Problem Set 3

Assigned 10/5/23, Due 10/26/23

Problem 1 (15 points): In this problem, we will consider how we can pose and solve some variants on a continuous simulation problem. For each problem variant, you should first pose it as a system of differential equations and then present iteration formulas that would be appropriate to numerically integrate the system. If the problem description leaves out details, fill in what you need to establish a fully specified system of differential equations and appropriate iteration formulas. You do not need to discuss initial or boundary conditions in your solutions.

We will consider some variants on disease models mentioned in class and will specifically examine a variant of a (S)usceptible, (E)xposed, (I)nfected, and (R)ecovered model in which recovered people only have temporary immunity before returning to the pool of infected people. This model describes disease propagation in terms of four coupled differential equations:

$$\frac{dS}{dt} = -\lambda_1 \frac{SI}{N} + \lambda_4 R$$

$$\frac{dE}{dt} = \lambda_1 \frac{SI}{N} - \lambda_2 E$$

$$\frac{dI}{dt} = \lambda_2 E - \lambda_3 I$$

$$\frac{dR}{dt} = \lambda_3 I - \lambda_4 R$$

where $N = S + E + I + R$

a. Consider a version of this model exactly as described, where we want to model assuming all people are found in a single, well-mixed geographic region.

b. Suppose we want to take the model from part a and allow for stochastic noise in the estimates. You should assume that the standard deviation in the noise in each step is proportional to the square root of the corresponding deterministic component and that you must have conservation in the noise describing exchange between populations for example. For example, when computing noise in the transfer of Recovered people to Susceptible, any extra people created in the Susceptible set due to noise must be exactly offset by fewer people in the Recovered set due to the same noise.

c. Now suppose we want to modify the model from part a to assume instead that people can migrate around a physical space. We will assume that infected people need to rest and stay in one place but everyone else "diffuses" around a two-dimensional region.

Problem 2 (20 points): In this problem, we will apply a few steps of some numerical integration schemes to simulate a simple reaction system:

$$A + B \rightleftharpoons^{k_1}_{k_2} 2C$$

where the reaction has forward rate k_1 and reverse rate k_2. This system is described by the mass action equations

$$\frac{dA}{dt} = k_2C^2 - k_1AB$$
$$\frac{dB}{dt} = k_2C^2 - k_1AB$$
$$\frac{dC}{dt} = 2k_1AB - 2k_2C^2$$

a. We first want to simulate this system with a forward Euler method. Provide iterators for implementing the system by Forward Euler.

b. Carry out one step of forward Euler integration (taking the system from $t = 0$ to $t = 0.1$), initializing the system with 2 units of A, 1 unit of B, and 0 units of C with parameters $k_1 = 1$, $k_2 = 2$, $\Delta t = 0.1$. Show your work.

c. We next want to simulate this system with backward Euler. Provide iterators for implementing the system by backward Euler. (Hint: It will be easier to reduce this first to a system of one variable, x, representing the progress of the overall reaction, with the substitutions $A = A(0) - x$, $B = B(0) - x$, $C = C(0) + 2x$, and $x(0) = 0$. Then derive $\frac{dx}{dt}$ and the integration formula for x, which you can use to find $A(t)$, $B(t)$, and $C(t)$.)

d. Carry out one step of backward Euler integration (taking the system from $t = 0$ to $t = 0.1$) initialing the system with 2 units of A, 1 unit of B, and 0 units of C with parameters $k_1 = 1$, $k_2 = 2$, $\Delta t = 0.1$. Showing your work.

e. We then want to simulate this system with the midpoint method. Provide iterators for implementing the system by the midpoint method.

f. Carry out one step of midpoint method integration (taking the system from $t = 0$ to $t = 0.1$), initializing the system with 2 units of A, 1 unit of B, and 0 units of C with parameters $k_1 = 1$, $k_2 = 2$, $\Delta t = 0.1$. Show your work.

Problem 3 (20 points): In this problem, we will develop a numerical scheme to handle a boundary condition we have not seen before. Assume we are trying to simulate a convection-diffusion system, which we will restrict to one dimension to keep things simple:

$$\frac{\partial C}{\partial t} = v\frac{\partial C}{\partial x} + d\frac{\partial^2 C}{\partial x^2}$$

We have a Neumann boundary condition, where instead of fixing the first derivative as we have seen in class, we fix the second derivative $\frac{\partial^2 C}{\partial x^2} = 0$ at the boundary. We want to derive a second order approximation to the first spatial derivative $\frac{\partial C}{\partial x}$ at this boundary.

a. We will derive an approximation at the boundary x_0 by assuming we know $\frac{\partial^2 C}{\partial x^2}(x_0)$, $C(x_0)$, and $C(x_0 + \Delta x)$. Derive any Taylor expansions we will need to set up the numerical derivative approximation, expanded as far as you will need to derive a second order method.

b. Now set up the system of equations we need to solve to find the coefficients for the approximation formula

$$\frac{\partial C}{\partial x} \approx aC(x_0) + bC(x_0 + \Delta x) + c\frac{\partial^2 C}{\partial x^2}(x_0)$$

.

c. Solve for the coefficients and provide your boundary formula $\frac{\partial C}{\partial x} \approx aC(x_0) + bC(x_0 + \Delta x) + c\frac{\partial^2 C}{\partial x^2}(x_0)$.

d. If you solved part c correctly, you might note that when you plug in $\frac{\partial^2 C}{\partial x^2}(x_0) = 0$, the formula you derived reduces to an approximation for $\frac{\partial C}{\partial x}$ that we have seen previously, and that you were previously told was first order accurate. Yet you should have shown here that it your formula is second order accurate. Explain the discrepancy.

Problem 4 (25 points): In this question, we will explore an example of a predator-prey model as we discussed in class. As in class, we will suppose that we are studying populations of lantern flies (x) on the assumption that praying mantises (y) eat lantern flies. We observe the lantern fly population growing, but as praying mantises start eating them, the praying mantis population also begins to grow and slow the growth of the lantern flies. We want to know if the praying mantises will eventually bring lantern fly numbers under control or if they will continue to explode until some other resource limit unaccounted for by our model is reached.

We will assume the system can be explained by a *Lotka-Volterra* model, a commonly used differential equation model for predator-prey systems described by the following ordinary differential equations:

$$\frac{dx}{dt} = ax - bxy$$

$$\frac{dy}{dt} = cxy - dy$$

for some real constants a, b, c, and d. The model assumes that the prey population x grows exponentially with rate ax but is reduced by predation at rate bxy. The predator population y grows as they feed on prey with rate cxy but also has some natural death rate dy. We will try a few different ways of simulating this system.

a. Derive a numerical iterator formulas for a forward Euler implementation of this system.

b. Derive numerical iterator formulas for a backward Euler implementation of this system.

c. Solving for your backward Euler iterates will require inverting a system of two variables. It may prove impossible to do analytically so we will propose that we can solve for it by finding a zero of the system:

$$\begin{bmatrix} x_{n+1} - x_n - \Delta t f_x(x_{n+1}, y_{n+1}) \\ y_{n+1} - y_n - \Delta t f_y(x_{n+1}, y_{n+1}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where $f_x(x, y) = \frac{dx}{dt}$ and $f_y(x, y) = \frac{dy}{dt}$.

Provide pseudocode for a Newton-Raphson solver for this system, assuming (x_n, y_n) is a good initial guess as to (x_{n+1}, y_{n+1}). You can assume ten rounds of N-R is enough for the N-R to converge.

d. Now write iterators for a leapfrog integrator for this system. Note that for the leapfrog or other implicit methods, you need multiple initial points get the next one, so you need to use an explicit method to get it started. You can use one step of forward Euler to derive (x_1, y_1) from (x_0, y_0) and then run leapfrog from there to get all subsequent (x_n, y_n).

e. Implement code to run your integrators for a given step size Δt and total time T and produce plots of x and y versus time. Your code should take as input a, b, c, d, $x(0)$, $y(0)$, T, and Δt. (Hint: You might want to implement the easiest method first (FE) and see what it does before trying to debug the hardest one (BE).)

f. Simulate your systems for $a = 1$, $b = 0.25$, $c = 0.1$, $d = 1$; initial conditions $x(0) = 50$, $y(0) = 1$ in some arbitrary units; and $\Delta t = 0.01$ and $T = 100$. Provide plots of the outputs.

g. Our original question concerns the stability behavior of the system. What answer does each of the simulations suggest: lantern fly populations will blow up indefinitely, collapse and stabilize, or oscillate? Do your simulations lead to the same answer and, if not, which one do you believe most likely reflects the stability behavior the model should exhibit if you could simulate it perfectly? Justify your answer in a sentence or two.

***Problem 5 (20 points):** In this problem, we will look to develop a new numerical integration method. Our goal will be to develop a version of the leapfrog method that works for variable step sizes. That is, we will assume our previous step covered some time Δt_1 and we now want to take another step Δt_2, with the assumption that in general $\Delta t_1 \neq \Delta t_2$. Our goal will be to derive a formula for a second derivative approximation combining $x(t - \Delta t_1)$, $x(t)$, and $x'(t)$ to derive $x(t + \Delta t_2)$.

a. First, we will need Taylor expansions of the terms around t. Provide this for the $x(t - \Delta t_1)$ and $x(t + \Delta t_2)$ terms. To derive second order approximations, we will want to truncate at with fourth order error terms.

b. Now, we will want to figure out what the coefficients should be for the terms to derive a consistent formula $ax(t - \Delta t_1) + bx(t) + cx'(t) \approx x(t + \Delta t_2)$. Pose this as a linear systems problem.

c. Solve for the coefficients for the scheme to give it as much accuracy as possible.

d. What order accuracy is the method we have derived?

*Recall that the starred problems are only for the 02-712 students.