Numerical modelling of Lorenz system

Pawel Czyz

St Hugh's College, University of Oxford

Abstract. We present a Python module implementing various ODE solvers, enabling one to do accurate and effective numerical simulations. We investigate it's accuracy investigating Lorenz chaotic system.

Keywords: numerical modelling, Runge-Kutta algorithm, chaos, Lorenz

1 Introduction

In 1963 E. Lorenz modelling atmospheric convection proposed [1] a dynamical system:

$$y_0' = a(y_1 - y_0) \tag{1}$$
$$y_1' = ry_0 - y_1 - y_0y_2 \tag{2}$$
$$y_2' = y_0y_1 - by_2, \tag{3}$$

where y_0 is the rate of convection, y_1 and y_2 represent temperature changes in two dimensions and a, b, r are model constants related to Prandtl number, Rayleigh number and geometry of the convective layer.

As dynamical systems are in general hard problems to be solved analytically, Lorenz system is solved numerically. We decided to use classical Runge-Kutta method (RK4) [2, 3]. It allows one to integrate a dynamical system:

$$y' = f(y), \tag{4}$$

where $y = (y_0, y_1, y_2)$ with a boundary condition $y(0) = (y_0^\circ, y_1^\circ, y_2^\circ)$.

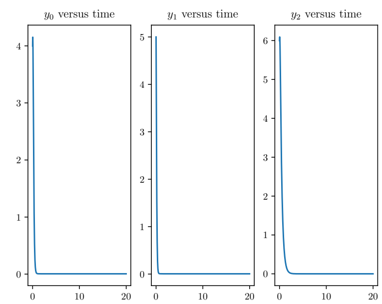
Runge-Kutta method was implemented as a Python package allowing to track the whole history of the system as a function of time.

2 Lorenz system

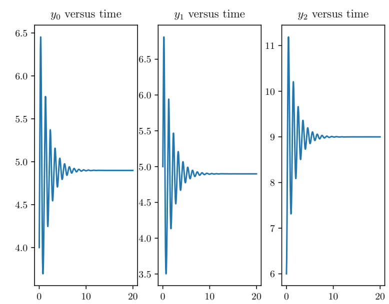
We solved system (1) for different ranges of parameter r and obtained curve $y(t)$, which projections are shown in the Figure 1. While the system exhibits damped oscillations for $r \leq 10$, starting at $r = 28$, the system seems to behave chaotically[1].

[1] Proof that Lorenz system actually *is* chaotic required huge computational power and can be found in [4].

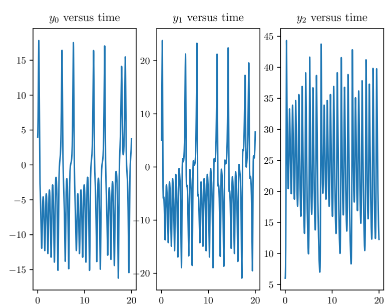
A chaotic behaviour is also visible via temperature phase diagrams in the Figure 2. It makes simulations hugely dependent on initial boundary condition. In the Figure 3. a time evolution of y_0 is presented for small perturbations in the boundary condition. Even for small (smaller than 1%) perturbations, the time evolution becomes inaccurate after short period of time.



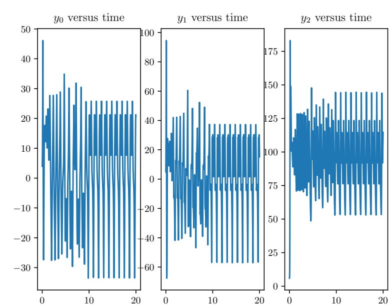
(a)



(b)



(c)

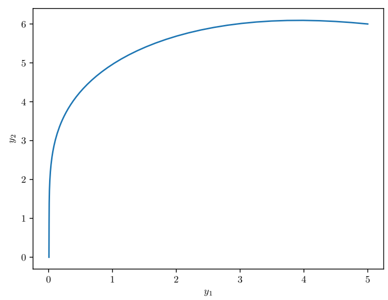


(d)

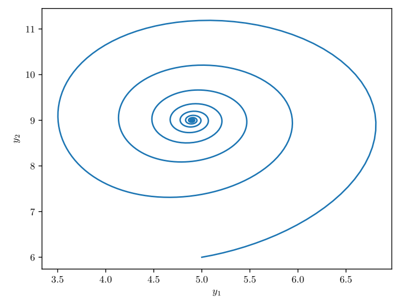
Fig. 1: Solutions $y(t)$ for $y(0) = (4, 5, 6)$ and $a = 10$, $b = 8/3$. Model constant r is different for every sub-figure: $r = 1$ for 1a, $r = 10$ for 1b, $r = 28$ for 1c. and $r = 100$ for 1d. We see a range of behaviours - for $r = 1$ we see overdamped oscillations, for $r = 10$ we see underdamped oscillations. Behaviour for $r = 28$ and $r = 100$ seems to be chaotic.

3 Conclusions

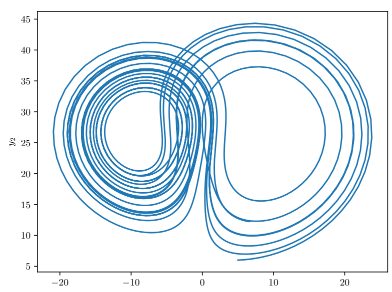
We showed why weather forecasting is a hard problem by solving Lorenz system and investigating it's time evolution using own-implemented Runge-Kutta method. We showed that the system can be believed to behave chaotically by



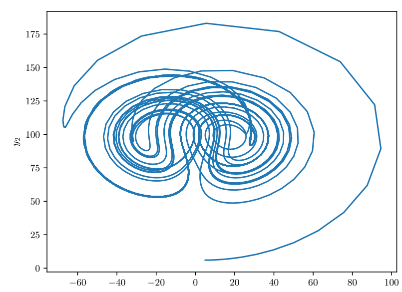
(a)



(b)



(c)



(d)

Fig. 2: Temperature phase diagrams (dependency between y_1 and y_2) for $y(0) = (4, 5, 6)$ and $a = 10$, $b = 8/3$. Model constant r is different for every sub-figure: $r = 1$ for 2a, $r = 10$ for 2b, $r = 28$ for 2c. and $r = 100$ for 2d. We see a range of behaviours - for $r = 1$ we see overdamped oscillations, for $r = 10$ we see underdamped oscillations (a spiral is an ellipse, as in oscillation, with shrinking axes). Behaviour for $r = 28$ and $r = 100$ seems to be chaotic, butterfly-shaped Lorenz attractor is present.

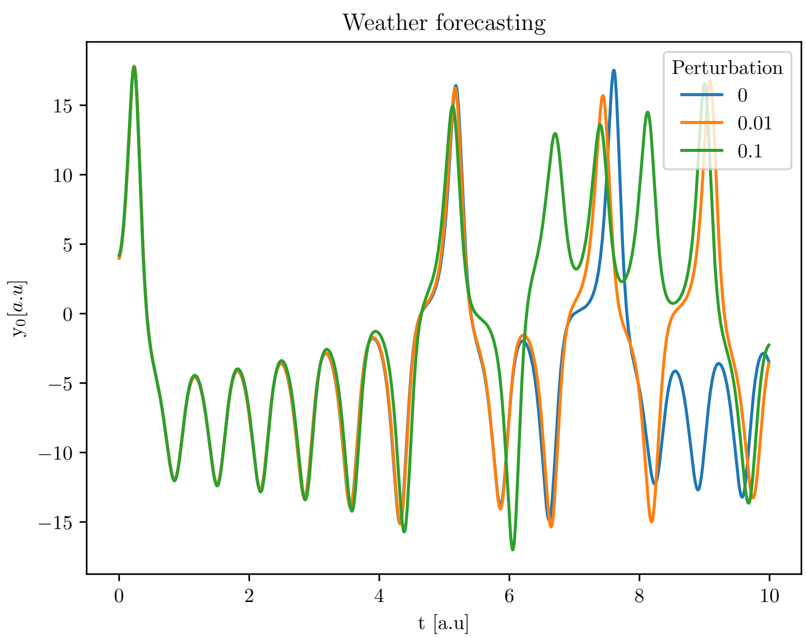


Fig. 3: Time evolution $y_0(t)$ for different boundary conditions. Model parameters are $a = 10$, $b = 8/3$, $r = 28$. Boundary conditions vary as $y(0, p) = (4 + p, 5 + p, 6 + p)$ for p taking values $0, 0.01, 0.1$. We see that after short period of time, perturbed curves become unrelated to the unperturbed curve.

investigation of phase diagrams and the dependency on perturbed initial conditions - smallest measurements errors in present increase exponentially with time. Created Python package can be also used to model different physical situations.

4 References

1. E. Lorenz, Deterministic nonperiodic flow, Journal of the Atmospheric Sciences 20, 1963
2. W. Press et al. Numerical Recipes: the Art of Scientific Computing, Cambridge University Press, 2007
3. Joint work, Chaos Lab Script, University of Oxford, 2018
4. B. Hassard et al. A Computer Proof that the Lorenz Equations Have Chaotic Solutions, Appl. Math. Lett. Vol. 7, 1994