



A posteriori error analysis of round-off errors in the numerical solution of ordinary differential equations


Downloaded from: <https://research.chalmers.se>, 2025-12-04 23:24 UTC

Citation for the original published paper (version of record):

Kehlet, B., Logg, A. (2017). A posteriori error analysis of round-off errors in the numerical solution of ordinary differential equations. Numerical Algorithms, 76(1): 191-210.
<http://dx.doi.org/10.1007/s11075-016-0250-4>

N.B. When citing this work, cite the original published paper.

A posteriori error analysis of round-off errors in the numerical solution of ordinary differential equations

Benjamin Kehlet^{1,2} · Anders Logg^{3,4} 

Received: 30 November 2015 / Accepted: 2 December 2016 / Published online: 3 January 2017
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract We prove sharp, computable error estimates for the propagation of errors in the numerical solution of ordinary differential equations. The new estimates extend previous estimates of the influence of data errors and discretization errors with a new term accounting for the propagation of numerical round-off errors, showing that the accumulated round-off error is inversely proportional to the square root of the step size. As a consequence, the numeric precision eventually sets the limit for the pointwise computability of accurate solutions of any ODE. The theoretical results are supported by numerically computed solutions and error estimates for the Lorenz system and the van der Pol oscillator.

Keywords Computability · High precision · High order · High accuracy · Probabilistic error propagation · Long-time integration · Finite element · Time-stepping · A posteriori · Lorenz · Van der Pol

✉ Anders Logg
logg@chalmers.se
Benjamin Kehlet
benjamik@simula.no

¹ University of Oslo, Oslo, Norway

² Simula Research Laboratory, P.O.Box 134, 1325, Lysaker, Norway

³ Department of Mathematical Sciences, Chalmers University of Technology, SE-41296, Gothenburg, Sweden

⁴ University of Gothenburg, Gothenburg, Sweden

1 Introduction

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

$$\begin{aligned}\dot{u}(t) &= f(u(t), t), \quad t \in (0, T], \\ u(0) &= u_0,\end{aligned}\tag{1}$$

where 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 a quantity of interest (functional) computed from an approximate solution $U : [0, T] \rightarrow \mathbb{R}^N$ computed by a single-step numerical method, such as an explicit or implicit Runge–Kutta method. For the numerical results presented at the end of this work, we have used a particular time-stepping method formulated as a Galerkin finite element method, which, for any particular choice of finite element basis and quadrature, will correspond to a particular implicit Runge–Kutta method. We stress that as a result of the generality of cG/dG Galerkin time-stepping formulations, the analysis applies to a wide range of single-step methods, in particular interpolatory implicit Runge–Kutta methods; see [1].

The propagation of local errors and accumulation of global errors in the numerical solution of ODE have been studied extensively in the literature, see e.g. [3, 7–9, 11]. These estimates are based on the formulation of an auxiliary dual problem: the linearised adjoint problem. From the solution of the dual problem, the accumulation rate of local errors may be computed, either as global stability factors or as local stability weights. These factors or weights, together with a measure of the local error, typically the residual $R(t) = \dot{U} - f(U(t), t)$ lead to a computable estimate of the global error.

Standard estimates may include various sources contributing to the global error, such as discretization errors, accounting for the use of finite time steps, quadrature errors, accounting for the approximation of the right-hand side f by a particular quadrature rule, and data errors, accounting for the approximation of the initial value u_0 . In this work, we extend these estimates by adding a new term accounting for the use of finite numeric precision in the computation of the numerical solution. This error is normally neglected, since it is typically much smaller than the contribution from the data or discretization error. However, when the system (1) is very sensitive to perturbations, when the time interval $[0, T]$ is very long, or when a solution is sought with very high accuracy, the effect of numerical round-off errors as a result of finite numeric precision can and will be the dominating error source, which ultimately limits the computability of a given problem.

2 Main results

We prove that the global error \mathbf{E} , defined below as a linear functional of the error $U(T) - u(T)$ at final time T , in a computed numerical solution U approximating the exact solution u of the ODE (1) 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 discretisation 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 which 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

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

where Δt is the size of the time step, r the order of convergence of the numerical method, and $S_D(T)$, $S_G(T)$, $S_C(T)$ are stability factors which can be computed a posteriori.

This estimate shows in particular that the size of the global error is determined in competition between the term $\mathbf{E}_G \sim \Delta t^r$, which decreases when the time step is reduced, and the term $\mathbf{E}_C \sim \Delta t^{-1/2}$, which *increases* when the time step is reduced. This has three effects:

- (i) there is an optimal step size or step size regime where none of the terms is dominating;
- (ii) for a given numerical precision, the computability of the system (1) is limited by the size of the minimum/optimal step size;
- (iii) the highest accuracy (smallest error) will be obtained by a high-order method which can achieve a small discretisation error \mathbf{E}_G for a relatively large step size, yielding a small computational error \mathbf{E}_C .

The points (i)–(ii) are illustrated in Fig. 1.

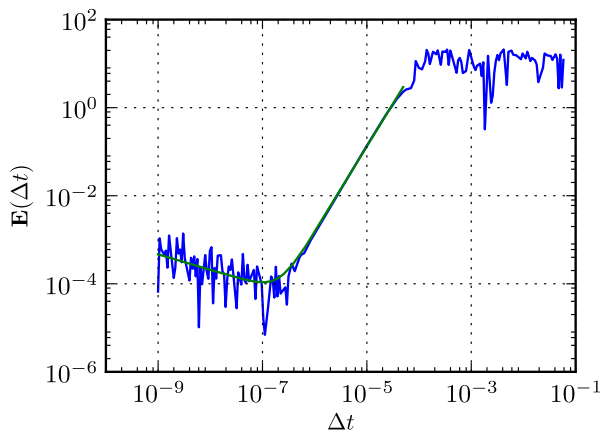


Fig. 1 The global error is initially reduced when the time step Δt is reduced but then starts to increase as a result of accumulated round-off errors; see Section 4.1.2 for details

3 Error analysis

Our error analysis is based on the solution of an auxiliary *dual problem* and follows the techniques developed in [7–9] and [2], with extensions to account for the accumulation of round-off errors. The key ideas of the dual-weighted approach to error estimation developed in the above references are:

- expression of the global error, or the error in a global functional of the computed solution, in terms of the residual $R = \dot{U} - f(U(t), t)$ of a computed solution U and the solution z of the auxiliary dual problem;
- approximate (numerical) solution of the dual problem;
- estimation of the global error in terms of the residual R and a computed approximation z_h of the dual solution z .

Challenges involved in the dual-weighted approach involve appropriate choice of initial data for the dual problem and numerical strategies for efficient (low-cost) solution of the dual problem. For a detailed discussion, we refer to references [7–9] and [2].

3.1 Sketch of proof

We first sketch out the main ideas of our analysis in the case of a linear system $Au = b$, where $A \in \mathbb{R}^{N \times N}$ and $u, b \in \mathbb{R}^N$. This sketch presents the essential ideas without unnecessary technical complications. We then return to the analysis of the system of ordinary differential (1). For a more detailed account, see for example [12].

Let u be the exact solution of the linear system $Au = b$ and let $U \approx u$ be an approximate solution with residual $R = AU - b \neq 0$. Our aim is to express the error $e = U - u$ in terms of the residual R .

Introduce the dual problem

$$A^\top z = \psi,$$

where A^\top denotes the transpose (adjoint) of the matrix A , z denotes the *dual solution* and $\psi \in \mathbb{R}^N$ is a given vector. It then follows that

$$(\psi, e) = (A^\top z, e) = (z, Ae) = (z, AU - Au) = (z, AU - b) = (z, R). \quad (2)$$

This *error representation* expresses any linear functional (represented by its Riesz representer ψ) of the error e in terms of the residual R via the dual solution z . Note that different linear functionals result in different data ψ for the dual problem and thereby different dual solutions z . Note also that the error representation (2) is valid independently of which numerical method is used to compute the approximation U . Assume now further that the numerical method can be formulated as a Galerkin (or Petrov-Galerkin) method, which is the case for many methods; see [30]. We may then expect the residual R to satisfy

$$(v, R) = 0 \quad (3)$$

for all vectors v in some subspace $V \subset \mathbb{R}^N$. We then obtain the *error estimate*

$$|(\psi, e)| = |(z, R)| = |(z - \pi z, R)| \leq |z - \pi z| |R|, \quad (4)$$

where $\pi z \in V$ is any approximation of the dual solution z in the subspace V . This shows that the error is a product of the factor $S_G = |z - \pi z|$, which measures how well

z may be approximated in V , and the residual R . However, if U is computed using finite numeric precision, typically using double precision arithmetic on a standard computer, we cannot expect that the (3) are satisfied exactly. The best we can hope is therefore that R satisfies

$$|(v, R)| \leq |v| \epsilon_{\text{mach}}$$

for all $v \in V$. Thus, we must modify the error estimate (4) as follows:

$$\begin{aligned} |(\psi, e)| &= |(z, R)| = |(z - \pi z, R) + (\pi z, R)| \leq |z - \pi z| |R| + |(\pi z, R)| \\ &\leq |z - \pi z| |R| + |\pi z| \epsilon_{\text{mach}} = S_G |R| + S_C \epsilon_{\text{mach}}, \end{aligned}$$

where the two stability factors are $S_G = |z - \pi z|$ and $S_C = |\pi z|$, accounting for accumulation of discretisation (Galerkin) and computational errors, respectively.

We now return to the analysis of the system (1), including the definition of the corresponding dual problem, derivation of the error representation and error estimate, and finally a careful analysis of the contribution from round-off errors.

3.2 Error representation

For the system (1), the dual (linearized adjoint) problem takes the form of an initial value problem for a system of linear ordinary differential equations:

$$\begin{aligned} -\dot{z}(t) &= \bar{A}^\top(t) z(t), \quad t \in [0, T), \\ z(T) &= z_T. \end{aligned} \tag{5}$$

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

$$\bar{A}(t) = \int_0^1 \frac{\partial f}{\partial u}(sU(t) + (1-s)u(t), t) \, ds. \tag{6}$$

For a system of ODEs, the choice of initial data z_T for the dual problem determines which component of the global error that should be estimated at final time. Thus with $z_T = (1, 0, 0, \dots, 0)$, one obtains an estimate for the error in the first component of the solution at final time. The data z_T corresponds to the vector ψ in Section 3.1.

Before deriving the error representation, we note the following important property (mean-value theorem) satisfied by the matrix \bar{A} :

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

Based on the formulation of the dual problem we may now derive a (standard) error representation (Theorem 1); see [7–9] and [2]. 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 (5). 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

$$R(t) = \dot{U}(t) - f(U(t), t). \quad (7)$$

Theorem 1 (Error representation) *Let $u : [0, T] \rightarrow \mathbb{R}^N$ be the exact solution of the initial value problem (1), let $z : [0, T] \rightarrow \mathbb{R}^N$ be the solution of the dual problem (5), and let $U : [0, T] \rightarrow \mathbb{R}^N$ be any piecewise smooth approximation of u on a partition $0 = t_0 < t_1 < \dots < t_M = T$ of $[0, T]$, that is, $U|_{(t_{m-1}, t_m)} \in \mathcal{C}^\infty((t_{m-1}, t_m])$ for $m = 1, 2, \dots, M$ (U is left-continuous). Then, the error $U(T) - u(T)$ may be represented by*

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

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

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

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

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

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

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, $\dot{e} - \bar{A}e = \dot{U} - f(U, \cdot) - \dot{u} + f(u, \cdot) = \dot{U} - f(U, \cdot) = R$, which completes the proof. \square

Remark 1 Theorem 1 holds for any piecewise smooth function $U : [0, T] \rightarrow \mathbb{R}^N$, in particular for any piecewise smooth extension of any approximate numerical solution obtained by any numerical method for (1).

3.3 Error estimation

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), $\mathbf{E} = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C$, and derive sharp bounds for each term.

The partition of the error into contributions from data errors, discretization errors and computational errors is guided by a natural decomposition of the error representation into terms that would give a zero net contribution to the total error, if the source of the error were to be removed. Thus, the data error \mathbf{E}_D is precisely zero

if the error in initial data is zero, the discretisation error \mathbf{E}_G is zero in the limit as the time step goes to zero, and the computational error is zero if the discrete equations that define the numerical solution in each time step are solved exactly, which can only be accomplished in the absence of round-off errors. The precise definitions of these contributions are stated in Theorem 2.

To estimate the computational error, we introduce the *discrete residual* \bar{R} defined as follows. For any $p \geq 0$, 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 $0 = \tau_0 < \tau_1 < \dots < \tau_p \leq 1$ of $[0, 1]$, that is, $\text{span}\{\lambda_k\}_{k=0}^p = \mathcal{P}^p([0, 1])$ and $\lambda_i(\tau_j) = \delta_{ij}$. Then, the discrete residual \bar{R}_k is defined on each interval $(t_{m-1}, t_m]$ by

$$\bar{R}_k^m = \lambda_k(0)[U]_{m-1} + \int_{t_{m-1}}^{t_m} \lambda_k((t - t_{m-1})/\Delta t_m) R(t) dt, \quad k = 0, 1, \dots, p. \quad (8)$$

We also define the corresponding interpolation operator π onto the space of piecewise polynomial functions on the partition $0 = t_0 < t_1 < \dots < t_M = T$ 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 2 (Error estimate) *Let $u : [0, T] \rightarrow \mathbb{R}^N$ be the exact solution of the initial value problem (1), let $z : [0, T] \rightarrow \mathbb{R}^N$ be the solution of the dual problem (5), and let $U : [0, T] \rightarrow \mathbb{R}^N$ be any piecewise smooth approximation of u on a partition $0 = t_0 < t_1 < \dots < t_M = T$ of $[0, T]$, that is, $U|_{(t_{m-1}, t_m]} \in \mathcal{C}^\infty((t_{m-1}, t_m])$ for $m = 1, 2, \dots, M$ (U is left-continuous). Then, for any $p \geq 0$ such that the dual solution z is $p + 1$ times differentiable, the following error estimate holds:*

$$\mathbf{E} \equiv \langle z_T, U(T) - u(T) \rangle = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C, \quad (9)$$

where

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

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

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

The precise definitions of the data error \mathbf{E}_D , the discretization error \mathbf{E}_G and the computational error \mathbf{E}_C are:

$$\begin{aligned}\mathbf{E}_D &= \langle z(0), e(0) \rangle, \\ \mathbf{E}_G &= \sum_{m=1}^M \left[\langle z(t_{m-1}) - \pi z(t_{m-1}^+), [U]_{m-1} \rangle + \int_{t_{m-1}}^{t_m} \langle z - \pi z, R \rangle dt \right], \\ \mathbf{E}_C &= \sum_{m=1}^M \left[\langle \pi z(t_{m-1}^+), [U]_{m-1} \rangle + \int_{t_{m-1}}^{t_m} \langle \pi z, R \rangle dt \right].\end{aligned}$$

Proof Starting from the error representation of Theorem 1, we add and subtract the degree p left-continuous piecewise polynomial interpolant πz defined above to obtain

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

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

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

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

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

where $\sum_{m=1}^M \int_{t_{m-1}}^{t_m} \|\pi z\| dt = \int_0^T \|\pi z\| dt \equiv S_C$ and C'_p is a constant depending only on p . This completes the proof. \square

Remark 2 Theorem 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 i th unit vector for $i = 1, 2, \dots, N$.

Remark 3 In the practical application of Theorem 2, we make the approximation $u \approx U$ in the linearisation (6) since the exact solution u is not known. This is common practice in the error analysis literature. As a result, the computed error estimate is valid only in the regime when the computed trajectory U stays close to the exact trajectory u . Interestingly, the result is that—for the examples we have encountered—the estimate of Theorem 2 *overestimates* the size of the error when U is no longer close to u and the linearisation is no longer valid.

Theorem 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, other than that the produced numerical solution U is piecewise differentiable with bounded derivatives. Second, it includes the effect of numerical round-off errors. A similar estimate can be found in [25] but only for the simplest case of the piecewise linear cG(1) method (Crank–Nicolson).

Our analysis effectively treats round-off errors in a similar way to how numerical quadrature errors are treated in the classical a posteriori error analysis [7]. In particular, the analysis is based on the formulation of a single continuous dual (adjoint) problem. In recent work, Estep and colleagues have shown that for iterative methods, the effect of the iteration error can be analyzed using a new dual problem with a different linearisation than the linearisation (6) commonly used for nonlinear problems [4, 10]. This improves the applicability of the a posteriori to account for the different sources contributing to the total error: discretisation error and iteration error. In the current work, the analysis is based on the classic linearisation (6) and a single dual problem to account for the stability and accumulation of all error sources. As will be shown in Section 4, this gives an estimate for the accumulation of round-off errors in very good agreement with numerical experiments.

We now investigate the propagation of numerical round-off errors in more detail. As in Theorem 2, E_C denotes the computational error defined by

$$E_C = \sum_{m=1}^M \left[\langle \pi z(t_{m-1}^+), [U]_{m-1} \rangle + \int_{t_{m-1}}^{t_m} \langle \pi z, R \rangle dt \right]. \quad (10)$$

Theorem 2 bounds the computational error in terms of the discrete residual defined in (8). The discrete residual tests the continuous residual $R = \dot{U} - f$ of (1) against polynomials of degree p . In particular, it tests how well the numerical method satisfies the relation

$$U(t_m) = U(t_{m-1}) + \int_{t_{m-1}}^{t_m} f(U, \cdot) dt. \quad (11)$$

With a machine precision of size ϵ_{mach} , our best hope is that the numerical method satisfies (11) to within a tolerance of size ϵ_{mach} for each component of the vector U . It follows by the Cauchy–Schwarz inequality that

$$\max_{k,m} \|\bar{R}_k^m\| \leq \epsilon_{\text{mach}} \sqrt{N}.$$

We thus have the following corollary.

Corollary 1 *The computational error \mathbf{E}_C of Theorem 2 is bounded by*

$$|\mathbf{E}_C| \leq S_C C'_p \frac{\epsilon_{\text{mach}} \sqrt{N}}{\min_{[0,T]} \Delta t}.$$

This indicates that the computational error scales like Δt^{-1} ; the smaller the time step, the larger the computational error. 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.

3.4 Estimation of round-off errors

The estimate of Corollary 1 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 $+\epsilon_{\text{mach}}$ or $-\epsilon_{\text{mach}}$ with equal probabilities,

$$(\bar{R}_k^m)_i = \begin{cases} +\epsilon_{\text{mach}}, & p = 0.5, \\ -\epsilon_{\text{mach}}, & p = 0.5, \end{cases} \quad (12)$$

for all m, k, i . In reality, round-off errors are not uncorrelated random variables, but the simple model (12) may still give useful results. For a discussion on the applicability of random models to the propagation of round-off errors, see [15] (Section 2.8) and [14].

Under the assumption (12), 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. A similar result is obtained in a series of papers by Li et al. [23, 24]. In [23], it is first noted that there exists an *optimal time step*; that is, a time step for which discretisation errors and round-off errors balance. In [24], it is then found that the round-off error is inversely proportional to the square root of the time step. These results are confirmed by the following theorem.

Theorem 3 *Assume that the round-off error is a random variable of size $\pm\epsilon_{\text{mach}}$ with equal probabilities. Then, the root-mean squared expected computational error \mathbf{E}_C of Theorem 2 is bounded by*

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

where $S_{C_2} = \left(\int_0^T \|\pi z\|^2 dt \right)^{1/2}$ and C'_p is a constant depending only on p .

Proof As in the proof of Theorem 2, we obtain

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

where by assumption $(\tilde{R}_k^m)_i = \epsilon_{\text{mach}} x_{mki}$ and $x_{mki} = \pm 1$ with probability 0.5 and 0.5, respectively. It follows that

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

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

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

where $S_{C_2} = \left(\int_0^T \|\pi z\|^2 dt \right)^{1/2}$. This completes the proof. \square

Remark 4 By Cauchy–Schwarz, the stability factor S_C of Theorem 2 is bounded by $\sqrt{T} S_{C_2}$.

Remark 5 In [14], the effect of numerical round-off error accumulation and its relation to Brownian motion (Brouwer’s law) are discussed in the context of symplectic methods for Hamiltonian systems. It should be noted that although the assumptions of Theorem 3 are similar to those in [14], namely that the process of error accumulation for round-off errors is random rather than systematic, the point under discussion in the present work is different: the effect of time step size rather than the effect of the interval length.

3.5 Application to Galerkin finite element methods

We conclude this section by discussing how the above error estimates apply to the particular methods used in this work. The estimate of Theorem 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 [6, 16–18]).

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

$$\int_{t_{n-1}}^{t_n} v R \, dt = 0 \quad (13)$$

for all $v \in \mathcal{P}^{q-1}([t_{n-1}, t_n])$. It follows that the discrete residual (8) is zero if $p \leq q-1$. However, this is only true in exact arithmetic. In practice, the discrete residual is nonzero and measures how well we solve the $cG(q)$ (13), including round-off errors and errors from numerical quadrature.¹ For the $cG(q)$ method, we further expect the residual to converge as Δt^q . Thus, choosing $p = q-1$ in Theorem 2, one may expect the error for the $cG(q)$ method to scale as

$$E = \mathbf{E}_D + \mathbf{E}_G + \mathbf{E}_C \leq S(T) \left(\epsilon_{\text{mach}} + \Delta t^{2q} + \Delta t^{-1/2} \epsilon_{\text{mach}} \right). \quad (14)$$

Here, $S(T)$ denotes a generic stability factor. As in Theorem 2, each term contributing to the total error is in reality multiplied by a particular stability factor. In practice, however, the growth rates of the different stability factors are similar and related by a constant factor.

4 Numerical results

In this section, we present numerical results in support of Theorem 2 and Theorem 3. The examples are the well-known Lorenz system and Van der Pol oscillator. Both examples illustrate the competing convergence rates for discretisation errors, decreasing rapidly for smaller time steps, and computational errors (round-off error), increasing for smaller time steps.

The numerical results were obtained using the authors' software package Tanganyika [27] which implements the methods described in [25] using high precision

¹To account for additional quadrature errors present if the integral of (13) is approximated by quadrature, one may add and subtract an interpolant πf of the right-hand side f in the proof of Theorem 2 to obtain an additional term $\mathbf{E}_Q = S_Q \max_{[0, T]} \|\pi f - f\|$ where $S_Q = \int_0^T \|z\| \, dt \approx S_C$.

numerics provided by GMP [13]. A complete code for reproducing all results in this paper is available at [22]. Although the software package Tanganyika supports adaptive time-stepping, the focus of the current examples are not on strategies for adaptive time step selection, but rather on verifying the theoretical predictions of the effect of numerical round-off errors and the separation of contributions to the total error stated in Theorem 2. For details on the implementation, see [20].

4.1 The Lorenz system

We first consider the well-known Lorenz system [28], a simple system of three ordinary differential equations exhibiting rapid amplification of numerical errors:

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = rx - y - xz, \\ \dot{z} = xy - bz, \end{cases} \quad (15)$$

where $\sigma = 10$, $b = 8/3$, and $r = 28$. We take $u(0) = (1, 0, 0)$.

The Lorenz system is deterministically chaotic. In the context of a posteriori error analysis of numerical methods for the solution of ODE initial value problems, as in the present work, this means that solutions may, in principle, be computed over arbitrarily long time intervals, but to a rapidly increasing cost as function of the final time T .

4.1.1 Computability and growth of stability factors

In [11], computability was demonstrated and quantified for the Lorenz on time intervals of moderate length ($T = 30$) on a standard desktop computer. This result was further extended to time $T = 48$ in [26], using high order ($\|e(T)\| \sim \Delta t^{30}$) finite element methods. Solutions over longer time intervals have been computed based on shadowing (the existence of a nearby exact solution), see [5], but for unknown initial data. Related work on high-precision numerical methods applied to the Lorenz system include [31] and [19].

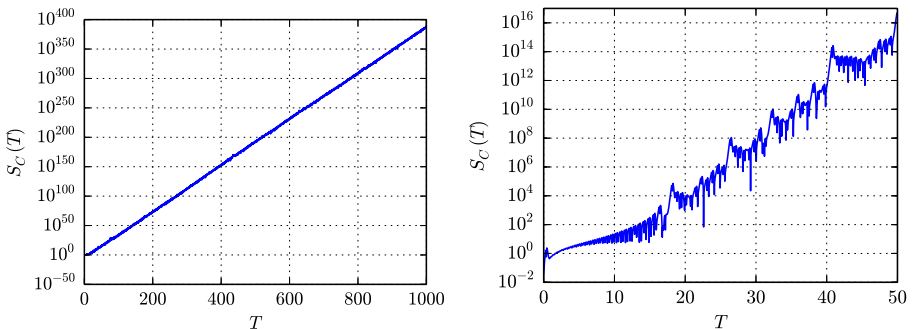


Fig. 2 Growth of the stability factor S_C (left) for the Lorenz system on the time interval $[0, 1000]$ and a detailed plot on the time interval $[0, 50]$ (right)

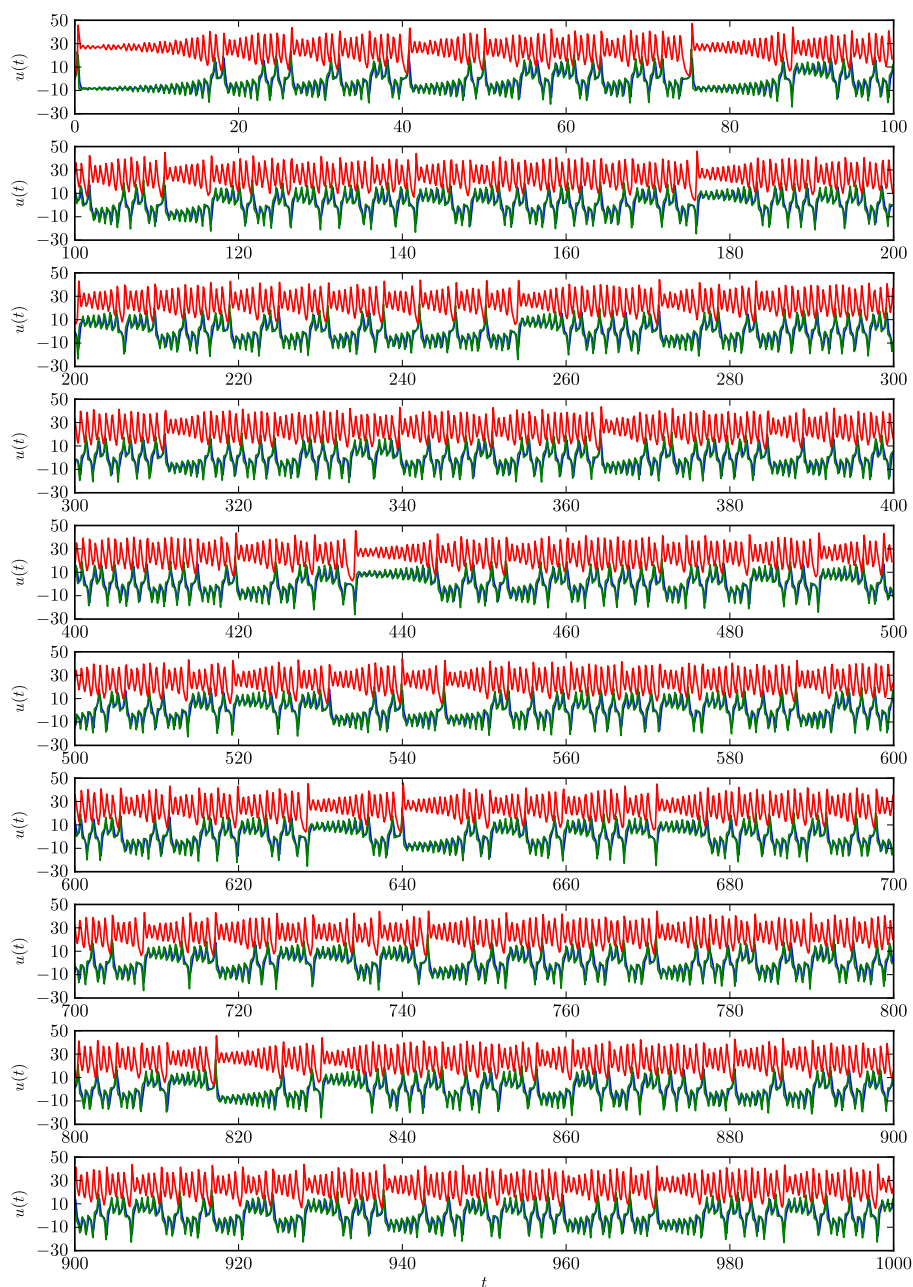


Fig. 3 Accurate reference solution for the three components of the Lorenz system on the interval $[0, 1000]$ with the x and y components plotted in *blue* and *green* respectively (and almost overlaid) and the z component in *red*

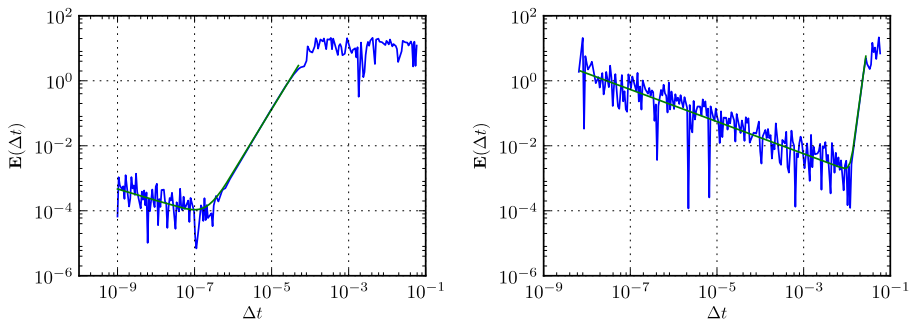


Fig. 4 Error at time $T = 30$ for the cG(1) solution (left) and at time $T = 40$ for the cG(5) solution (right) of the Lorenz system. The slopes of the green lines are $-0.35 \approx -1/2$ and $1.95 \approx 2$ for the cG(1) method. For the cG(5) method, the slopes are $-0.49 \approx -1/2$ and $10.00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

In [21], the authors study the computability of the Lorenz system in detail on the time interval $[0, 1000]$. Computability is here defined as the maximal final time $T = T(\epsilon_{\text{mach}})$ such that a solution may be computed with a given machine precision ϵ_{mach} . The computability may be estimated by examining the growth rate of the stability factors appearing in the error estimate of Theorem 2. By numerical solution of the dual problem, it was found in [21] that the stability factors grow exponentially as $S(T) \sim 10^{0.388T} \sim 10^{0.4T}$; see Fig. 2. By examining in detail the terms contributing

to the error estimate (9), one finds that an optimal step size is given by $\Delta t \sim \epsilon_{\text{mach}}^{\frac{1}{2q + \frac{1}{2}}}$ and that the computability of the Lorenz system is given by

$$T(\epsilon_{\text{mach}}) \sim 2.5n_{\text{mach}}, \quad (16)$$

where $n_{\text{mach}} = -\log_{10} \epsilon_{\text{mach}}$ is the number of significant digits. Based on this estimate, one may conclude that with 16-digit precision, the Lorenz system is computable on $[0, 40]$, while using 400 digits, the Lorenz system is computable on $[0, 1000]$.

We stress that the plot of the stability factor in Fig. 2 gives a good account for the rate of error accumulation *as long as the numerical solution U stays close to the exact solution u* , which is indeed the case for our computed solution. However, once U departs significantly from u , the growth rate indicated by Fig. 2 will grossly overestimate the error accumulation, since the numerical solution will settle into a bounded orbit (of radius $R \approx 50$). The total error itself will remain bounded, whereas the stability factor would indicate an exponential growth. This means that the exponential growth rate $S(T) \sim 10^{0.4T}$ can be used to estimate the limit of computability—the point at which the solution is no longer computable with given resources—but does not give a correct account for the growth rate beyond that point.

In Fig. 3, we plot the solution of the Lorenz system on the interval $[0, 1000]$. 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

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

for the discrete residual of size $\epsilon_{\text{mach}} \approx 2.26 \cdot 10^{-424}$. The very rapid (exponential) accumulation of numerical errors makes the Lorenz “fingerprint” displayed in Fig. 3 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.

4.1.2 Order of convergence and optimal step size

We next investigate how the accumulated error at final time depends on the size of the time step Δt . According to (14), we expect the error to scale like $\Delta t^{2q} + \Delta t^{-\frac{1}{2}} \epsilon_{\text{mach}}$.

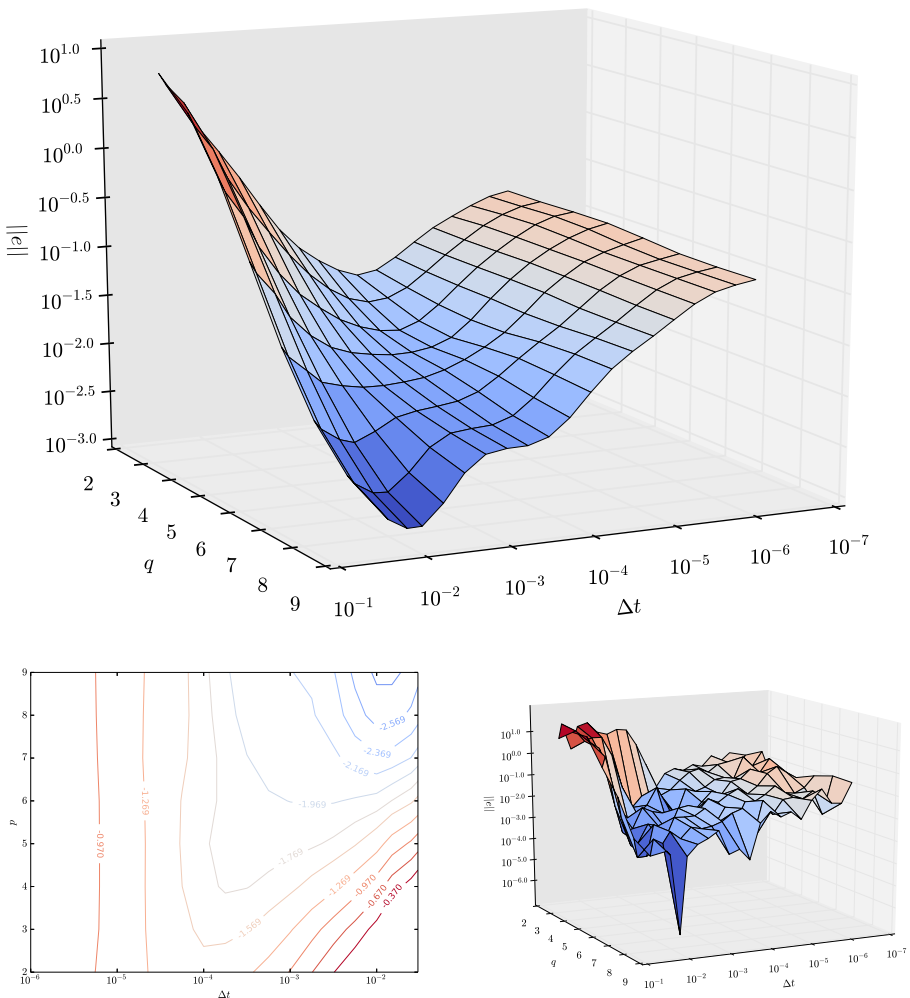


Fig. 5 *Top*: The accumulated total error at final time $T = 40$ for numerical solutions of the Lorenz system with different step size Δt and polynomial degree q using the cG(q) method. Due to the random nature of the round-off errors, the data has been smoothed. *Lower left*: Contour lines of the smoothed data. *Lower right*: The raw data included for completeness

Thus, for a gradually decreasing step size, we expect the error to decrease at a rate of Δt^{2q} . However, as the time step becomes smaller the second term $\Delta t^{-\frac{1}{2}}$ will grow and, for small enough Δt , be the dominating contribution to the error. This picture is confirmed by the results presented in Fig. 4 for two numerical methods, the 2nd order cG(1) and the 10th order cG(5) method. Of particular interest in this figure is the very short range in which the 10th order convergence of the cG(5) method is recovered; with only 16 digits of precision, the dominating contribution to the total error is the accumulated round-off error. We also note that for both methods, one may find an optimal size of the time step Δt for which both contributions to the total error are balanced.

In Fig. 5, results are presented for an investigation of the influence on both the step size Δt and the polynomial degree q in the cG(q) method. As expected, the minimal error is obtained when both the polynomial degree q and the step size are *maximal*. Maximising the step size minimizes the influence of numerical round-off errors (the term $\Delta t^{-\frac{1}{2}}$), and as a consequence the polynomial degree q must be large in order to suppress the discretisation error (the term Δt^{2q}).

4.2 The Van der Pol oscillator

We next consider the Van der Pol oscillator, given by the second order ODE

$$\ddot{u} = \mu(1 - u^2)\dot{u} - u.$$

Rewritten as a system of first order equations, it reads

$$\begin{cases} \dot{u}_1 = u_2, \\ \dot{u}_2 = \mu(1 - u_1^2)u_2 - u_1. \end{cases} \quad (17)$$

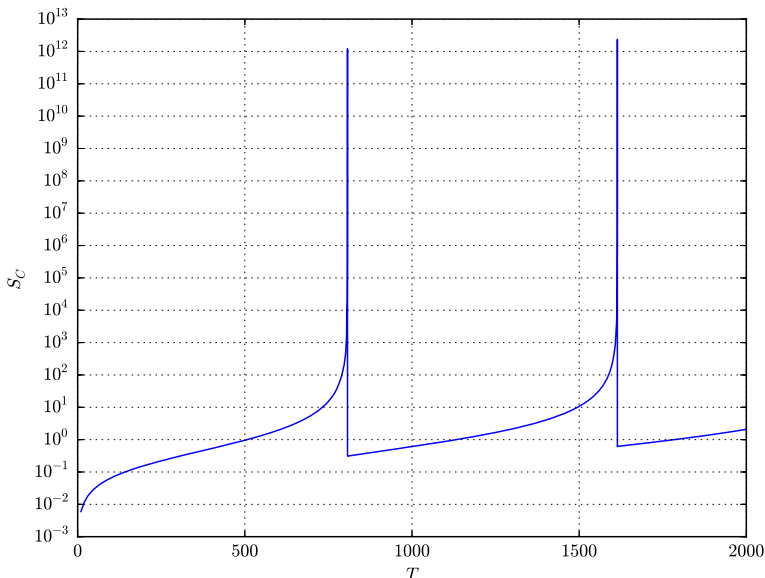


Fig. 6 Growth of the computational stability factor $S_C(T)$ for the Van der Pol oscillator (17)

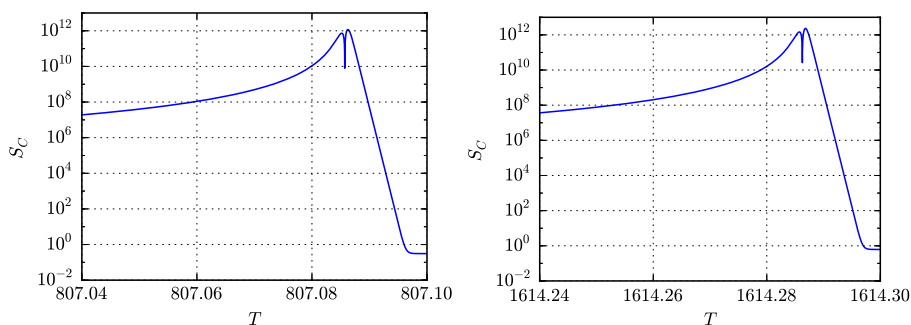


Fig. 7 Detail of growth of the computational stability factor $S_C(T)$ for the Van der Pol oscillator (17)

We compute solutions on $[0, 2\mu]$ for $\mu = 10^3$ and $u(0) = (2, 0)$. This configuration is used as a test problem for ODE solvers in [29]. For large values of the parameter μ , the solution quickly approaches a limit cycle.

As for the Lorenz system, the stability factor(s) grow very rapidly (exponentially), as indicated in Figs. 6 and 7. However, the rapid growth is localized in time close to $T \approx 807 \cdot n$ for $n \in \mathbb{N}$. For times before or after these points of instability, the stability factor is of moderate size. This means that solutions are difficult to compute only at points near the points of instability; that is, a solution may be easily computed at time $t = 1000$ but not at time $t = 807$.

This rapid growth of stability factors is reflected in the growth of the error for numerical solutions as shown in Figs. 8 and 9. Examining these plots in more detail, we notice that the error grows in accordance with the error estimate of Theorem 2

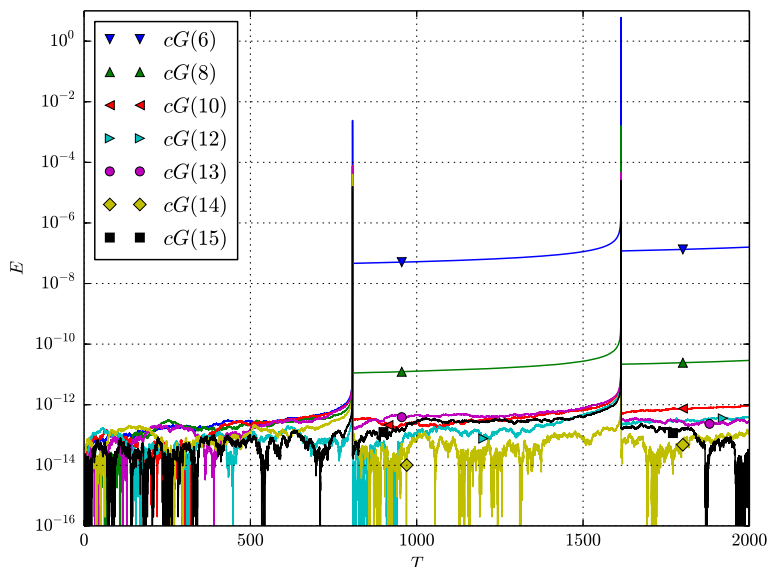


Fig. 8 Growth of error for solutions of the Van der Pol oscillator (17) computed with time step $\Delta t = 10^{-3}$

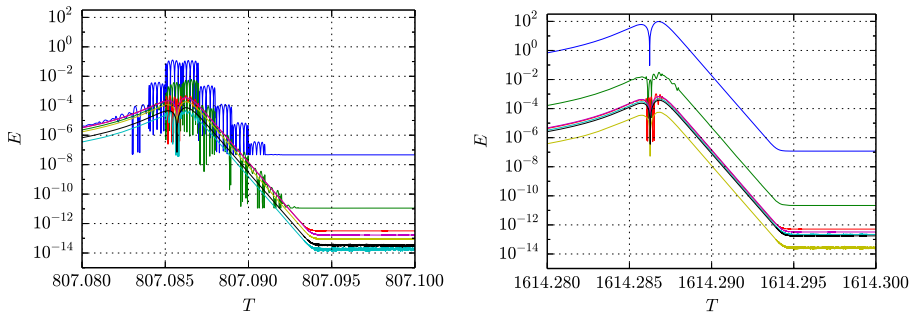


Fig. 9 Detail of growth of error for solutions of the Van der Pol oscillator (17) computed with time step $\Delta t = 10^{-3}$

and (14). For the numerical solutions studied in Figs. 8 and 9, the discretisation error dominates for low order methods. As the polynomial degree q is increased, the error decreases until the point when the computational error starts to dominate. We notice that the baseline error is of size $E \sim 10^{-14}$ for the highest order methods when the stability factor is of size $S \sim 10^2$, and the error spikes at $E \sim 10^{-4}$ at times when the stability factor takes on large values $S \sim 10^{12}$. This is in good agreement with the error estimate: $E \sim S \cdot 10^{-16}$.

5 Conclusions

We have proved error estimates accounting for data, discretization, and computational (round-off) errors in the numerical solution of initial value problems for ordinary differential equations. These error estimates quantify the accumulation rates for numerical round-off error as inversely proportional to the square root of the step size, and proportional to a specific computable stability factor. The effect of round-off errors is mostly pronounced for large values of the stability factor, which includes both chaotic dynamical systems as well as long-time integration of systems which exhibit only a moderate growth of the stability factor.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Akrivis, G., Makridakis, C., Nochetto, R.H.: Galerkin and Runge–Kutta methods: unified formulation, a posteriori error estimates and nodal superconvergence. *Numer. Math.* **118**(3), 429–456 (2011)
2. Becker, R., Rannacher, R.: An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica* **10**, 1–102 (2001)
3. Cao, Y., Petzold, L.: A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM J. Sci. Comput.* **26**(2), 359–374 (2004)

4. Chaudhry, J., Estep, D., Ginting, V., Tavener, S.: A posteriori analysis of an iterative multi-discretization method for reaction–diffusion systems. *Comput. Methods Appl. Mech. Eng.* **267**, 1–22 (2013)
5. Coomes, B.A., Kocak, H., Palmer, K.J.: Rigorous computational shadowing of orbits of ordinary differential equations. *Numerische Mathematik* **69**(4), 401–421 (1995)
6. Delfour, M., Hager, W., Trochu, F.: Discontinuous Galerkin methods for ordinary differential equations. *Math. Comp.* **36**, 455–473 (1981)
7. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to adaptive methods for differential equations. *Acta Numerica* **4**, 105–158 (1995)
8. Estep, D.: A posteriori error bounds and global error control for approximations of ordinary differential equations. *SIAM J. Numer. Anal.* **32**, 1–48 (1995)
9. Estep, D., French, D.: Global error control for the continuous Galerkin finite element method for ordinary differential equations. *m2AN* **28**, 815–852 (1994)
10. Estep, D., Ginting, V., Tavener, S.: A posteriori analysis of a multirate numerical method for ordinary differential equations. *Comput. Methods Appl. Mech. Eng.* **223**, 10–27 (2012)
11. Estep, D., Johnson, C.: The pointwise computability of the Lorenz system. *Math. Models. Meth. Appl. Sci.* **8**, 1277–1305 (1998)
12. Estep, D.J.: A Short Course on Duality, Adjoint Operators, Green’s Functions, and A Posteriori Error Analysis, Department of Mathematics Colorado State University (2004)
13. Granlund, T.: the GMP development team: GNU MP: The GNU Multiple Precision Arithmetic Library. <http://gmplib.org/> (2015)
14. Hairer, E., McLachlan, R.I., Razakrivony, A.: Achieving brouwer’s law with implicit runge–kutta methods. *BIT Numer. Math.* **48**(2), 231–243 (2008)
15. Higham, N. Accuracy and stability of numerical algorithms, 2nd edn. Society for Industrial Mathematics (2002)
16. Hulme, B.L.: Discrete Galerkin and related one-step methods for ordinary differential equations. *Math. Comput.* **26**(120), 881–891 (1972)
17. Hulme, B.L.: One-step piecewise polynomial Galerkin methods for initial value problems. *Math. Comput.* **26**(118), 415–426 (1972)
18. Johnson, C.: Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations. *SIAM J. Numer. Anal.* **25**(4), 908–926 (1988)
19. Jorba, A., Zou, M.: A software package for the numerical integration of ODEs by means of high-order Taylor methods. *Exp. Math.* **14**(1), 99–117 (2005)
20. Kehlet, B.: 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)
21. Kehlet, B., Logg, A.: Quantifying the Computability of the Lorenz System. In: *Adaptive Modeling and Simulation* (2013)
22. Kehlet, B., Logg, A.: Code package for the paper A posteriori error analysis of round-off errors in the numerical solution of ordinary differential equations (2015). doi:[10.5281/zenodo.16671](https://doi.org/10.5281/zenodo.16671)
23. Li, J., Zeng, Q., Chou, J.: Computational uncertainty principle in nonlinear ordinary differential equations i: Numerical results. *Sci. China (E)* **43**(5), 449–460 (2000)
24. Li, J., Zeng, Q., Chou, J.: Computational uncertainty principle in nonlinear ordinary differential equations ii: Theoretical analysis. *Sci. China (E)* **44**(1), 55–74 (2001)
25. Logg, A.: Multi-Adaptive Galerkin methods for ODEs I. *SIAM J. Sci. Comput.* **24**(6), 1879–1902 (2003)
26. Logg, A.: Multi-Adaptive Galerkin methods for ODEs II: Implementation and applications. *SIAM J. Sci. Comput.* **25**(4), 1119–1141 (2003)
27. Logg, A., Kehlet, B.: Tanganyika. <https://bitbucket.org/benjamik/tanganyika>
28. Lorenz, E.N.: Deterministic nonperiodic flow, vol. 20 (1963)
29. Mazzia, F., Magherini, C.: Test set for initial value problem solvers, release 2.4. Technical Report 4, Department of Mathematics, University of Bari, Italy. Available at <http://pitagora.dm.uniba.it/~testset> (2008)
30. Saad, Y., Schultz, M.H.: Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Math. Comput.* **44**(170), 417–424 (1985)
31. Viswanath, D.: The fractal property of the Lorenz attractor. *Physica D: Nonlinear Phenomena* **190**(1–2), 115–128 (2004)