LONG-TIME COMPUTABILITY OF THE LORENZ SYSTEM

BENJAMIN KEHLET AND ANDERS LOGG

Abstract. The purpose of this paper is threefold: (i) to demonstrate that the Lorenz system is computable over long time intervals; (ii) to prove sharp computable error estimates for the propagation of round-off errors and quantify the computability of the Lorenz system; and (iii) to present a reference solution of the Lorenz system on the time interval [0, 1000].

1. Introduction

In a classic paper from 1963 [22], 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[1] 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 [24, 21, 25]. 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 $T \approx 25$.

Aside from its implications on the possibility of weather predictions over long time intervals, the Lorenz system has gained significant interest among mathematicians, in particular in the dynamical systems community. The main focus of interest has been directed towards the existence of a (strange) attractor [26, 28] and the geometric properties of the attractor [1, 11, 31]. See [29] for an overview.

Surprisingly, there have been few results on the computability of the Lorenz system. A common misconception is that the Lorenz system is not computable, or computable only over very short time intervals. Indeed, a standard *a priori* error estimate indicates that the growth rate of the error is

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

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

[1] The actual computations were carried out by one Miss Ellen Fetter on a Royal McBee LGP-30 weighing in at 333kg, operating at 1500W, and capable of integrating the Lorenz system one time step per second. The size of the time step was $\Delta t = 0.01$ and the precision was six digits.

to machine precision ($\epsilon \sim 10^{-16}$ on most computers).[2] 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 solutions over time intervals of length $T = 25$ with any standard ODE solver.

In [9], 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 [19], the computability of the Lorenz system was further extended to $T = 48$ using high order ($\|e(T)\| \sim \Delta t^{30}$) finite element methods. This is the longest interval on which solutions have been published, as far as the authors are aware. As we shall see below, this is also 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 [4], but for unknown initial data. Other related work on high-precision numerical methods applied to the Lorenz system include [30] and [16]. For an overview of some recent results obtained with high-precision numerical methods, we also refer to [2].

In the present paper, we extend the computability of the Lorenz system to time $T = 1000$. Furthermore, we quantify the computability and try to 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 obtained simply by decreasing the size of the time step, see for example [27]. This has led to misconceptions regarding the computability of the Lorenz system, see for example [32]. To obtain a sequence of converging solutions, one must also control the effect of round-off errors. This was also noted by Lorenz in [23] as a response to [32].

2. Main results

We prove below that the error $\mathbf{E}$ in a computed numerical solution U approximating the exact solution u of the Lorenz system is the sum of three contributions,

$$\mathbf{E} = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C,$$

where $\mathbf{E}_D$ is the data error, which is nonzero if $U(0) \neq u(0)$; $\mathbf{E}_G$ is the discretization error, which is nonzero as a result of a finite time step; and $\mathbf{E}_C$ is the computational error, which is nonzero as a result of finite numerical precision. Furthermore, we bound each of the three contributions as the product of a stability factor and a residual that measures the size of local contributions to the error. The size of the residuals may be estimated in terms of the size of the time step. We find that

$$\mathbf{E} \sim S_D(T)\|U(0) - u(0)\| + S_G(T)\Delta t^p + S_C(T)\Delta t^{-1/2},$$

[2] 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 $[0, 1000]$.

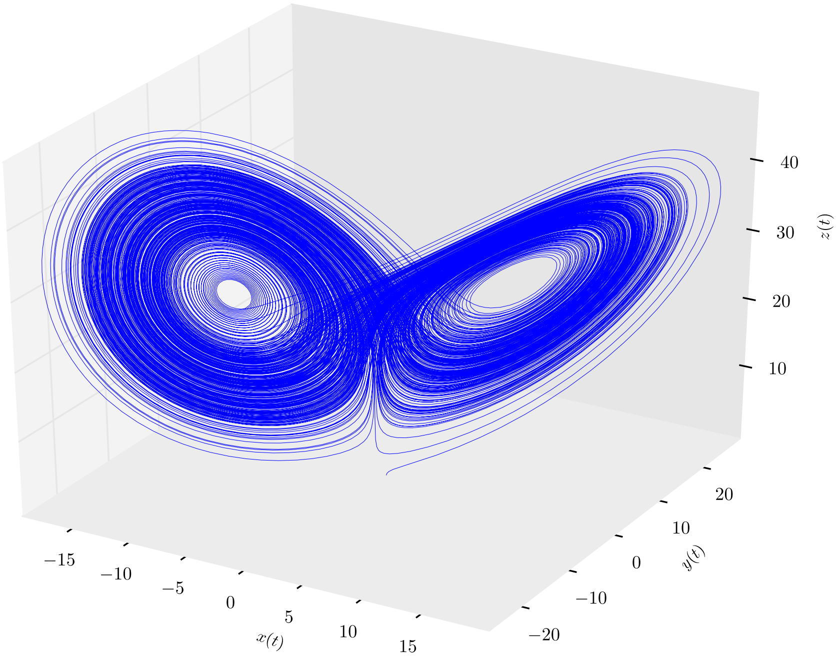

Figure 1. Phase portrait of the solution of the Lorenz system on the time interval $[0, 1000]$ for $u(0) = (1, 0, 0)$.

where Δt is the size of the time step, p the order of convergence of the numerical method, and $S_D(T)$, $S_G(T)$, $S_C(T)$ are stability factors that grow exponentially as function of the final time T with an approximate rate of $10^{0.4T}$.

We conclude that the computability of the Lorenz system is bounded by finite precision arithmetic at the time T when $\mathbf{E}_C = \mathbf{E}_G$. With 16 digits of precision, this limit occurs at $T \approx 50$. With 420 digits, the limit occurs at $T \approx 1050$. We compute a converging sequence of solutions up to order $p = 200$ and precision $\epsilon = 10^{-420}$ to obtain a solution of the Lorenz system accurate on the time interval $[0, 1000]$. We further demonstrate that the computability of the Lorenz system is given by

$$T \sim 2.5\, n_\epsilon,$$

where $n_\epsilon = \log_{10} \epsilon$ is the number of significant digits.

Figure 1 shows the familiar phase portrait of the Lorenz system. 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.

3. Error analysis

To analyze the computability of the Lorenz system, we prove an error estimate for the propagation of errors in numerical solutions of the Lorenz system. The proof follows the now well established techniques developed in [8, 7, 6, 3] but extensions are made to account for the accumulation of round-off errors.

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

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, $\bar{A}$ denotes the Jacobian matrix of the right-hand side f averaged over the approximate solution U and the exact solution u,

$$\bar{A}(t) = \int_0^1 \frac{\partial f}{\partial u}(sU(t) + (1-s)u(t), t) \, \mathrm{d}s. \tag{5}$$

By the chain rule, it follows that

$$\begin{aligned} \bar{A}(t)(U(t) - u(t)) &= \int_0^1 \frac{\partial f}{\partial u}(sU(t) + (1-s)u(t), t)(U(t) - u(t)) \, \mathrm{d}s \\ &= \int_0^1 \frac{\partial}{\partial s} f(sU(t) + (1-s)u(t), t) \, \mathrm{d}s = f(U(t), t) - f(u(t), t). \end{aligned}$$

Remark. The Lorenz system is quadratic in the primal variable u. Hence, the average (5) 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

$$\begin{cases} -\dot{\xi} = -\sigma\xi + (r - \bar{z})\eta + \bar{y}\zeta, \\ -\dot{\eta} = \sigma\xi - \eta + \bar{x}\zeta, \\ -\dot{\zeta} = -\bar{x}\eta - b\zeta, \end{cases} \tag{6}$$

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

The following error representation follows directly from the definition of the dual problem. It represents the error in an approximate solution U (computed by any numerical method) in terms of the residual R of the computed solution and the solution z of the dual problem (4). The only assumption we make on the numerical solution U is that it is piecewise smooth on a partition of the interval $[0, T]$ (or that it may be extended to such a function). At points where U is smooth, the residual is defined by

$$R(t) = \dot{U}(t) - f(U(t), t).$$

Theorem 3.1 (Error representation)**.** *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, the error $U(T) - u(T)$ may be represented by*

$$\langle z_T, U(T) - u(T)\rangle = \langle z(0), U(0) - u(0)\rangle + \sum_{m=1}^{M} \langle z(t_{m-1}), [U]_{m-1}\rangle + \int_0^T \langle z, R\rangle \,\mathrm{d}t,$$

where $R(t) = \dot{U}(t) - f(U(t),t)$ is the residual of the approximate solution U and $[U]_{m-1} = U(t_{m-1}^+) - U(t_{m-1}) = \lim_{t\to t_{m-1}^+} U(t) - U(t_{m-1})$.

Proof. By the definition of the dual problem, we find that

$$\langle z_T, e(T)\rangle = \langle z_T, e(T)\rangle - \int_0^T \langle \dot{z} + \bar{A}^\top z, e\rangle \,\mathrm{d}t = \langle z_T, e(T)\rangle - \sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \langle \dot{z} + \bar{A}^\top z, e\rangle \,\mathrm{d}t,$$

where $e = U - u$. Noting that $\langle \bar{A}^\top z, e\rangle = \langle z, \bar{A}e\rangle$ and integrating by parts, we obtain

$$\langle z_T, e(T)\rangle = \langle z(0), e(0)\rangle + \sum_{m=1}^{M} \langle z(t_{m-1}), [U]_{m-1}\rangle + \int_{t_{m-1}}^{t_m} \langle z, \dot{e} - \bar{A}e\rangle \,\mathrm{d}t,$$

where $[U]_{m-1} = U(t_{m-1}^+) - U(t_{m-1}^-) = U(t_{m-1}^+) - U(t_{m-1})$ denotes the jump of U at $t = t_{m-1}$. By the construction of $\bar{A}$, it follows that $\bar{A}e = f(U,\cdot) - f(u,\cdot)$. Hence, $\dot{e} - \bar{A}e = \dot{U} - f(U,\cdot) - \dot{u} + f(u,\cdot) = R$, which completes the proof. □

Remark. Theorem 3.1 holds for any piecewise smooth function $U : [0,T] \to \mathbb{R}^N$, in particular for any piecewise smooth extension of any approximate numerical solution obtained by any numerical method for (3).

We next investigate the contribution to the error in the computed numerical solution U from errors in initial data, numerical discretization, and computation (round-off errors),

$$\mathbf{E} = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C.$$

Here, $\mathbf{E}_D$ is the data error, which is nonzero if $U(0) \neq u(0)$; $\mathbf{E}_G$ is the discretization error (G for Galerkin), which is nonzero as a result of a finite time step; and $\mathbf{E}_C$ is the computational error, which is nonzero as a result of finite numerical precision.

To estimate the computational error, we introduce the *discrete residual* $\bar{R}$ defined as follows. For $p \geq 0$, let $\{\lambda_k\}_{k=0}^p$ be the Lagrange nodal basis for $\mathcal{P}^p([0,1])$, the space of polynomials of degree $\leq p$ on $[0,1]$, on a partition $0 \leq \tau_0 < \tau_1 < \cdots < \tau_p \leq 1$ of $[0,1]$, that is, $\mathrm{span}\{\lambda_k\}_{k=0}^p = \mathcal{P}^p([0,1])$ and $\lambda_i(\tau_j) = \delta_{ij}$. Then, the discrete residual $\bar{R}_m$ is defined on each interval $(t_{m-1},t_m]$ by

$$\bar{R}_m^k = \lambda_k(0)[U]_{m-1} + \int_{t_{m-1}}^{t_m} \lambda_k((t - t_{m-1})/\Delta t_m)R(t)\,\mathrm{d}t, \quad k = 0,1,\ldots,p. \tag{7}$$

We also define the corresponding interpolation operator π onto the space of piecewise polynomial functions on the partition $0 = t_0 < t_1 < \cdots < t_M = T$ by

$$\pi v(t) = \sum_{k=0}^{p} v(t_{m-1} + \tau_k \Delta t_m)\,\lambda_k((t - t_{m-1})/\Delta t_m), \quad t \in (t_{m-1},t_m].$$

We may now prove the following a posteriori error estimate.

Theorem 3.2 (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

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

Proof. Starting from the error representation of Theorem 3.1, we add and subtract the degree p left-continuous piecewise polynomial interpolant πz defined above to obtain

$$\begin{aligned}
\langle z_T, e(T)\rangle &= \langle z(0), e(0)\rangle\\
&\quad + \sum_{m=1}^{M} \langle z(t_{m-1}) - \pi z(t_{m-1}^+), [U]_{m-1}\rangle + \int_{t_{m-1}}^{t_m} \langle z - \pi z, R\rangle\,\mathrm{d}t\\
&\quad + \sum_{m=1}^{M} \langle \pi z(t_{m-1}^+), [U]_{m-1}\rangle + \int_{t_{m-1}}^{t_m} \langle \pi z, R\rangle\,\mathrm{d}t\\
&\equiv \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C.
\end{aligned}$$

We first note that the data error $\mathbf{E}_D$ may be bounded by $\|z(0)\|\,\|e(0)\| \equiv S_D\,\|e(0)\|$. By an interpolation estimate, we may estimate the discretization error $\mathbf{E}_G$ as follows:

$$\begin{aligned}
\mathbf{E}_G &\leq \sum_{m=1}^{M} \|z(t_{m-1}) - \pi z(t_{m-1}^+)\|\,\|[U]_{m-1}\| + \int_{t_{m-1}}^{t_m} \|z - \pi z\|\,\|R\|\,\mathrm{d}t\\
&\leq C_p \max_{[0,T]} \left\{\Delta t^{p+1}(\|[U]\|/\Delta t + \|R\|)\right\} \sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \|z^{(p+1)}\|\,\mathrm{d}t,
\end{aligned}$$

where $\sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \|z^{(p+1)}\|\,\mathrm{d}t = \int_0^T \|z^{(p+1)}\|\,\mathrm{d}t \equiv S_G$ and C_p is an interpolation constant. Finally, to estimate the computational error, we expand πz in the nodal basis to obtain

$$\begin{aligned}
\mathbf{E}_C &= \sum_{m=1}^{M} \sum_{k=0}^{p} \left\langle z(t_{m-1}+\tau_k \Delta t_m), \lambda_k(0)[U]_{m-1} + \int_{t_{m-1}}^{t_m} \lambda_k((t-t_{m-1})/\Delta t_m) R(t)\,\mathrm{d}t \right\rangle \\
&= \sum_{m=1}^{M} \sum_{k=0}^{p} \langle z(t_{m-1}+\tau_k \Delta t_m), \bar{R}_m^k \rangle = \sum_{m=1}^{M} \Delta t_m \sum_{k=0}^{p} \langle z(t_{m-1}+\tau_k \Delta t_m), \Delta t_m^{-1} \bar{R}_m^k \rangle \\
&\le \sum_{m=1}^{M} \Delta t_m \sum_{k=0}^{p} \|z(t_{m-1}+\tau_k \Delta t_m)\| \, \|\Delta t_m^{-1} \bar{R}_m^k\| \\
&\le \max_{[0,T]} \|\Delta t^{-1} \bar{R}\| \sum_{m=1}^{M} \Delta t_m \sum_{k=0}^{p} \|z(t_{m-1}+\tau_k \Delta t_m)\| \\
&\le C_p' \max_{[0,T]} \|\Delta t^{-1} \bar{R}\| \sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \|\pi z\|\,\mathrm{d}t,
\end{aligned}$$

where $\sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \|\pi z\|\,\mathrm{d}t = \int_0^T \|\pi z\|\,\mathrm{d}t \equiv S_C$ and C_p' is a constant depending only on p. □

Remark. Theorem 3.2 estimates the size of $\langle z_T, U(T) - u(T)\rangle$ for any given vector z_T. We may thus estimate any bounded linear functional of the error at the final time by choosing z_T as the corresponding Riesz representer. In particular, we may estimate the error in any component $u_i(T)$ of the solution by setting z_T to the ith unit vector for $i = 1, 2, \ldots, N$.

Theorem 3.2 extends standard a posteriori error estimates for systems of ordinary differential equations in two ways. First, it does not make any assumption on the underlying numerical method. Second, it includes the effect of numerical round-off errors. A similar estimate can be found in [18] but only for the simplest case of the piecewise linear cG(1) method (Crank–Nicolson).

We now investigate the propagation of numerical round-off errors in more detail. In the following, $\mathbf{E}_C$ denotes the computational error defined by

$$\mathbf{E}_C = \sum_{m=1}^{M} \langle \pi z(t_{m-1}^+), [U]_{m-1}\rangle + \int_{t_{m-1}}^{t_m} \langle \pi z, R\rangle\,\mathrm{d}t. \tag{8}$$

Theorem 3.2 bounds the computational error in terms of the discrete residual defined in (7). The discrete residual tests the continuous residual $R = \dot{U} - f$ of (3) against polynomials of degree p. In particular, it tests how well the numerical method satisfies the relation

$$U(t_m) = U(t_{m-1}) + \int_{t_{m-1}}^{t_m} f(U, \cdot)\,\mathrm{d}t. \tag{9}$$

With a machine precision of size ϵ_{mach}, our best hope is that the numerical method satisfies (9) to within a tolerance of size $\epsilon > \epsilon_{\mathrm{mach}}$ for each component of the vector U. It follows by the Cauchy–Schwarz inequality that

$$\max_{m,k} \|\bar{R}_m^k\| \le \epsilon\sqrt{N}.$$

We thus have the following corollary.

Corollary 3.3. *The computational error $\mathbf{E}_C$ of Theorem 3.2 is bounded by*

$$|\mathbf{E}_C| \leq S_C \, C_p' \frac{\epsilon \sqrt{N}}{\min_{[0,T]} \Delta t}.$$

This indicates that the computational error scales like Δt^{-1}; the smaller the time step, the larger the computational error. This explains why it is difficult to compute a converging sequence of solutions for the Lorenz system and other chaotic systems. At first, this seems non-intuitive, but it is a simple consequence of the fact that a smaller time step leads to a larger number of time steps and thus a larger number of round-off errors. However, it does not mean that one cannot obtain a converging sequence of solutions to the Lorenz system as is commonly believed. Converging solutions can be obtained as long as the discrete residual is kept under control, that is, if $\mathbf{E}_C \leq \mathbf{E}_G$.

Furthermore, the estimate of Corollary 3.3 is overly pessimistic. It is based on the assumption that round-off errors accumulate without cancellation. In practice, the round-off error is sometimes positive and sometimes negative. As a simple model, we make the assumption that the round-off error is a random variable which takes the value $+\epsilon$ or $-\epsilon$ with equal probabilities,

$$(\bar{R}_m^k)_i = \begin{cases} +\epsilon, & p = 0.5, \\ -\epsilon, & p = 0.5, \end{cases} \tag{10}$$

for all m, k, i. Under this assumption, we find that the expected size of the computational error scales like $\Delta t^{-1/2}$. As we shall see in the next section, this is also confirmed by numerical experiments.

In reality, round-off errors are not uncorrelated random variables, but the simple model (10) may still give useful results. For a discussion on the applicability of random models to the propagation of round-off errors, see [12] (Section 2.8).

Theorem 3.4. *Assume that the round-off error is a random variable of size $\pm\epsilon$ with equal probabilities. Then the root-mean squared expected computational error $\mathbf{E}_C$ of Theorem 3.2 is bounded by*

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

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.

Proof. As in the proof of Theorem 3.2, we obtain

$$\mathbf{E}_C = \sum_{m=1}^{M} \sum_{k=0}^{p} \langle z(t_{m-1} + \tau_k \Delta t_m), \bar{R}_m^k \rangle = \sum_{m=1}^{M} \sum_{k=0}^{p} \sum_{i=1}^{N} z_i(t_{m-1} + \tau_k \Delta t_m)(\bar{R}_m^k)_i,$$

where by assumption $(\bar{R}_m^k)_i = \epsilon x_{mki}$ and $x_{mki} \pm 1$ with probability 0.5 and 0.5 respectively. It follows that

$$\begin{aligned}\mathbf{E}_C^2 &= \sum_{m,n=1}^{M} \sum_{k,l=0}^{p} \sum_{i,j=1}^{N} z_i(t_{m-1} + \tau_k \Delta t_m) z_j(t_{n-1} + \tau_l \Delta t_n)\, \epsilon^2 x_{mki} x_{nlj} \\ &= \sum_{(m,k,i)=(n,l,j)} z_i^2(t_{m-1} + \tau_k \Delta t_m)\, \epsilon^2 x_{mki}^2 \\ &+ \sum_{(m,k,i)\neq(n,l,j)} z_i(t_{m-1} + \tau_k \Delta t_m) z_j(t_{n-1} + \tau_l \Delta t_n)\, \epsilon^2 x_{mki} x_{nlj}.\end{aligned}$$

We now note that $x_{mki}^2 = 1$. Furthermore, $y_{ijklmn} = x_{mki} x_{nlj}$ is a random variable which takes the values $+1$ and -1 with equal probabilities. We thus find that

$$\begin{aligned}E[\mathbf{E}_C^2] &= \epsilon^2 \sum_{m=1}^{M} \sum_{k=0}^{p} \sum_{i=1}^{N} z_i^2(t_{m-1} + \tau_k \Delta t_m) + 0 = \epsilon^2 \sum_{m=1}^{M} \sum_{k=0}^{p} \|z(t_{m-1} + \tau_k \Delta t_m)\|^2 \\ &\leq \frac{\epsilon^2}{\min_{[0,T]} \Delta t} \sum_{m=1}^{M} \Delta t_m \sum_{k=0}^{p} \|z(t_{m-1} + \tau_k \Delta t_m)\|^2 \leq S_{C_2}^2\, C_p' \frac{\epsilon^2}{\min_{[0,T]} \Delta t},\end{aligned}$$

where $S_{C_2} = \left(\int_0^T \|\pi z\|^2 \,\mathrm{d}t\right)^{1/2}$. This completes the proof. □

Remark. By additional assumptions on the smoothness of the dual solution z, one may relate the error to the expectation value of the distance from the starting point for a random walk which is $\sqrt{n}$ for n steps (for n large), and prove a similar estimate for the expected absolute value of the computational error, $E[|\mathbf{E}_C|] \sim S_C \epsilon/\sqrt{\Delta t}$, where $S_C = \int_0^T \|\pi z\| \,\mathrm{d}t$.

Remark. By Cauchy–Schwarz, the stability factor S_C of Theorem 3.2 is bounded by $\sqrt{T} S_{C_2}$. For the Lorenz system, the two stability factors grow at equal rates with $S_{C_2} \sim 3.3\, S_C$.

We conclude this section by discussing how the above error estimates apply to the particular methods used in this work. The estimate of Theorem 3.2 is valid for any numerical method but is of particular interest as an *a posteriori* error estimate for the finite element methods cG(q) and dG(q) (see [13, 14, 15, 5]).

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

$$\int_{t_{n-1}}^{t_n} v\, R \,\mathrm{d}t = 0 \tag{11}$$

for all $v \in \mathcal{P}^{q-1}([t_{n-1}, t_n])$. It follows that the discrete residual (7) is zero if $p = q-1$. However, this is only true in exact arithmetic. In practice, the discrete residual is nonzero and measures how well we solve the cG(q) equations (11), including round-off errors and errors from numerical quadrature.[3] For the cG(q) method, we further expect the residual to converge as Δt^q. Thus, choosing $p = q - 1$ in Theorem 3.2, one may expect the error for the cG(q) method to scale as

$$E = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C \sim \epsilon + \Delta t^{2q} + \Delta t^{-1/2}\epsilon.$$

4. Numerical results

In this section, we present numerical results in support of Theorem 3.2 and Theorem 3.4. The results were obtained using the finite element package DOLFIN [20] version 0.9.2 together with the multi-precision library GMP [10]. For a detailed discussion on the implementation, we refer to [17]. The source code as well as scripts to reproduce all results presented in this manuscript are made available at `http://www.lorenzsystem.net/`.

4.1. **Solution of the Lorenz system.** In Figure 2, we plot the solution of the Lorenz system on the interval $[0, 1000]$. The corresponding phase portrait 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[4], and a tolerance for the discrete residual of size $\epsilon \approx 2.26 \cdot 10^{-424}$.

The very rapid (exponential) accumulation of numerical errors makes the Lorenz "fingerprint" displayed in Figure 2 useful as a reference for verification of solutions of the Lorenz system. If a solution is only slightly wrong, the error is quickly magnified so that the error becomes visible by a direct inspection of a plot of the solution. In fact, as will be demonstrated below, errors accumulate at a rate of approximately a factor $10^{0.4} \approx 2.5$ per unit time.

Some interesting statistics may be computed from the solution displayed in Figures 1 and 2. One such statistic is shown in Figure 3 where we plot the fraction of time spent around each of the two unstable fixed points $P_{\pm} = (\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27)$. We define the fraction of time $\alpha_1(t)$ spent around the fixed point P_+ by

$$\alpha_1(t) = \frac{1}{t}\int_0^t \chi_1(s)\,\mathrm{d}s,$$

where

$$\chi_1(t) = \begin{cases} 1, & \text{if } \|u(t) - P_+\| < \|u(t) - P_-\|, \\ 0, & \text{if } \|u(t) - P_+\| \geq \|u(t) - P_-\|. \end{cases}$$

We define $\alpha_2(t)$ analogously. The result seems to indicate that over long time, equal time is spent around each of the two fixed points.

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 3.2, 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

[3] To account for additional quadrature errors present if the integral of (11) is approximated by quadrature, one may add and subtract an interpolant πf of the right-hand side f in the proof of Theorem 3.2 to obtain an additional term $\mathbf{E}_Q = S_Q \max_{[0,T]} \|\pi f - f\|$ where $S_Q = \int_0^T \|z\|\,\mathrm{d}t \approx S_C$.

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

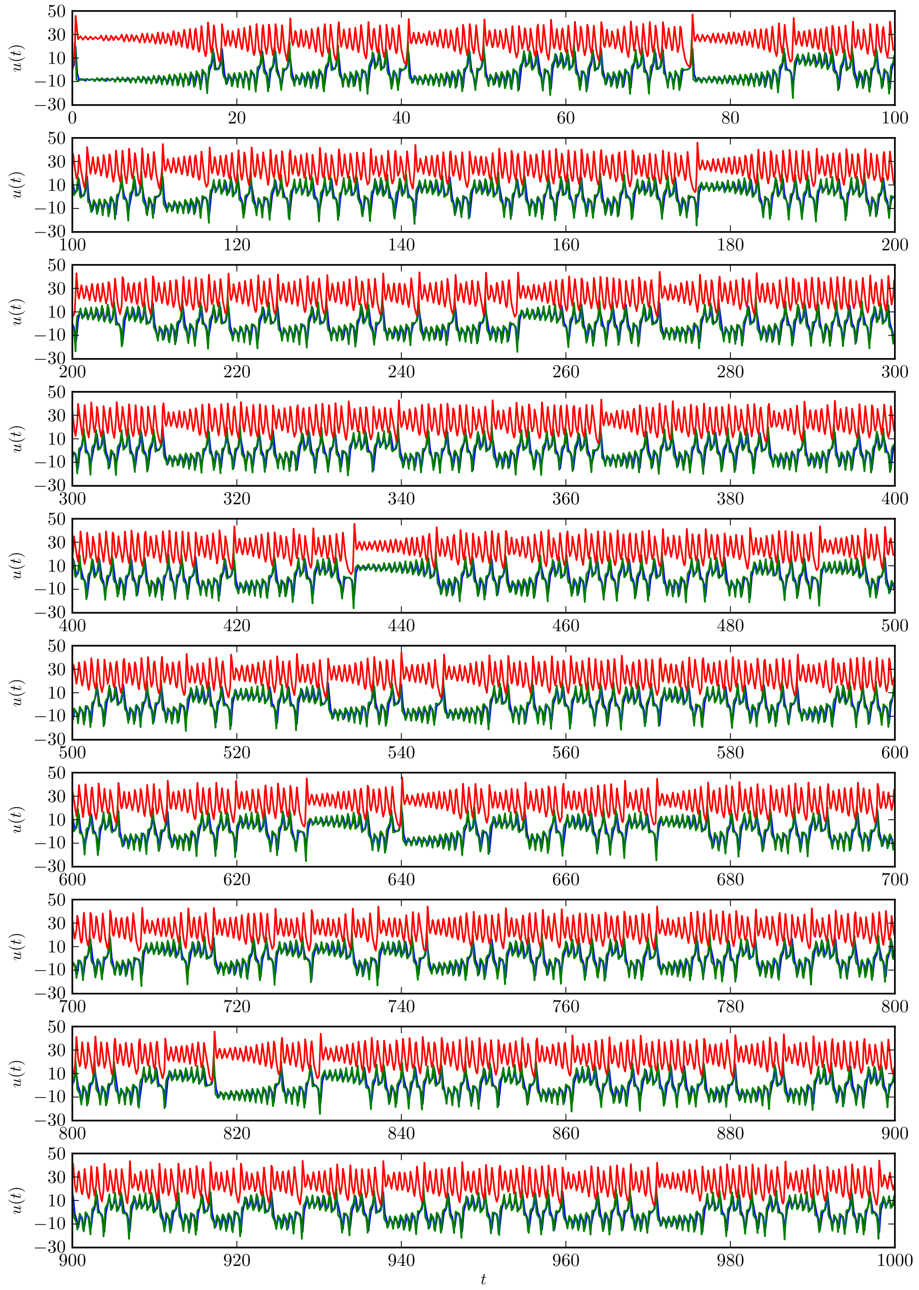

Figure 2. The three components of the Lorenz system on the interval $[0, 1000]$ with the x and y components plotted in blue and green respectively (and almost overlaid) and the z component in red.

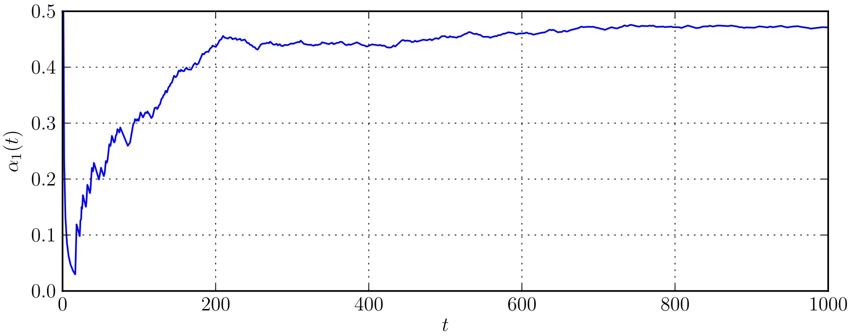

FIGURE 3. Fraction of time $\alpha_1(t)$ spent around the first of the two fixed points $P_\pm = (\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27)$.

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 4. 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 3.2 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 3.2 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 [16]. As a reference, we include values of the solution of the Lorenz system with 16 digits on the interval $[0, 1000]$ in an appendix of this paper. The full reference solution is available at `http://www.lorenzsystem.net/`.

4.2. **Dual solution and stability factors.** The solution of the dual problem of the Lorenz system on the interval $[0, 1000]$ is plotted in Figure 5. 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 (3.2), 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 6. Note that each data point (T, S) in Figure 6 corresponds to a solution of the dual problem on the interval $[0, T]$.

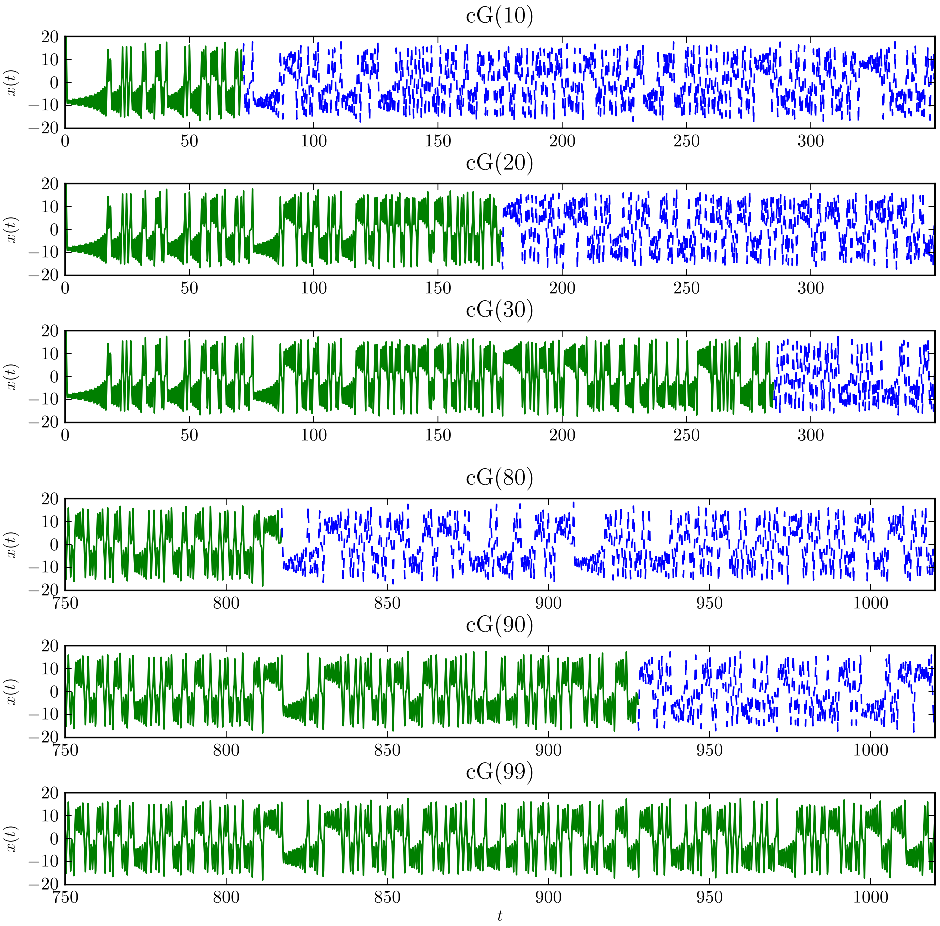

FIGURE 4. Computed numerical solutions for the Lorenz system with methods of increasing order, starting at cG(10) (a method of order 20) and increasing up to cG(99) (a method of order 198).

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 $T = 1000$.

However, the linearity of the dual problem and the small size of the Lorenz system makes it possible to store the solution operators $z(T) \mapsto z(t_n)$ for a partition $0 = t_0 < t_1 < \cdots < t_M = T$ as a sequence of 3×3 matrices and reuse those solution operators when solving the dual problem on the interval $[0, T + \Delta T]$. This procedure is described in more detail in [19, 17].

By Figure 6, 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

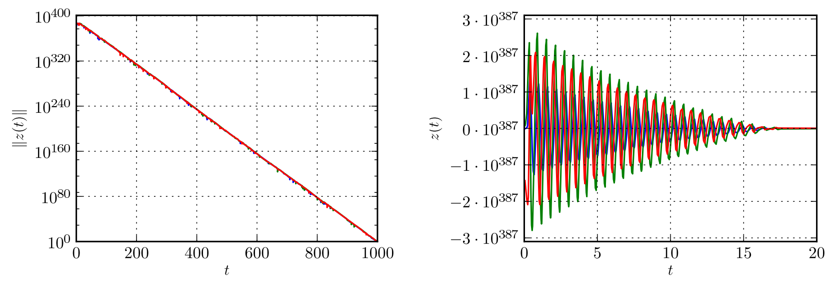

Figure 5. A logarithmic plot of the absolute value of the dual solution of the Lorenz system on the time interval $[0, 1000]$ (left) and a detail of the solution on the time interval $[0, 20]$ (right). The dual solution was computed with final time data $z_T = (1, 0, 0)$.

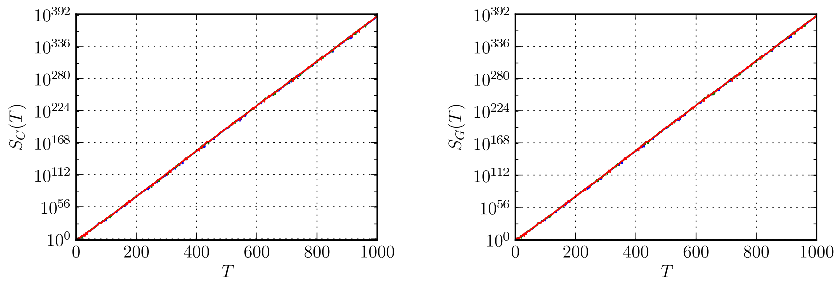

Figure 6. Growth of the stability factors S_C (left) and S_G (right) for the Lorenz system on the time interval $[0, 1000]$. The stability factor $S_G = \int_0^T \|\dot{z}\| \, \mathrm{d}t$ is the discretization stability factor for the cG(1) method.

by

$$S(T) \sim 10^{0.388T} \sim 10^{0.4T}. \tag{12}$$

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 7) explains the discrepancy between the two estimates. The growth rate of the 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.

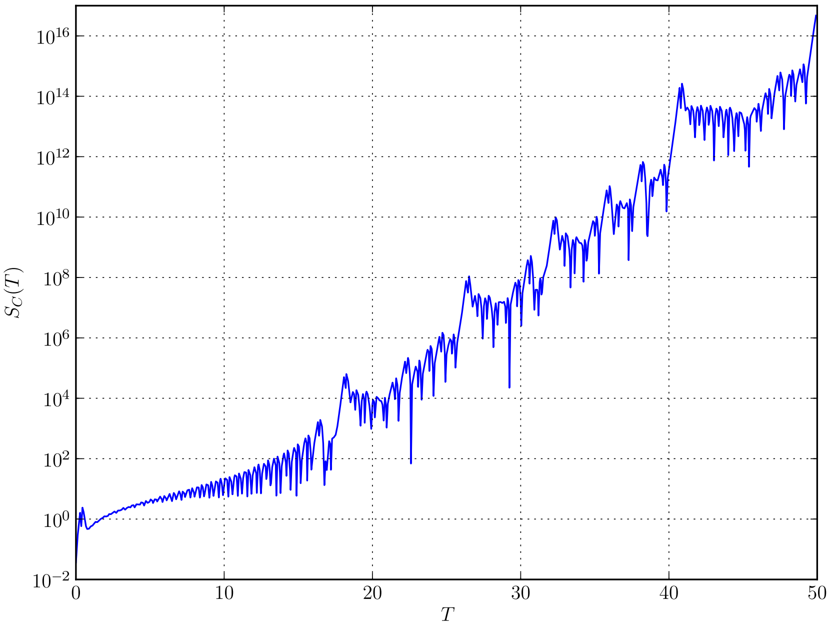

FIGURE 7. Growth of the stability factor S_C accounting for accumulation of round-off errors on the time interval $[0, 50]$.

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 is confirmed by the results presented in Figure 8, 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 simple 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 simple 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,

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

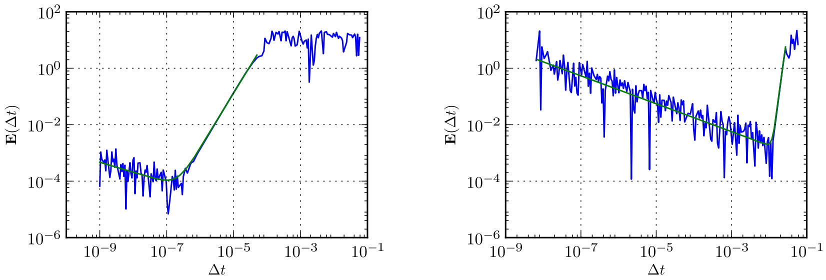

FIGURE 8. Error at time $T = 30$ for the cG(1) solution (left) and at time $T = 40$ for the cG(5) solution (right) of the Lorenz system. The slopes of the green lines are $-0.35 \approx -1/2$ and $1.95 \approx 2$ for the cG(1) method. For the cG(5) method, the slopes are $-0.49 \approx -1/2$ and $10.00 \approx 10$.

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 $T = 40$.

To determine the constants α, β, $C_1^{[q]}$, $C_2^{[q]}$, and $C_3^{[q]}$, we repeat the experiment of Figure 8 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 3.4.

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 3.4). 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:

$$
\mathbf{E} \approx \Big[0.5\,\|U(0) - u(0)\| + 0.001\Delta t^{2q} + (0.002 + 0.0005q)\Delta t^{-1/2}\epsilon)\Big] \cdot 10^{0.388T}. \tag{13}
$$

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.

5.2. **Optimal time step.** Based on the model (13), we determine an estimate of the optimal time step size by setting $\mathbf{E}_G = \mathbf{E}_C$. We find that

$$\Delta t = ((2 + 0.5q)\epsilon)^{\frac{1}{2q+1/2}} \approx \epsilon^{\frac{1}{2q+1/2}} \tag{14}$$

for large values of q. Inserting the values $\epsilon = 10^{-420}$ 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 (14) into (13). Neglecting data errors, that is, assuming $U(0) = u(0)$, we find that

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

for large values of q. Let $n_\epsilon = \log_{10} \epsilon$ be the number of significant digits. It follows that $\mathbf{E} \approx 0.002 \cdot 10^{0.4T - n_\epsilon}$. We conclude that the time T at which the solution is no longer accurate is

$$T = n_\epsilon / 0.4 = 2.5\, n_\epsilon.$$

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

$$\mathbf{E} \approx 0.001 \epsilon\, S_C(T).$$

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

6. Conclusions

We have demonstrated that the Lorenz system is computable over long time intervals by high order and high precision numerics. In particular, we have demonstrated that the solution is computable on the time interval $[0, 1000]$. Increasing the number of significant digits beyond the 420 digits used in this work increases the interval on which the Lorenz solution is computable. In fact, we have demonstrated that the size of the time interval on which the solution is computable scales linearly with the number of digits, $T \sim 2.5\, n_\epsilon$. Thus, if a precision of 840 digits is used, one may compute the solution on $[0, 2000]$ and if a precision of 4200 digits is used, one may compute the solution on $[0, 10000]$.

Furthermore, we have derived a quantitative global error estimate for the Lorenz system (and other systems of ordinary differential equations) that includes the effect of errors in initial data, discretization errors, and computational (round-off) errors. The results of numerical experiments agree well with the theoretical predictions based on this estimate.

References

[1] V. S. Afraimovich, V. V. Bykov, and L. P. Shilikov, *On the appearance and structure of the Lorenz attractor*, in Dokl. Acad. Sci. USSR, vol. 234, 1977, p. 6.

[2] D. H. Bailey, R. Barrio, and J. M. Borwein, *High-precision computation: Mathematical physics and dynamics*, Submitted to SIAM Review, (2010).

[3] R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica, 10 (2001), pp. 1–102.

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

[5] M. Delfour, W. Hager, and F. Trochu, *Discontinuous Galerkin methods for ordinary differential equations*, Math. Comp., 36 (1981), pp. 455–473.

[6] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numerica, 4 (1995), pp. 105–158.

[7] D. Estep, *A posteriori error bounds and global error control for approximations of ordinary differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 1–48.

[8] D. Estep and D. French, *Global error control for the continuous Galerkin finite element method for ordinary differential equations*, M2AN, 28 (1994), pp. 815–852.

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

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

[11] J. Guckenheimer and R. F. Williams, *Structural stability of Lorenz attractors*, Publications Mathmatiques de l'IHS, 50 (1979), pp. 59–72.

[12] N. Higham, *Accuracy and stability of numerical algorithms*, Society for Industrial Mathematics, second ed., 2002.

[13] B. L. Hulme, *Discrete Galerkin and related one-step methods for ordinary differential equations*, Math. Comput., 26 (1972), pp. 881–891.

[14] ———, *One-step piecewise polynomial Galerkin methods for initial value problems*, Math. Comput., 26 (1972), pp. 415–426.

[15] C. Johnson, *Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations*, SIAM J. Numer. Anal., 25 (1988), pp. 908–926.

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

[17] B. 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.

[18] A. Logg, *Multi-Adaptive Galerkin methods for ODEs I*, SIAM J. Sci. Comput., 24 (2003), pp. 1879–1902.

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

[20] A. Logg and G. N. Wells, *DOLFIN: Automated finite element computing*, ACM Transactions on Mathematical Software, 37 (2010), pp. 20:1–20:28.

[21] E. N. Lorenz, *Maximum simplification of the dynamic equations*, Tellus, 12 (1960), pp. 243–254.

[22] ———, *Deterministic nonperiodic flow*, J. Atmosph. Sci., 20 (1963), pp. 130–141.

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

[24] L. Rayleigh, *On convective currents in a horizontal layer of fluid when the higher temperature is on the under side*, Phil. Mag., 32 (1916), pp. 529–546.

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

[26] S. Smale, *Mathematical problems for the next century*, The Mathematical Intelligencer, 20 (1998), pp. 7–15.

[27] J. Teixeira, C. 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 (2007), pp. 175–189.

[28] W. Tucker, *The Lorenz attractor exists*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 328 (1999), pp. 1197–1202.

[29] M. Viana, *What's new on Lorenz strange attractors?*, The Mathematical Intelligencer, 22 (2000), pp. 6–19.

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

[31] R. F. Williams, *The structure of Lorenz attractors*, Publications Mathematiques de l'IHES, 50 (1979), pp. 73–99.

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

Appendix A. Reference solution on $[0, 1000]$

t	x	y	z
0	1	0	0
1	−9.408450567056036	−9.096199071186479	28.581627624392340
2	−7.876082549991658	−8.761621817314939	24.990260995565926
3	−8.143999245434069	−6.942058961947722	28.120426584688346
4	−9.453542010189011	−10.430214212303325	26.938025375071280
5	−6.974570472684818	−7.021060890822531	25.119616492127594
10	−5.857685382424090	−5.831082486426101	23.932132987027561
15	−10.307035782050431	−4.451151241384303	35.094663385722221
20	−8.021143613317436	−11.905464749171751	19.856374858398414
25	−0.960995516331200	−2.395630257446737	20.215326486394101
30	−3.892637337379485	0.274019816217374	27.866107798922574
35	−13.160033671307707	−17.934308953754190	27.389641766647930
40	−0.194731422418737	0.181102990970540	17.234465691306429
45	−4.209878880668230	−4.949651517445130	20.445977088053489
50	16.246275544569393	23.171100649775568	30.162707065608277
100	12.286006677018985	16.581464262811753	26.597525090160012
150	3.714163850157513	4.963799440808135	18.356814911871552
200	−0.428878362621397	−0.812388566743750	8.141562045191055
250	2.502814098879616	4.820514815351314	7.743299823079606
300	−7.946536271496051	−1.159949771727215	33.524744262421997
350	9.112823629247304	14.898280249726538	17.891171661916864
400	5.943245586080340	9.672638471715088	16.076376870383008
450	−6.180721929981728	−10.835782455846392	13.289688415684369
500	−11.118772890892362	−15.011144913050707	25.223573344110680
550	−3.035500113897037	0.579934174022604	26.679609648806029
600	13.982585780713302	12.606312490063734	35.686835498440018
650	−0.731432166791398	0.267894604554171	20.518258634657265
700	−7.974669839553495	−2.477557738987132	32.428427138140691
750	−5.483956851307489	−9.590492702691833	13.012815684212407
800	0.882587574884531	1.111923175877187	15.724886069017078
850	9.522594933374956	14.745161623188793	20.068031706266400
900	15.450276601479965	21.458347336788307	29.834655787376352
950	3.491185243251774	6.231329755516670	11.366593194327329
1000	12.537229584422063	12.842013118843248	31.953398729274273