18 February 2025

Problem Set 1 : The Lorenz Model

1 Numerical Approach to the Lorenz Model

We have seen in class that the Lorenz equations are :

$$\begin{cases} \dot{X} = \sigma(Y - X) \\ \dot{Y} = -XZ + rX - Y \\ \dot{Z} = XY - bZ \end{cases}$$

(a) Using Matlab, create a program that integrates the Lorenz equations for given initial conditions. Compare the explicit Euler and Runge-Kutta 4$^{\text{th}}$ order algorithms as well as Matlab's `ode45` function. Use initial condition $(X_0, Y_0, Z_0) = (0, 1, 0)$, parameters $\sigma = 10$, $r = 10$, $b = 8/3$, intitial time $t_0 = 0$, final time $t_f = 2$ and time step $\Delta t = 0.01$.
Reminders :
For the differential equation $\dot{\vec{x}} = \vec{f}(t, \vec{x})$, the explicit Euler method approximates

$$\vec{x}(t + \Delta t) \approx \vec{x}(t) + \Delta t \cdot \vec{f}(t, \vec{x}(t))$$

while Runge-Kutta 4$^{\text{th}}$ order approximates

$$\vec{x}(t + \Delta t) \approx \vec{x}(t) + \frac{1}{6}\left[\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4\right]$$

with

$$\begin{aligned} \vec{k}_1 &= \Delta t \cdot \vec{f}(t, \vec{x}(t)) \\ \vec{k}_2 &= \Delta t \cdot \vec{f}\Big(t + \frac{\Delta t}{2}\ ,\ \vec{x}(t) + \frac{1}{2}\vec{k}_1\Big) \\ \vec{k}_3 &= \Delta t \cdot \vec{f}\Big(t + \frac{\Delta t}{2}\ ,\ \vec{x}(t) + \frac{1}{2}\vec{k}_2\Big) \\ \vec{k}_4 &= \Delta t \cdot \vec{f}\Big(t + \Delta t\ ,\ \vec{x}(t) + \vec{k}_3\Big) \end{aligned}$$

We also note that for Matlab's `ode45` the command line is `ode45(@f,[t0 tf],X0)`, where :
— `t0` and `tf` are the initial and final times
— `X0` the *column* vector containing the initial conditions $\vec{x}(t = t_0)$
— `f` is a function whose arguments are the time t and the *column* vector $\vec{x}$. It returns the *column* vector $\vec{f}(t, \vec{x})$

(b) Now refine the time step to show that your Euler and Runge-Kutta methods converge according to the order of the method. To do so, run simulations with different Δt. For the two schemes, consider the final position $\vec{x}(t_f)$ of the simulation with the smallest Δt to be your exact solution. Plot the error on the final state $||\vec{x}_{\text{exact}}(t_f) - \vec{x}(t_f)||_2$ for the various time steps.

(c) It is in general very difficult to find an analytical solution for the nonlinear system $\dot{\vec{x}} = \vec{f}(\vec{x})$. However, if there is an $\vec{x}^*$ such that $\vec{f}(\vec{x}^*) = 0$ then $\vec{x}(t) = \vec{x}^*$ is an analytical stationary solution. The points $\vec{x}^*$ that give $\vec{f}(\vec{x}^*) = 0$ are called equilibrium or fixed points. Analytically find the equilibrium points of the Lorenz system. With the parameters given in (a), towards which stationary point does your system converge ?

(d) Change to $r = 21$ to study transient chaos. This is where the intermediate state is chaotic and unpredictable (for example, the numerical error produced by different integrators can give completely different results) but, in the end, the system converges to a stationary point. Try $(X_0, Y_0, Z_0) = (10, 0, 10)$ or find other interesting initial conditions.

(e) Now for the most famous case studied by Lorenz, use $r = 28$ and $(X_0, Y_0, Z_0) = (0, 1, 0)$ to enter the chaotic regime and observe the trajectories. You can use `comet3` to see the system evolve.

(f) Analyse the sensitivity of initial conditions in this chaotic regime. Start by generating a set of $N = 100$ vectors following a Gaussian distribution of mean $(X_0, Y_0, Z_0) = (0, 1, 0)$ and standard deviation $\Delta|\vec{x}_0| = 10^{-3}$. For each of the three variables use `normrnd(X0(i),dX0,[N,1])` to generate a column vector of size `N` with a Gaussian distribution of mean `X0(i)` and standard deviation `dX0`. Integrate using the three methods with parameters $\Delta t = 0.015$ and $t_f = 30$. Plot in phase space only the final positions of these simulations to see how they are distributed.

(g) For big $r = 400$, the system can converge to a periodic behavior. Compare the different integrators. For clearer results, try initial conditions near the stable periodic trajectory $(X_0, Y_0, Z_0) = (-54, -59, 485)$.

(h) Is there order in chaos ? In the chaotic regime with $r = 28$, we can try to find some ordered patterns. Looking at the oscillations of variable Z, find the values of all local maxima of Z (Matlab's `findpeaks` will do it in a breeze). Then plot on a 2D graph the pairs (z_n, z_{n+1}), where z_n is the amplitude of the n^{th} peak. Longer simulations give more (z_n, z_{n+1}) pairs, so clearer results.

BONUS

2 Derivation of the Lorenz Equations

Following what we did in class, derive the first equation of the Lorenz model i.e. $\frac{\mathrm{d}}{\mathrm{d}\tau}X = \frac{\nu}{K}(Y - X)$. Starting from the Navier-Stokes equation

$$\frac{\partial}{\partial t}\nabla^2\psi - \frac{\partial\psi}{\partial z}\frac{\partial}{\partial x}\nabla^2\psi + \frac{\partial\psi}{\partial x}\frac{\partial}{\partial z}\nabla^2\psi - g\varepsilon\frac{\partial\theta}{\partial x} - \nu\nabla^4\psi = 0, \tag{1}$$

imposes a solution of the form

$$\psi = X(t)\alpha_1 \sin(\pi a x/H)\sin(\pi z/H)$$
$$\theta = Y(t)\alpha_2\cos(\pi a x/H)\sin(\pi z/H) + Z(t)\alpha_3\sin(2\pi z/H)$$
$$\text{with } \alpha_1 = \frac{\sqrt{2}(1+a^2)K}{a},\ \alpha_2 = \frac{\sqrt{2}\Delta T}{\pi r},\ \alpha_3 = \alpha_2/\sqrt{2} \text{ and } r = \frac{g\varepsilon H^3 \Delta T a^2}{\nu K \pi^4(1+a^2)^3}$$

To compute the derivatives, softwares capable of symbolic computation like Mathematica can be of great help. To define a function use command `f[x_,y_]:= a Cos[x] + x f2[y]`. Mathematica does not need know what function `f2` or variable `a` are to do litteral computation. `Derivative[0,3][f][b,c]` can be used for derivatives[1]. This here is the third derivative of `f` in the second variable (which is `y` in this case) evaluated at `x=b` and `y=c`. At the end make sure to renormalize to the dimensionless time $\tau = \frac{\pi^2(1+a^2)K}{H^2}t$.

1. https ://reference.wolfram.com/language/ref/Derivative.html