points) based on the centered difference formula

$$\frac{y_{i+1} - y_{i-1}}{2h}.$$

(You may assume that $n \geq 3$ when you write your function.) Make sure that your differentiation matrix is $O(h^2)$ accurate — you will have to be careful with your treatment of the boundaries of the interval to ensure that this is the case.

For $f(x) = e^x \sin(5x)$ on the interval $[-1, 1]$, plot $f'(x)$ along with your approximation of $f'(x)$ for $n = 50$. Also, for $n = 10 : 10 : 2000$, plot the infinity norm error (evaluated on the n-point grid) in the approximation to $f'(x)$ obtained by `cdiff1` and confirm, based on your plot, that you get the expected asymptotic convergence rate.

(b) [8 points] Suppose now that we wish to develop a second-derivative differentiation matrix for functions that are periodic on the interval $[a, b]$. Write a function `D2_per = cdiff2_per(n,a,b)` that returns the $(n-1) \times (n-1)$ differentiation matrix for the second derivative associated with n evenly spaced points in the interval $[a, b]$ (where a and b are grid points), based on the centered difference formula

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

For $f(x) = \sin(2\pi x)$ on the interval $[-1, 1]$, plot $f''(x)$ along with your approximation of $f''(x)$ for $n = 50$. Also, as in (a), plot the infinity norm error for $n = 10 : 10 : 2000$ in the approximation to $f''(x)$ obtained by `cdiff2_per` and confirm, based on your plot, that you get the expected asymptotic convergence rate. Also, plot and comment on the sparsity pattern of `D2_per` for $n = 20$ (use `spy(D2_per)`).

(c) [6 points] (Written question, no code required) What is the rank of the $n \times n$ differentiation matrix from (a)? What is the rank of the $(n-1) \times (n-1)$ differentiation matrix from (b)? Provide a basis for the nullspace of any matrix that you claim is rank-deficient, and explain how your answers relate to the operators that these matrices approximate.

Question 3 [18 points]

(a) [8 points] (Written question, no code required) Applying the midpoint quadrature rule (i.e. $n = 0$ Newton-Cotes with quadrature point at the midpoint) on the interval $[t_k, t_{k+1}]$ to $y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(t, y(t))\mathrm{d}t$ leads to the implicit *midpoint method*

$$y_{k+1} = y_k + hf(t_{k+1/2}, (y_k + y_{k+1})/2),$$

where $t_{k+1/2} \equiv t_k + h/2$. Use Taylor series expansions to show that the order of accuracy of this method is 2. Assume here that $y \in \mathbb{R}$ and $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. (Hint: Use the Taylor series expansion $f(x, y + h) = f(x, y) + hf_y(x, y) + O(h^2)$.)

(b) [5 points] (Written question, no code required) What is the stability region of the method from (a) for the equation $y' = \lambda y$, i.e. for what values of $\bar{h} \equiv h\lambda \in \mathbb{C}$ is the method stable?

(c) [5 points] (Written question, no code required) Van der Pol's equation

$$y'' - \varepsilon(1 - y^2)y' + y = 0$$

with $y(0) = A_1$, $y'(0) = A_2$ and a parameter $\varepsilon > 0$ models electrical circuits connected with electronic oscillators. Rewrite the equation as a coupled system of two first-order equations with appropriate initial conditions. Write down the (forward) Euler method for this system, when $\varepsilon = 1$, $A_1 = A_2 = 1/2$, for a "step size" of h. What are the Euler approximations to $y(h)$ and $y'(h)$?

Question 4 [30 points] The N-body Problem: In celestial mechanics, the N-body problem involves finding the trajectories of point masses that are gravitationally attracted to each other. Closed-form solutions are known for the $N = 2$ case, but in general the N-body problem cannot be solved exactly for $N \geq 3$. As a result, numerical approximation is the standard tool for exploring N-body systems.

Consider an N-body system in which the i^{th} body has mass $m^i \in \mathbb{R}_{>0}$ and position $q^i(t) \in \mathbb{R}^3$ at time t. The force acting on body i (denoted $F^i \in \mathbb{R}^3$) is governed by Newton's Law of Universal Gravitation:

$$F^i = G \sum_{j=1, j \neq i}^{N} \frac{m^i m^j}{(\|r^{ij}\|_2)^3} r^{ij}, \tag{1}$$

where $r^{ij} = q^j - q^i \in \mathbb{R}^3$, and $G = 6.674 \times 10^{-11} \frac{m^3}{kg \cdot s^2}$ is the gravitational constant.

The motion of each body in the system can then be determined by Newton's Second Law of Motion: $\ddot{q}^i(t) = F^i(t)/m_i$, where $\ddot{q}^i$ denotes the second-derivative of q^i, i.e. $\ddot{q}^i = \frac{\mathrm{d}^2 q^i}{\mathrm{d}t^2}$ (and similarly $\dot{q}^i$ denotes the first derivative of q^i).

In parts (a)-(e) of this question we will consider a 5-body system with initial state (i.e. at $t = 0$) defined by the data in `System-Q4.txt` (or equivalently, the data is provided in MATLAB format in `System-Q4.mat`). The data in `System-Q4.txt` is in SI units. Throughout the rest of this question we refer to this system as `System-Q4`.

(a) [3 points] Write a function `nbodyforce` that takes the masses and positions of an arbitrary number of bodies as inputs and returns the force vector due to gravity on each body. Use `nbodyforce` to calculate the force vector on each of the 5 bodies in `System-Q4` at $t = 0$.

(b) [3 points] The potential energy, U, and kinetic energy, T, of an N-body system are given by:

$$U = -G \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{m^i m^j}{\|r^{ij}\|_2}, \tag{2}$$

$$T = \frac{1}{2} \sum_{i=1}^{n} m^i \left(\|\dot{q}^i\|_2\right)^2. \tag{3}$$

The total energy of the system is then given by $E = U + T$.

Write a function `nbodyenergy` that takes the mass, position and velocity of an arbitrary number of bodies as input and returns the total energy of the system. Use `nbodyenergy` to calculate the total energy of `System-Q4` at $t = 0$.

(c) [6 points] Use the Forward Euler method to simulate the evolution of `System-Q4` for $t \in [0, t_{\text{final}}]$, where $t_{\text{final}} = 4 \times 10^7$. Use a constant time-step size of $\Delta t = 4 \times 10^3$. Plot the (x, y, z) trajectory of body 5, i.e. plot $q^5(t)$ for $t \in [0, t_{\text{final}}]$. On a separate figure, use your Forward Euler simulation results to plot the total energy of `System-Q4` as a function of time (your plot should show the of energy at each time-step).

Also, provide a script, `nbodyanimate`, which animates the evolution of the positions of all bodies in the system as a function of time for $t \in [0, t_{\text{final}}]$. You should not include any plots from this animation in your write-up, but it should be easy for the grader to run your `nbodyanimate` script in order to view your animation. (Note: In MATLAB you can call the `pause` function between plotting calls to control the speed of the animation.)

(d) [6 points] Repeat part (c) MATLAB's `ode45` function instead of Forward Euler. (You can let `ode45` choose the time steps in the interval $t \in [0, t_{\text{final}}]$ for you.)

(If you aren't using MATLAB, then you should use a similar "built-in" ODE solver that is available in the numerical environment that you are using.)

(e) [8 points] Symplectic integrators are a class of ODE solvers which are designed to conserve the total energy of a kinematic system (like the N-body problem) to a high degree of accuracy. Here we consider the Störmer-Verlet method, which is a popular symplectic integrator for N-body problems.

The Störmer-Verlet method is defined as follows:

$$q^i_{n+1} = 2q^i_n - q^i_{n-1} + \ddot{q}^i_n (\Delta t)^2, \tag{4}$$

$$\dot{q}^i_n = \frac{1}{2\Delta t}\left(q^i_{n+1} - q^i_{n-1}\right). \tag{5}$$

To use this method we also need the positions of the bodies at the first time-step (i.e. at $t = t_1$). Use the following Taylor expansion to approximate these positions:

$$q^i_1 = q^i_0 + \dot{q}^i_0 \Delta t + \frac{1}{2}\ddot{q}^i_0 (\Delta t)^2.$$

Also, note that we can't apply (5) at the final time-step. Hence instead you should use the following one-sided approximation to $\dot{q}^i$ at the final time-step:

$$\dot{q}^i_k = \frac{1}{\Delta t}\left(q^i_k - q^i_{k-1}\right).$$

Repeat part (c) using the Störmer-Verlet method instead of Forward Euler. (Use the same choice of Δt as in (c).)

(f) [4 points] `System-Q4f-perturb` specifies two initial configurations. In the second configuration, the initial position and velocity of body 3 is perturbed from the first configuration by 0.01%, and the initial states of all other bodies are unchanged. Compute the evolution of both systems in `System-Q4f-perturb` using the same solver and solver options that you used in part (d) (i.e. use `ode45`, or other "built-in" ODE solver of your choice).

Plot the trajectory of body 5 (i.e. $q^5(t)$, $t \in [0, t_{\text{final}}]$) for both systems and superimpose the two trajectories on a single plot (use different colors to distinguish the two trajectories). This plot should convince you that the N-body problem is a "chaotic system" in the sense that it exhibits sensitive dependence on initial conditions.