VI International Conference on Adaptive Modeling and Simulation
ADMOS 2013
J. P. Moitinho de Almeida, P. Díez, C. Tiago and N. Parés (Eds)

QUANTIFYING THE COMPUTABILITY OF THE LORENZ SYSTEM

Benjamin Kehlet[*†] **and Anders Logg**[†*]

[*]Department of Informatics, University of Oslo
P.O.Box 1080 Blindern, 0316 OSLO,
Norway

[†]Center for Biomedical Computing at Simula Research Laboratory,
P.O.Box 134, 1325 Lysaker,
Norway

Key words: Lorenz, chaos, high precision, high order, finite element, time stepping

Abstract. It is well known that the computation of accurate trajectories of the Lorenz system is a difficult problem. Computed solutions are very sensitive to the discretization error determined by the time step size and polynomial order of the method, as well as round-off errors.

In this work, we show how round-off errors limit the computability of the Lorenz system and quantify exactly the length of intervals over which solutions can be computed, expressed in terms of the floating point precision. Using adjoint-based *a posteriori* error analysis techniques, we estimate the stability of computations with respect to initial data, discretization, and round-off errors, respectively.

The analysis is verified by computing an accurate solution on the time interval $[0, 1000]$ using a very high order (order 200) finite element method and very high floating point precision (400 digits).

1 Introduction

In a classic paper from 1963 [13], Edward Lorenz studied the computability of a simple system of three ordinary differential equations,

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = rx - y - xz, \\ \dot{z} = xy - bz, \end{cases} \tag{1}$$

where $\sigma = 10$, $b = 8/3$, and $r = 28$. Lorenz computed numerical solutions of the system (1) and found the solutions to be very sensitive to changes in initial data. The equations had been devised by Lorenz as a simple model of atmospheric flow, based on a truncated Fourier expansion of the partial differential equations governing Rayleigh–Bénard convection [14, 12, 15]. In his paper, Lorenz computed solutions on the interval $[0, 60]$. As we shall see below, the Lorenz system is not computable on the equipment that was available to Lorenz in 1963 beyond time $T \approx 25$.

It is known that, given enough resources, the Lorenz system is computable over arbitrarily long time intervals. However, one may easily (and falsely) come to the conclusion that the Lorenz system is not computable, or computable only over very short time intervals, either by numerical experiments or by a simplistic analysis. Indeed, a standard *a priori* error estimate indicates that the growth rate of the error is

$$\|e(T)\| \leq C e^{LT} \epsilon, \tag{2}$$

where $\|e(T)\|$ denotes some norm of the error at the final time T, L is the Lipschitz constant of (1), and ϵ is the size of the residual or local truncation error in a numerical solution of (1). The Lipschitz constant is of size $L \approx 33$ which indicates that solutions are not computable beyond $T \approx 1.1$, even if the residual is close to machine precision ($\epsilon_{\mathrm{mach}} \sim 10^{-16}$ on most computers).[1] However, the estimate (2) is overly pessimistic; it is well known that solutions of the Lorenz system may be computed on short time intervals. In fact, one may easily compute accurate solutions over time intervals of length $T = 25$ with any standard ODE solver.

In [3], it was demonstrated that the Lorenz system is indeed computable on intervals of moderate length ($T = 30$) on a standard desktop computer. The computability of the Lorenz system was linked to the growth of a *stability factor* in an *a posteriori* estimate of the error at the final time. It was shown that the growth rate of the stability factor is non-constant. On average the growth is exponential but with a rate much smaller than indicated by (2).

In [10], the computability of the Lorenz system was further extended to $T = 48$ using high order ($\|e(T)\| \sim \Delta t^{30}$) finite element methods. As we shall see below, this is the “theoretical limit” for computations with 16 digit precision. Solutions over longer time intervals have been computed based on shadowing (the existence of a nearby exact solution), see [2], but for unknown initial data. Other related work on high-precision numerical methods applied to the Lorenz system include [17] and [5]. For an overview of some recent results obtained with high-precision numerical methods, we also refer to [1].

In this paper, we study and quantify the computability of the Lorenz system. In particular we answer the following fundamental question: *How far is the Lorenz system computable for a given machine precision?*

As we shall see, obtaining a sequence of converging approximations for the solution of the Lorenz system is non-trivial. In particular, such a sequence of solutions cannot be

[1] The value of the Lipschitz constant was computed as the maximum l^2-norm of the Jacobian $J = \partial f / \partial u$ of the right-hand side f of the Lorenz system over the interval $[0, 1000]$.

obtained by simply decreasing the size of the time step; see for example [16]. This has led to misconceptions regarding the computability of the Lorenz system; see for example [18]. To obtain a sequence of converging solutions, one must also control the effect of round-off errors. This was also noted by Lorenz [11] in a response to [18].

In this manuscript, we define computability as the length T of the maximum time interval $[0, T]$ on which a solution is computable to within a given precision $\epsilon > 0$ using a given machine precision $0 < \epsilon_{\mathrm{mach}} < \epsilon$; that is, the maximum T such that $\inf_U \|u - U\|_{L^\infty(0,T;l^\infty)} \leq \epsilon$, where the infimum is taken over all numerical approximations U of the exact solution u computed with some time-stepping method and machine precision ϵ_{mach} (as made more precise in Section 3). If the computability $T_\epsilon = T_\epsilon(\epsilon_{\mathrm{mach}})$ does not depend strongly on ϵ, we write $T = T(\epsilon_{\mathrm{mach}})$. As we shall see, this is the case for the Lorenz system as a result of exponential growth of errors as function of the final time T. The definition of computability $T(\epsilon_{\mathrm{mach}})$ is closely related to the definition of a critical predictable time T_c in [8] and the definition of a decoupling time $\hat{T}$ in [16].

2 Numerical method and implementation

We consider the numerical solution of general initial value problems for systems of ordinary differential equations,

$$\begin{aligned} \dot{u}(t) &= f(u(t), t), \quad t \in (0, T], \\ u(0) &= u_0. \end{aligned} \tag{3}$$

The right-hand side $f : \mathbb{R}^N \times [0, T] \to \mathbb{R}^N$ is assumed to be Lipschitz continuous in u and continuous in t. Our objective is to analyze the error in an approximate solution $U : [0, T] \to \mathbb{R}^N$, for example a numerical solution of the Lorenz system.

The continuous and discontinuous Galerkin methods cG(q) and dG(q) are formulated by requiring that the residual $R = \dot{U} - f(U, \cdot)$ be orthogonal to a suitable space of test functions. By making a piecewise polynomial Ansatz, the solution may be computed on a sequence of intervals partitioning the computational domain $[0, T]$ by solving a system of equations for the degrees of freedom on each consecutive interval. For a particular choice of numerical quadrature and degree q, the cG(q) and dG(q) methods both reduce to standard implicit Runge–Kutta methods.

In the case of the cG(q) method, the numerical solution U is a continuous piecewise polynomial of degree q that on each interval $[t_{n-1}, t_n]$ satisfies $\int_{t_{n-1}}^{t_n} R\, v \,\mathrm{d}t = 0$ for all $v \in \mathcal{P}^{q-1}([t_{n-1}, t_n])$.

The results were obtained using the finite element package DOLFIN [9] version 0.9.2 together with the multi-precision library GMP [4]. For a detailed discussion on the implementation, we refer to [7]. The source code as well as scripts to reproduce all results presented in this manuscript are available on request.

3 Error analysis

The error analysis is based on the solution of an auxiliary *dual problem*. The dual (adjoint) problem takes the form of an initial value problem for a system of linear ordinary differential equations,

$$\begin{aligned} -\dot{z}(t) &= \bar{A}^\top(t) z(t), \quad t \in [0,T), \\ z(T) &= z_T. \end{aligned} \tag{4}$$

Here, $\bar{A}(t) = \int_0^1 \frac{\partial f}{\partial u}(sU(t) + (1-s)u(t), t)\,\mathrm{d}s$ denotes the Jacobian matrix of the right-hand side f averaged over the approximate solution U and the exact solution u.

The Lorenz system is quadratic in the primal variable u. Hence, the average in $\bar{A}$ corresponds to evaluating the Jacobian matrix at the midpoint between the two vectors $U(t)$ and $u(t)$. It follows that the dual problem of the Lorenz system is

$$\begin{cases} -\dot{\xi} = -\sigma\xi + (r - \bar{z})\eta + \bar{y}\zeta, \\ -\dot{\eta} = \sigma\xi - \eta + \bar{x}\zeta, \\ -\dot{\zeta} = -\bar{x}\eta - b\zeta, \end{cases} \tag{5}$$

where $z = (\xi, \eta, \zeta)$ denotes the dual solution and $(\bar{x}, \bar{y}, \bar{z}) = (U + u)/2$.

In [6], we prove the following *a posteriori* error estimate:

Theorem 3.1 (Error estimate)**.** *Let $u : [0,T] \to \mathbb{R}^N$ be the exact solution of* (3) (*assuming it exists*)*, let $z : [0,T] \to \mathbb{R}^N$ be the solution of* (4)*, and let $U : [0,T] \to \mathbb{R}^N$ be any piecewise smooth approximation of u on a partition $0 = t_0 < t_1 < \cdots < t_M = T$ of* $[0,T]$*, that is, $U|_{(t_{m-1},t_m]} \in \mathcal{C}^\infty((t_{m-1}, t_m])$ for $m = 1, 2, \ldots, M$ (U is left-continuous).*

Then, for any $p \geq 0$, the following error estimate holds:

$$\langle z_T, U(T) - u(T) \rangle = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C,$$

where

$$\begin{aligned} |\mathbf{E}_D| &\leq S_D \, \|U(0) - u(0)\|, \\ |\mathbf{E}_G| &\leq S_G \, C_p \max_{[0,T]} \left\{ \Delta t^{p+1} (\|[U]\|/\Delta t + \|R\|) \right\}, \\ |\mathbf{E}_C| &\leq S_C \, C_p' \max_{[0,T]} \|\Delta t^{-1} \bar{R}\|, \end{aligned}$$

where C_p and C_p' are constants depending only on p. The stability factors S_D, S_G, and S_C are defined by

$$S_D = \|z(0)\|, \quad S_G = \int_0^T \|z^{(p+1)}\| \,\mathrm{d}t, \quad S_C = \int_0^T \|\pi z\| \,\mathrm{d}t.$$

Furthermore the following bound for the computational error is proved in [6]:

Theorem 3.2. *Assume that the round-off error is a random variable of size* $\pm\epsilon_{mach}$ *with equal probabilities. Then the root-mean squared expected computational error* $\mathbf{E}_C$ *of Theorem 3.1 is bounded by*

$$(E[\mathbf{E}_C^2])^{1/2} \leq S_{C_2}\sqrt{C_p'}\frac{\epsilon_{mach}}{\min_{[0,T]}\sqrt{\Delta t}},$$

where $S_{C_2} = \left(\int_0^T \|\pi z\|^2 \,\mathrm{d}t\right)^{1/2}$ *and* C_p' *is a constant depending only on* p.

We note that the computational error (accumulated round-off error) is inversely proportional to (the square root of) the time step; that is, a *smaller* time step yields a *larger* accumulated round-off error.

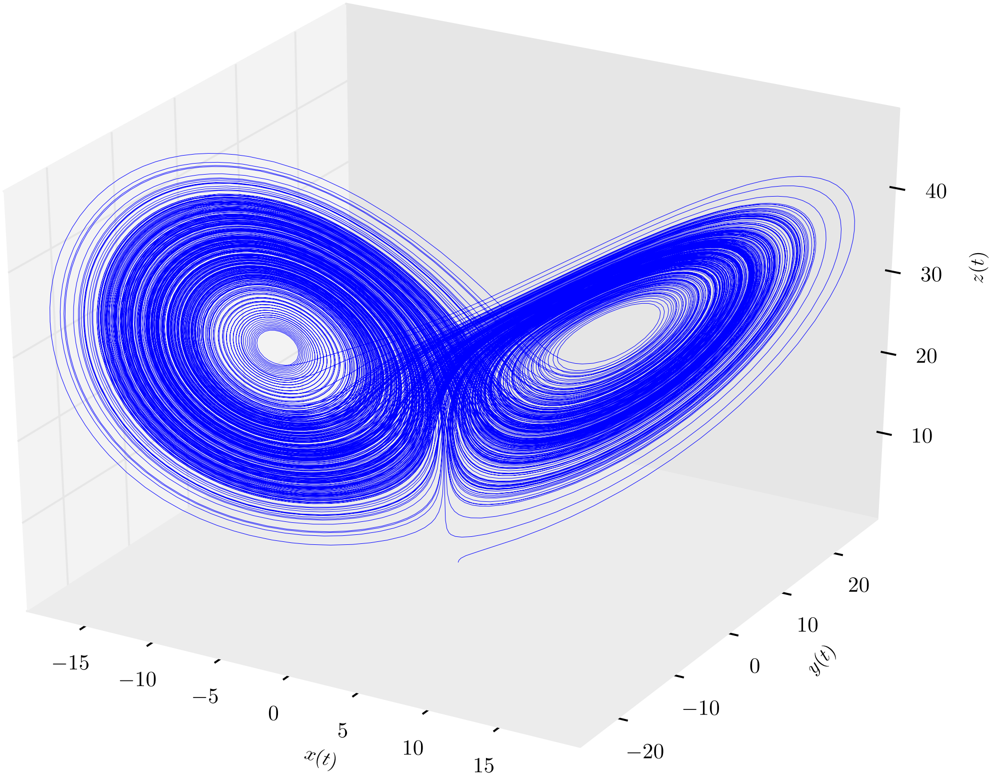

Figure 1: Phase portrait of the solution of the Lorenz system on the time interval $[0, 1000]$ for $u(0) = (1, 0, 0)$.

4 Numerical results

In this section, we present numerical results in support of Theorem 3.1 and Theorem 3.2.

4.1 Solution of the Lorenz system

The phase portrait of the solution of the Lorenz system is plotted in Figure 1. The solution was computed with cG(100), which is a method of order $2q = 200$, a time step of size $\Delta t = 0.0037$, 420-digit precision arithmetic[2], and a tolerance for the discrete residual of size $\epsilon_{\mathrm{mach}} \approx 2.26 \cdot 10^{-424}$. The solution trajectory revolves around one of the two unstable fixed points $P_{\pm} = (\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27)$ for a while and then, seemingly at random, jumps to the other fixed point. Phase portraits ("attractors") resembling the phase portrait of Figure 1 are commonly displayed in most books on dynamical systems and chaos theory. However, in one way the phase portrait of Figure 1 is significantly different. It is the phase portrait of a well-defined dynamical system, namely the Lorenz system (1) with initial condition $(1, 0, 0)$, not the result of an unspecified discrete map which includes both the effect of a particular time-stepping scheme and the unknown effect of round-off errors.

To verify the computed solution, we perform a simple experiment where we compute the solution with methods of increasing order. The time step is fixed ($\Delta t = 0.0037$) and so is the arithmetic precision (420 digits). By Theorem 3.1, we expect the discretization error $\mathbf{E}_G$ to decrease exponentially with increasing order while the computational error $\mathbf{E}_C$ remains bounded. The error should therefore decrease, until $\mathbf{E}_G < \mathbf{E}_C$. Since no analytic solution or other reference solution is available, we compare the cG(10) solution with the cG(20) solution and conclude that when the two solutions no longer agree to within some tolerance (here 10^{-16}), the cG(10) solution is no longer accurate. The same experiment is repeated for cG(20/30), cG(30/40), ..., cG(90/100), cG(99/100). The solutions are displayed in Figure 2. The results indicate that the cG(99) solution is accurate on the time interval $[0, 1025]$. Alone, this does not prove that the cG(99) is accurate at time $T = 1025$. However, together with the error estimate of Theorem 3.1 and the numerically computed values of the stability factors presented below, there is strong evidence that the solution is accurate over $[0, 1025]$.

We emphasize that similar results may be obtained with other numerical methods and other software. In particular, Theorem 3.1 shows that the solution is computable with any solver that (i) discretizes the equations with high order and (ii) solves the discrete equations with high precision. The authors are aware of two such solvers: the DOLFIN solver used in this work and Taylor [5]. The full reference solution is available on request.

[2] The requested precision from GMP was 420 digits. The actual precision is somewhat higher depending on the number of significant bits chosen by GMP.

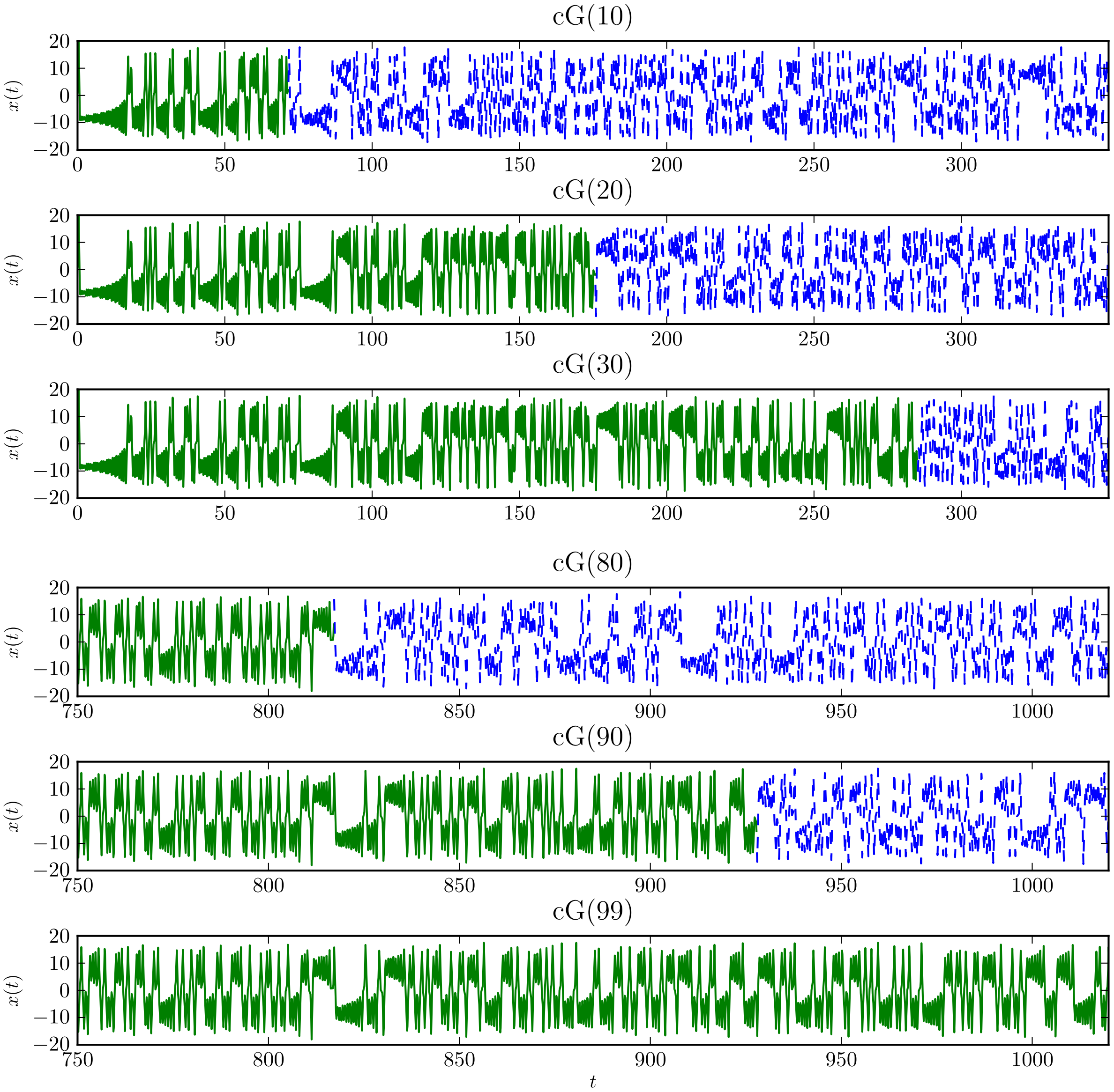

Figure 2: Computed numerical solutions (x-component) for the Lorenz system with methods of increasing order, starting at cG(10) (a method of order 20) and increasing up to cG(99) (a method of order 198).

S_D	S_G	S_C
$0.510 \cdot 10^{388}$	$28.9 \cdot 10^{388}$	$2.08 \cdot 10^{388}$

Table 1: Size of the stability factors S_D, S_G (for cG(1)), and S_C at $T = 1000$.

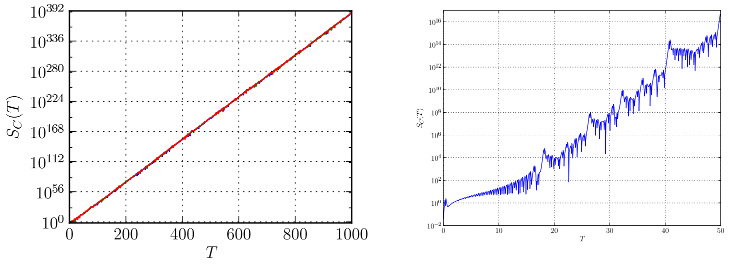

Figure 3: Growth of the stability factors S_C on the time interval $[0, 1000]$ (left) and $[0, 50]$ right.

4.2 Dual solution and stability factors

The dual solution grows exponentially backward in time. The size of the dual solution at time $t = 0$ is $S_D = \|z(0)\| \approx 0.510 \cdot 10^{388}$. By Theorem (3.1), it follows that perturbations in initial data for the Lorenz system are amplified by a factor 10^{388} at time $T = 1000$. The amplification of round-off errors may be estimated similarly by integrating the norm of the dual solution over the time interval. One finds that $S_C = \int_0^T \|\pi z\| \, \mathrm{d}t \approx 2.08 \cdot 10^{388}$, which is the amplification of errors caused by finite precision arithmetic. The stability factor for discretization errors depends on the numerical method and in the case of the cG(1) method, one finds that $S_G = \int_0^T \|\dot{z}\| \, \mathrm{d}t \approx 28.9 \cdot 10^{388}$. This is summarized in Table 1.

By repeatedly solving the dual problem on time intervals of increasing size, it is possible to examine the growth of the stability factors as function of the end time T. The result is displayed in Figure 3. Note that each data point (T, S) in Figure 3 corresponds to a solution of the dual problem on the interval $[0, T]$.

By Figure 3, it is evident that the stability factors grow exponentially with the end time T. On $[0, 1000]$, the growth of the stability factor(s) may be approximated by

$$S(T) \sim 10^{0.388T} \sim 10^{0.4T}. \tag{6}$$

The rate of growth is very stable and it is therefore reasonable to extrapolate beyond time $T = 1000$ to predict the computability of the Lorenz system on $[0, \infty)$. We return to this question below in Section 5.

A growth rate of $10^{0.388T}$ is far below the growth rate e^{33T} indicated by the simple analytic *a priori* error estimate (2). A close inspection of the growth of the stability factor S_C (Figure 3) explains the discrepancy between the two estimates. The growth rate of the

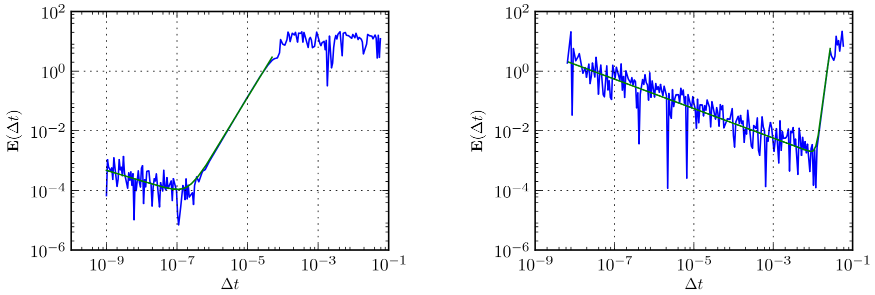

Figure 4: Error at time $T = 30$ for the cG(1) solution (left) and at time $T = 40$ for the cG(5) solution (right) of the Lorenz system. The slopes of the green lines are $-0.35 \approx -1/2$ and $1.95 \approx 2$ for the cG(1) method. For the cG(5) method, the slopes are $-0.49 \approx -1/2$ and $10.00 \approx 10$.

stability factor is not constant; it is not even monotonically increasing. While it sometimes grows very rapidly, the average growth rate is much smaller. The analytic *a priori* estimate must account for the worst case growth rate and will therefore overestimate the rate of error accumulation by a large margin.

4.3 Error propagation

We conclude this section by examining how the error depends on the size of the time step Δt. In Section 3, we found that the discretization error $\mathbf{E}_G$ scales like Δt^{2q} for the cG(q) method. On the other hand, we expect the computational error $\mathbf{E}_C$ to scale like $\Delta t^{-1/2}$. Since initial data is represented with very high precision, we have $\mathbf{E} \approx \mathbf{E}_G + \mathbf{E}_C \sim \Delta t^{2q} + \Delta t^{-1/2}$. We thus expect the error to decrease when the time step is decreased, at least initially. However, at the point where $\mathbf{E}_G = \mathbf{E}_C$, the computational error will start to dominate and we expect to see the error *increase* with decreasing time step. This is confirmed by the results presented in Figure 4, which also confirm the convergence rates $\mathbf{E}_G \sim \Delta t^{2q}$ and $\mathbf{E}_C \sim \Delta t^{-1/2}$. We also note that the error remains bounded for large values of Δt; the numerical solution stays close to the attractor but in the wrong place.

5 Computability of the Lorenz system

5.1 A model for the computability of the Lorenz system

Based on the analysis of Section 3 and the numerical results of Section 4, we develop a model for the computability of the Lorenz system. We consider the cG(q) method and make the following Ansatz for the error at the final time T as function of the time step Δt, the polynomial degree q, and the precision ϵ_{mach},

$$\mathbf{E} = \left[C_1^{[q]} \|U(0) - u(0)\| + C_2^{[q]} \Delta t^{\alpha} + C_3^{[q]} \Delta t^{\beta} \epsilon_{\mathrm{mach}} \right] \cdot 10^{0.388T}.$$

q	2	3	4	5
α	4.04	5.46	8.15	10.00
β	-0.47	-0.50	-0.50	-0.49

Table 2: Values of the constants α and β as function of q at time $T = 40$.

q	2	3	4	5
$C_2^{[q]}$	0.000356	0.000135	0.000032	0.000007
$C_3^{[q]}$	0.0031	0.0036	0.0042	0.0048

Table 3: Values of the constants $C_2^{[q]}$ and $C_3^{[q]}$ as function of q.

To determine the constants α, β, $C_1^{[q]}$, $C_2^{[q]}$, and $C_3^{[q]}$, we repeat the experiment of Figure 4 for $q = 2, 3, 4, 5$ on the interval $[0, 40]$ using the cG(100) solution as a reference. The constants α and β may be determined by a least-squares fitting of a linear polynomial to the regime where the error is dominated by the discretization error $\mathbf{E}_G$ or the computational error $\mathbf{E}_C$, respectively. The results are given in Table 2. As expected, we find that $\alpha \approx 2q$. Furthermore, we find that $\beta \approx -1/2$ in agreement with Theorem 3.2.

Next, we fix the constants $\alpha = 2q$ and $\beta = -1/2$ and determine the constants $C_1^{[q]}$, $C_2^{[q]}$, and $C_3^{[q]}$ as function of q. In Section 3, we found that $S_D(T) = \|z(0)\| \approx 0.510 \cdot 10^{0.388T}$; hence $C_1^{[q]} \approx 0.5$. By fitting curves of the form $C_2^{[q]} \Delta t^{2q} \cdot 10^{0.388T}$ and $C_3^{[q]} \Delta t^{-1/2} \cdot 10^{0.388T}$ to the two regimes where either $\mathbf{E}_G$ or $\mathbf{E}_C$ dominates, we find values for the constants $C_2^{[q]}$ and $C_3^{[q]}$. We expect $C_2^{[q]}$ to decrease with increasing q (it is essentially an interpolation constant) and $C_3^{[q]}$ to grow at a moderate rate (by a close inspection of the proof of Theorem 3.2). The results are listed in Table 3. Based on these results, we find that

$$\begin{aligned} C_2^{[q]} &< 0.001, \\ C_3^{[q]} &\approx 0.002 + 0.0005q. \end{aligned}$$

We thus arrive at the following model for the propagation of errors:

$$\mathbf{E} \approx \left[0.5\,\|U(0) - u(0)\| + 0.001\Delta t^{2q} + (0.002 + 0.0005q)\Delta t^{-1/2}\epsilon_{\mathrm{mach}})\right] \cdot 10^{0.388T}. \quad (7)$$

5.2 Optimal time step

Based on the model (7), we determine an estimate of the optimal time step size by setting $\mathbf{E}_G = \mathbf{E}_C$. We find that

$$\Delta t = ((2 + 0.5q)\epsilon_{\mathrm{mach}})^{\frac{1}{2q+1/2}} \approx \epsilon_{\mathrm{mach}}^{\frac{1}{2q+1/2}} \quad (8)$$

for large values of q. Inserting the values $\epsilon_{\text{mach}} = 10^{-420}$ and $q = 100$ used in this work, we find $\Delta t \approx 0.008$ which is reasonably close to the value of $\Delta t = 0.0037$ which was used to compute the solution.

5.3 Computability as function of machine precision

To answer the question posed in the introduction — *How far is the solution computable for a given machine precision?* — we insert the approximate optimal time step Δt given by (8) into (7). Neglecting data errors, that is, assuming $U(0) = u(0)$, we find that

$$\mathbf{E} \approx 2 \cdot 0.001 \Delta t^{2q} \cdot 10^{0.388T} \approx 0.002 \epsilon_{\text{mach}}^{\frac{2q}{2q+1/2}} \cdot 10^{0.388T} \approx 0.002 \epsilon_{\text{mach}} \cdot 10^{0.4T}$$

for large values of q. Let $n_{\text{mach}} = -\log_{10} \epsilon_{\text{mach}}$ be the number of significant digits. It follows that $\mathbf{E} \approx 0.002 \cdot 10^{0.4T - n_{\text{mach}}}$. We conclude that the computability T_ϵ, that is, the time T_ϵ at which the solution is no longer accurate to within a precision ϵ is

$$T_\epsilon(\epsilon_{\text{mach}}) = \frac{n_{\text{mach}} + \log_{10}(\epsilon/0.002)}{0.4}.$$

Since T_ϵ does not depend strongly on ϵ (for $\epsilon_{\text{mach}} \ll \epsilon$), we find that the computability of the Lorenz system is given by

$$T(\epsilon_{\text{mach}}) = n_{\text{mach}}/0.4 = 2.5\, n_{\text{mach}}.$$

With six significant digits available to Lorenz in 1963, the computability was limited to $T \approx 2.5 \cdot 6 = 15$. With 16 significant digits, the computability is limited to $T \approx 2.5 \cdot 16 = 40$. Finally, with 420 significant digits, as was used in this work, the computability is limited to

$$T \sim 2.5 \cdot 420 = 1050 > 1000.$$

A more precise estimate is possible by considering the actual size of the stability factor at any given time T. Noting that $S_C(T) \approx 2 \cdot 10^{388}$ at $T = 1000$, we may obtain the estimate

$$\mathbf{E} \approx 0.001 \epsilon_{\text{mach}}\, S_C(T).$$

With $\epsilon_{\text{mach}} = 10^{-16}$, it follows from Figure 3 that $\mathbf{E} = 0.001$ at $T \approx 50$. Furthermore, for $\epsilon_{\text{mach}} = 10^{-6}$ we find that the computability is limited to $T \approx 25$.

6 Conclusions

We have investigated the computability of the Lorenz system and come to the conclusion that the size of the time interval on which the solution is computable scales linearly with the number of digits, $T \sim 2.5\, n_{\text{mach}}$. Thus, with 420 digits of precision, as used in this work, the computability is limited to $2.5 \cdot 420 \approx 1000$. Furthermore, if a precision of 840 digits is used, one may compute the solution on the time interval $[0, 2000]$ and if a precision of 4200 digits is used, one may compute the solution on the time interval $[0, 10000]$.

Acknowledgements

This work is supported by an Outstanding Young Investigator grant from the Research Council of Norway, NFR 180450. This work is also supported by a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory.

REFERENCES

[1] D.H. Bailey, R. Barrio, and J.M. Borwein. High-precision computation: Mathematical physics and dynamics. *Applied Mathematics and Computation*, 218(20):10106–10121, 2012.

[2] B. A Coomes, H. Kocak, and K. J Palmer. Rigorous computational shadowing of orbits of ordinary differential equations. *Numerische Mathematik*, 69(4):401–421, 1995.

[3] D. Estep and C. Johnson. The pointwise computability of the Lorenz system. *Math. Models. Meth. Appl. Sci.*, 8:1277–1305, 1998.

[4] T. Granlund et al. GMP (GNU multiple precision arithmetic library). *http://gmplib.org/*, 1996–2010.

[5] A. Jorba and M. Zou. A software package for the numerical integration of ODEs by means of high-order Taylor methods. *Experimental Mathematics*, 14(1):99–117, 2005.

[6] B. Kehlet and A. Logg. A posteriori error analysis of round-off errors in the numerical discretization of ordinary differential equations. *In prepration*, 2013.

[7] Benjamin Kehlet. *Analysis and implementation of high-precision finite element methods for ordinary differential equations with application to the Lorenz system.* MSc thesis, Department of Informatics, University of Oslo, 2010.

[8] S. Liao. On the reliability of computed chaotic solutions of non-linear differential equations. *Tellus A*, 61(4):550–564, 2009.

[9] A. Logg, K.-A. Mardal, and G. N. Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering.* Springer, 2012.

[10] Anders Logg. Multi-Adaptive Galerkin methods for ODEs II: Implementation and applications. *SIAM J. Sci. Comput.*, 25(4):1119–1141, 2003.

[11] E. N Lorenz. Reply to comment by L.-S. Yao and D. Hughes. *Tellus A*, 60(4):806–807, 2008.

[12] Edward N. Lorenz. Maximum simplification of the dynamic equations. *Tellus*, 12:243–254, 1960.

[13] Edward N. Lorenz. Deterministic nonperiodic flow. *J. Atmosph. Sci.*, 20:130–141, 1963.

[14] Lord Rayleigh. On convective currents in a horizontal layer of fluid when the higher temperature is on the under side. *Phil. Mag.*, 32:529–546, 1916.

[15] B. Saltzman. Finite amplitude free convection as an initial value problem. *J. Atmosph. Sci.*, 19:329–341, 1962.

[16] J. Teixeira, C.A. Reynolds, and K. Judd. Time step sensitivity of nonlinear atmospheric models: Numerical convergence, truncation error growth, and ensemble design. *Journal of the Atmospheric Sciences*, 64(1):175–189, 2007.

[17] D. Viswanath. The fractal property of the Lorenz attractor. *Physica D: Nonlinear Phenomena*, 190(1-2):115–128, 2004.

[18] L. S Yao and D. Hughes. Comment on "Computational periodicity as observed in a simple system" by Edward N. Lorenz (2006a). *Tellus A*, 60(4):803–805, 2008.