Physics 305 – Computational Physics, Fall 2020
Term Project
Full project submission Due Date: Tuesday December 15, 5pm
Presentation Phase: November 30 - December 11

The program in your term project can be either submitted as a python program or ipython notebook, where the latter is preferred. The program, an explanation of what the program does, along with answers to **all** questions asked should be uploaded to d2l.

You are expected to write a term paper (in word or Latex) on your project that discusses the problem you are trying to solve, the basic equations that govern the problem, includes plots that show the solutions, and describes the solution and the numerical method involved. In addition, you must demonstrate that your solution is correct by showing that the code converges at the expected order. If your code does not converge at the expected order you should try to identify potential reasons for why this is the case. You are expected to work on your term project by yourself..

Your term project will receive full credit **only** if: (a) the program runs successfully without errors using python 3, (b) the programs have explanatory comments and variable names that identify with the problem equations you are trying to solve, (c) give the correct output, and (d) demonstrate the validity of the solution through convergence plots. No credit will be given to late term projects.

The term paper is as important as the code (50% of the term project credit will go to the code and the other 50% to the paper). Answers to the questions and analysis requested below should be elaborated in the report. Plots should be clearly labeled and be properly described in the report, and not just shown. You will need to explain what each and every plot demonstrates. A polished paper written in word or LaTex (preferred, e.g. please try overleaf) is expected to get full credit.

Note: Before you present results from numerical integrations that answer the questions in the project, it is critical to *first* perform the convergence tests for one case, and to estimate errors. This will tell you how small a step size is necessary for accurate solutions. Only after errors are estimated, does it make sense to run your code for producing results that help you learn more about the system you study.

I. THE LORENZ ATTRACTOR: THE "BUTTERFLY" EFFECT AND CHAOS

You will be examining the properties of the Lorenz equations. The Lorenz equations (after Edward Lorenz 1963) provide a simplified mathematical model for atmospheric convection. The equations of motion (EOM) are

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x)\\ \frac{dy}{dt} &= x(\rho-z)-y\\ \frac{dz}{dt} &= xy-\beta z\end{aligned} \tag{1}$$

The equations describe the evolution of the properties of a 2D fluid layer uniformly warmed from below and cooled from above. Here, x is proportional to the rate of convection, y is proportional to the horizontal temperature variation, and z proportional to the vertical temperature variation. The constants σ, ρ, β are parameters proportional to the Prandtl number, Rayleigh number (important parameters for fluids), and the physical dimension of the fluid layer. Notice that for $\sigma > 0$, if we neglect y in the first equation it becomes an equation that tries to damp x (i.e. the rate of convection). The second equation also has a term trying to damp y (the $-y$ term in the right-hand-side) with coeficient -1. Similarly, the third equation term $-\beta z$ is trying to damp z.

It turns out that when $\rho < 1$ there is only one equilibrium point $x = y = z = 0$ which is an attractor. For $\rho > 1$ there are two critical (or equilibrium) points (for which $dx/dt = dy/dt = dz/dt = 0$): $(x, y, z) = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1$ and $(x, y, z) = (-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho-1$. These points are stable only if

$$\rho < \sigma\frac{\sigma+\beta+3}{\sigma-\beta-1}. \tag{2}$$

The equations of motion (1) have multiple time scales involved, and hence non-dimensionalizing the system completely is not possible. Instead you will be experimenting with different values for the parameter ρ for $\sigma = 10$, and $\beta = 8/3$.

1. Use RK4 to integrate numerically the system of ODEs (1) from $\tilde{t} = 0$ and for sufficiently long times. You will need to be plotting x, y, z to see if the orbits are converging or diverging to determine when to stop the integration. You will need initial conditions to integrate the equations.

   - First test that the origin is an attractor. Choose $\rho = 0.5$ and evolve the system. Choose $x = y = z = 1$ at $t = 0$. Does the solution converge to the origin $x = y = z = 0$? Show a plot of the orbits that demonstrate this.
   - Now test that the points $(x_1, y_1, z_1) = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho{-}1$ or $(x_1, y_1, z_1) = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1$ are attractors. For them to be attractors we must satisfy Eq. (2), i.e., $1 < \rho < 470/19 \approx 24.7368$. Choose a value for ρ that satisfies the inequalities and evolve the system with $x = y = z = 1$ at $t = 0$. Does it converge to one of the two equilibrium points? Show a plot of the orbits in 3D that demonstrate this. Also show a plot of the distance of the solution from these two points $L_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}, i = 1, 2$ vs time.
   - Now test that the point $(x_0, y_0, z_0) = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1$ is unstable if $\rho > 470/19 \approx 24.7368$. Choose a value for $\rho = 28$ that satisfies the previous inequality. Evolve the system with $(x, y, z) = (\sqrt{\beta(\rho-1)} + \epsilon, \sqrt{\beta(\rho-1)} + \epsilon, \rho - 1 + \epsilon$ at $t = 0$ with $\epsilon = 1e-5$. Is the point $(x_0, y_0, z_0) = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1$ unstable, i.e., does the solution move away from it? Show a plot of the orbits in 3D that demonstrate this (evolve long enough to show the "butterfly plot"). Show also a plot showing the distance from this point versus time, i.e., $L = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}$.
   - Next demonstrate sensitivity to initial conditions, one of the important ingredients of chaos. Choose a value for $\rho = 28$. Evolve the system using at $t = 0$ $(x, y, z) = (\sqrt{\beta(\rho-1)} + \epsilon, \sqrt{\beta(\rho-1)} + \epsilon, \rho - 1 + \epsilon$ for two values of epsilon $\epsilon_1 = 1e-5$ and $\epsilon_2 = 2e-5$. Plot the distance between the two different solutions, i.e., $L = \sqrt{(x_{\epsilon_2} - x_{\epsilon_1})^2 + (y_{\epsilon_2} - y_{\epsilon_1})^2 + (z_{\epsilon_2} - z_{\epsilon_1})^2}$. Does the distance increase/decrease vs time or does is appear chaotic. What do you infer from this experiment about the ability to make long-term predictions using this system of equations.

   Use your judgement as to how small a step size you need to solve this system accurately. If you cannot figure this out from pure thought, experiment with different step sizes.

2. **Self-convergence**: Use a number of step sizes and make a plot to demonstrate that the code solution for x, y and z at the final time self-converges for fixed initial conditions. Does the order of convergence match your expectation? If not, try to explain why.

3. Using the order of convergence you determined, employ Richardson extrapolation to determine an error for the solution for $x(t)$, $y(t)$, $z(t)$ at a time of your choosing.