ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Basic Sciences
Section of Mathematics
Institute of Analysis and Scientific Computing

Solving Chaotic Differential Equations using an Adaptive Algorithm

Lionel Walter

Prof. Alfio Quarteroni, Dr. Erik Burman, Dr. Paolo Zunino

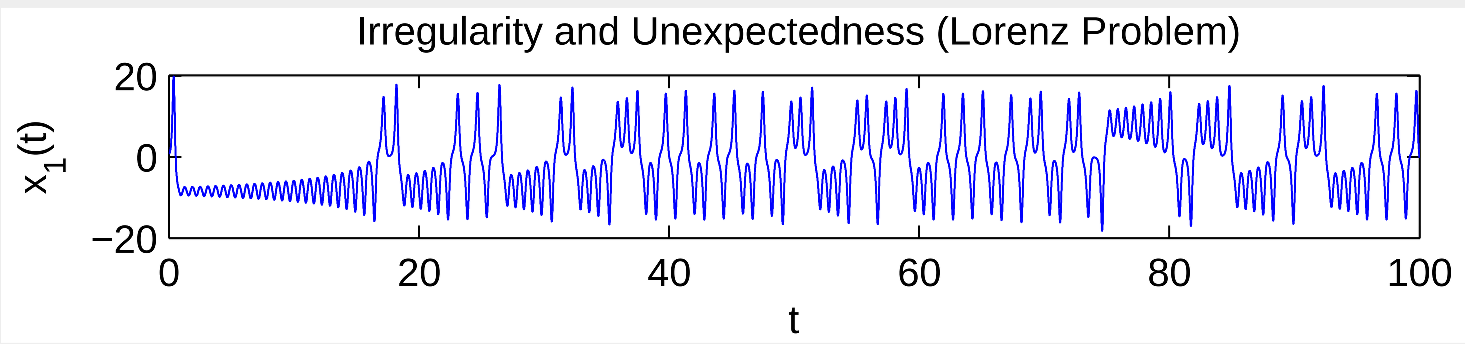
cmcs
Chair of Modelling and Scientific Computing

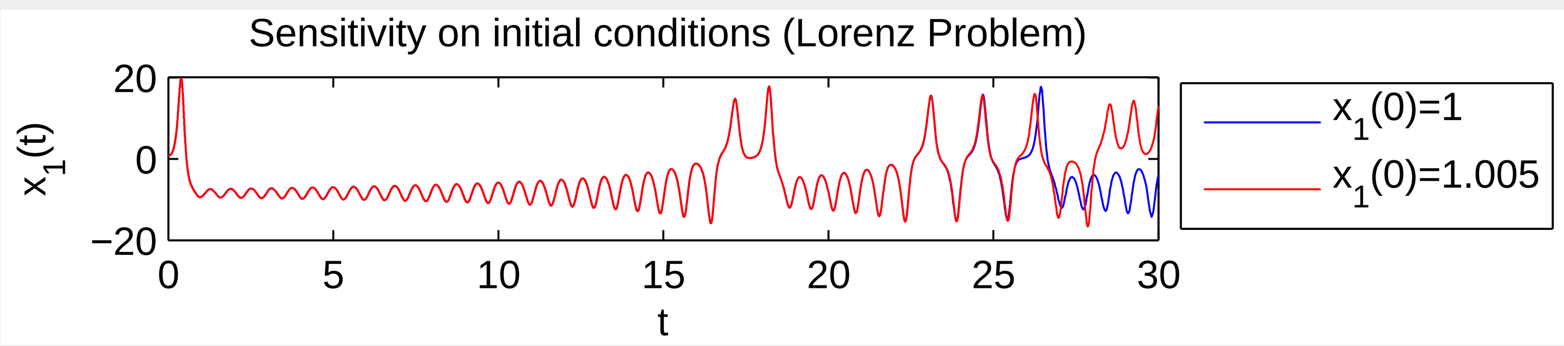
1 Introduction

Our aim is to solve chaotic differential equations, monitoring the error at the end of the time interval $[0\ T]$. Meteorology, fluid dynamics and stock exchange values are domains where our methodology applies. One typical example could be : we are Monday and the temperature is 10°C. We want to estimate the temperature of Friday with an error less than 1°C using the least possible computational effort.

2 The Chaos

By chaotic, we mean a behaviour that is very irregular and unexpected. For example, a chaotic equation is very sensitive to the initial conditions. A Monday temperature of 10°C can result in a Friday temperature of 25°C and a Monday temperature of 9.99°C can give a Friday temperature of 0°C. An initial difference of 0.01°C gives a final difference of 25°C.

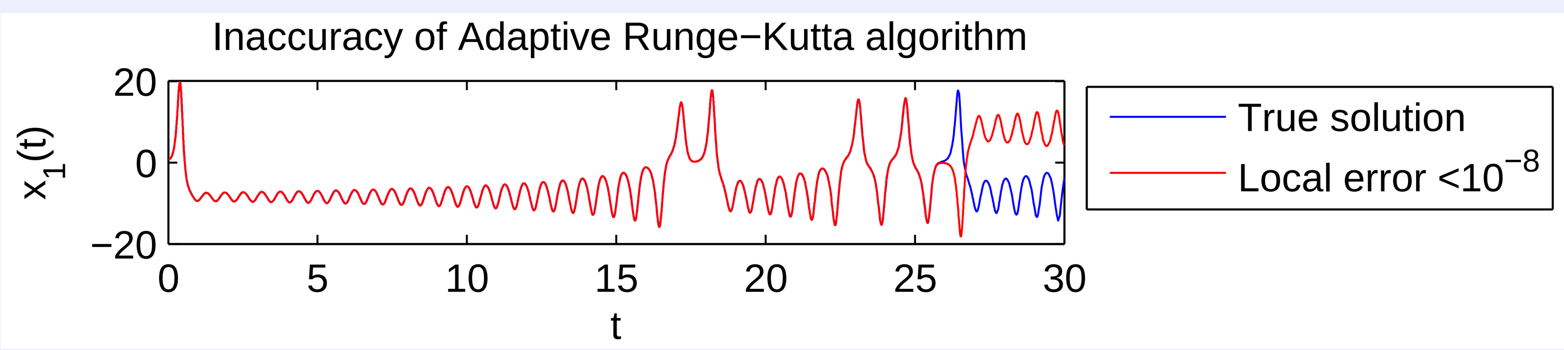




Usual methods for differential equations may fail for chaotic problems. They have big computing time and give no assessment of the quality of the solution. Using the theory of [3] we developed an algorithm designed especially for chaotic problems (`mstz` below).

3 The Ideas we used

Classical methods for differential equations monitor only the local error, i.e. the error done at each timestep. For chaotic problems, this is not enough since small local errors can give a big error at time T.



For each timestep, we calculate how a small error amplifies until T and denote it by ψ. A big value of ψ implies important amplification of the error whereas a small value of ψ indicates that the problem is unsensitive to perturbations. To compute ψ, we solve a dual problem, corresponding to the linearized original problem solved backwards in time.
Using the local errors weighted with the stability factors given by ψ `mstz` decides in what portions of the interval $[0\ T]$ precise computations are important in order to approximate the solution at time T to the prescribed precision.

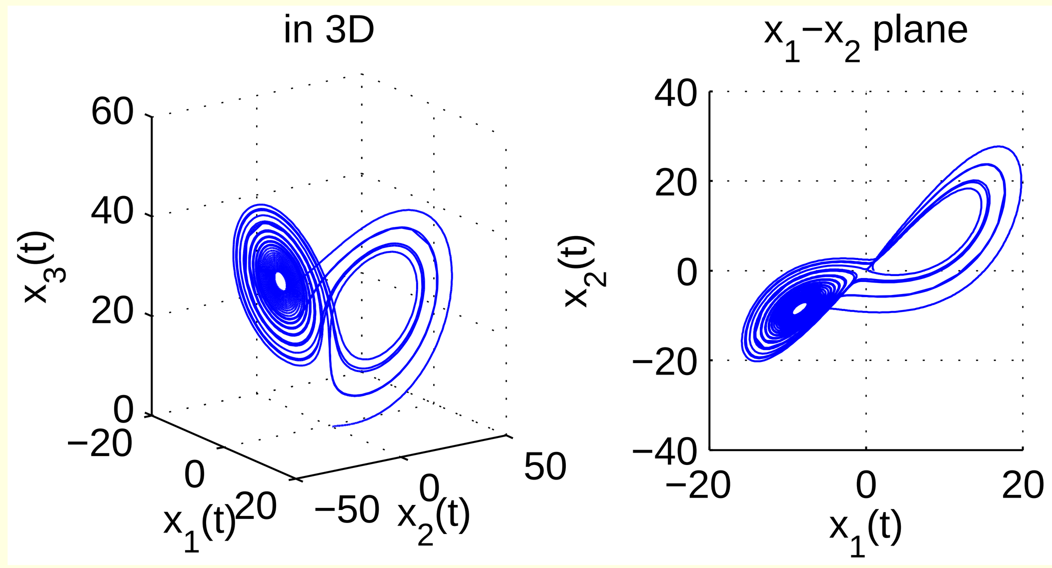
4 Results

Our reference algorithm for differential equations, denoted `RK45`, is an algorithm based on the adaptive Runge-Kutta algorithm from Dormand and Prince. It reduces the bound on the local error until the error at time T is small enough.
After having validated our Matlab code on different examples, we compared it with `RK45` on two chaotic benchmark problems.
1. The Lorenz Problem. A 3D ordinary differential equation which comes from meteorology.

$$\begin{cases} x_1'(t) = -\sigma x_1(t) + \sigma x_2(t) \\ x_2'(t) = r x_1(t) - x_2(t) - x_1(t)x_3(t) \\ x_3'(t) = x_1(t)x_2(t) - b x_3(t) \\ t \in [0\ 30],\ \sigma = 10,\ b = 8/3,\ r = 28 \end{cases}$$

Results :

Algorithm	Error	# Timesteps	Computing time[s]
`RK45`	-0.0015	16651	42
`mstz`	-0.0042	7970	26

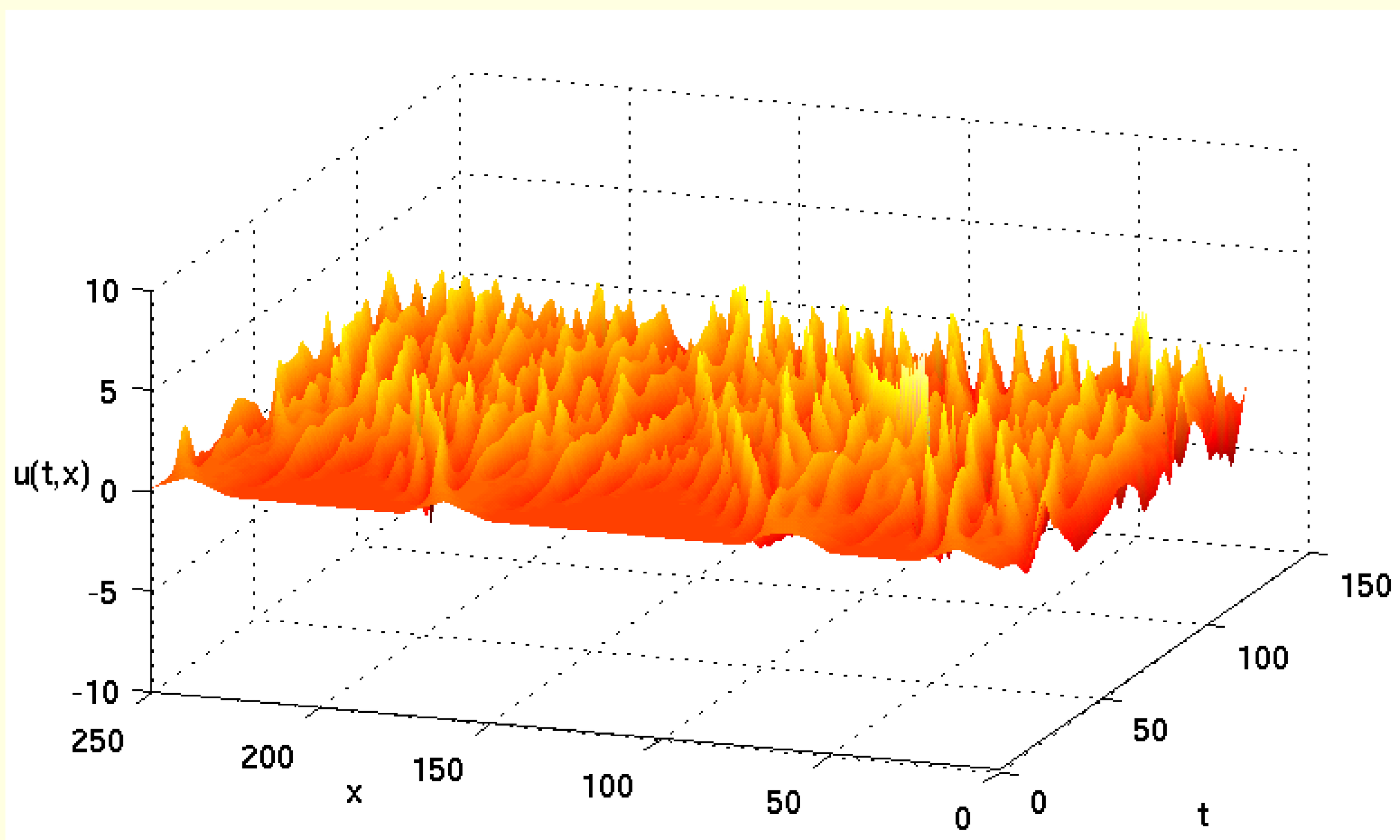


2. The Kuramoto-Sivashinsky Equation. A partial differential equation modelling certain aspects of concentration waves, flame propagation or hydrodynamic turbulence. $u : (t,x) \in \mathbb{R}^2 \mapsto u(t,x) \in \mathbb{R}$. The space semi-discretization is done using the pseudo-spectral method.

$$\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} - \frac{1}{2}\frac{\partial u^2}{\partial x}$$
$$0 \le t \le 120 \quad x \in [0\ 251]$$

Results :

Algorithm	Error	# Timesteps	Computing time[s]
`RK45`	$8.2 \cdot 10^{-7}$	5855	138
`mstz`	$4.6 \cdot 10^{-8}$	2064	158



5 Conclusions

For chaotic problems, it is very important to monitor the stability factors. It gives more accurate and faster algorithms. Also, it gives an idea of the computability of the problem. Problems which are too chaotic are impossible to solve numerically.
To have a good efficiency, we have shown numerically the advantage of solving the dual problem less accurately than the original problem.

[1] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, 6(1):19–26, 1980.
[2] Kyoung-Sook Moon, Anders Szepessy, Raúl Tempone, and Georgios E. Zouraris. Convergence rates for adaptive approximation of ordinary differential equations. *Numer. Math.*, 96(1):99–129, 2003.
[3] Kyoung-Sook Moon, Anders Szepessy, Raúl Tempone, and Georgios E. Zouraris. A variational principle for adaptive approximation of ordinary differential equations. *Numer. Math.*, 96(1):131–152, 2003.