Temple Math 5044 **Introduction to Numerical Analysis II** **Spring 2023**

Problem Set 2

(Out Thu 02/02/2023, Due Thu 02/16/2023)

Problem 3

Consider again our beloved Lotka-Volterra predator-prey model

$$\frac{d}{dt}\vec{u}(t) = \vec{f}(\vec{u}(t)) \ , \tag{1}$$

where $\vec{u} = \begin{pmatrix} u \\ v \end{pmatrix}$ and $\vec{f}(\vec{u}) = \begin{pmatrix} f_1(u,v) \\ f_2(u,v) \end{pmatrix} = \begin{pmatrix} u - 4uv \\ -v + 5uv \end{pmatrix}$, and use it to perform a numerical error analysis of various numerical methods. To that end, consider $t \in [0, 31]$, and initial conditions $\vec{u}(0) = \begin{pmatrix} 0.73 \\ 0.25 \end{pmatrix}$.

(a) First, obtain a reference solution[1] for the final state $\vec{u}(31)$, by using RK4 with step size $h = 10^{-4}$ (or smaller if you like). Call this vector $\vec{u}_{\text{true}}$.

(b) For a sequence of time steps $k = 10^{\beta}$, where $\beta \in \{-3.0, -2.75, -2.5, -2.25, -2.0, -1.75, -1.5, -1.25, -1.0\}$, approximate[2] the final state $\vec{u}(31)$ by using the following numerical schemes: (i) forward Euler, (ii) RK4, (iii) Crank-Nicolson, and (iv) the two-stage Gauss-Legendre method.[3]

(c) For each of the 4×9 results for the final state that you obtained in (b), compute the error $\|\vec{u}(31) - \vec{u}_{\text{true}}\|_2$, and plot the sequence (with respect to h) of errors in four curves in log–log scale. Explain what the slopes of these curves tell you, and what their abscissae tell you.

(d) In two new figures, plot the approximate solution curves $(u(t), v(t))$, obtained with (i) RK4 and (ii) Crank-Nicolson for system (1), with $t \in [0, 30000]$ and time step size $k = 0.3$.[4] What do you observe? Explain your observations.

Problem 4

Download the Matlab program `temple5044_voyager.m` from the course web site, and run it.

(a) Describe in your own words what the program does, and what it shows.

(b) Find out how to use Matlab's `ode45.m`, and apply the solver to the given test case, using a relative tolerance of 10^{-9}.

(c) Modify the system's initial conditions correctly, so that the `ode45.m` solution matches the true Voyager 1 trajectory towards and past jupiter. Use online resources to find the true trajectory. Note that Voyager 1 swang by jupiter 546 days after its start.

(d) Now modify the `temple5044_voyager.m` code to incorporate *your own* adaptive time integrator. Use the `ode45.m` reference solution to verify that your numerical solution is reasonably accurate.

---

[1] If an analytical solution is unavailable, one can use a highly resolved computational approximation in lieu of it.

[2] Don't forget to round k so that you do an integer number of time steps.

[3] Use three Newton iteration steps per time step for the implicit schemes.

[4] Yes, you are doing a lot of time steps here. See this as a simulation of an ecological systems over a millennium time scale.

(e) On your developed test case (of a realistic swing-by), run your own adaptive integrator with three choices of tolerances TOL, as follows: (i) TOL is too large, so the solution is inaccurate; (ii) TOL is sufficiently small, so that the numerical solution is satisfactory in the "eye-norm"; and (iii) TOL is one tenth of the choice in (ii). For all three cases, plot the path of the voyager probe, and its position after 1,164 days. Where should it be, and where does it end up in the numerical solutions?

Instructions

For each problem set, you need to submit one document, either in class or via email to the course instructor, that contains plots and explanations (hand-written or typed). If you decide to email the document, name it `yourfamilyname_problemset1.pdf`, where 1 stands for the number of the problem set.

In addition, for each programming task, email your respective program to the course instructor, under the filename `yourfamilyname_problem1a.m`, where `1` stands for the problem number and `a` for the sub-problem letter.