

# Convergence Rates for Adaptive Approximation of Ordinary Differential Equations Based on Global and Local Errors

Kyoung-Sook Moon<sup>†</sup>   Anders Szepessy<sup>†‡</sup>   Raúl Tempone<sup>†</sup>   Georgios E. Zouraris<sup>§</sup>

October 17, 2001

## Abstract

A variational principle is used to derive an error representation of the form “Global error =  $\sum$  local error  $\cdot$  weight” for general approximation of functions of solutions to ordinary differential equations. The analysis includes approximation of this representation with computable leading order term, based on approximation of the local errors and weights including roundoff errors. The error representation is shown, in a sense, to be the only useful global error indicator for adaptive mesh refinements. An adaptive algorithm including dividing and merging of time steps, based on the error expansion, is proven to stop with the optimal number,  $N$ , of steps up to a problem independent factor defined in the algorithm. A version of the algorithm with decreasing tolerance stops with also the total number of steps, including all refinement levels, bounded by  $\mathcal{O}(N)$ ; an alternative version with constant tolerance stops with  $\mathcal{O}(N \log N)$  total steps. The global error is bounded by the tolerance parameter, asymptotically as the tolerance tends to zero. For a  $p$ -th order accurate method the optimal number of adaptive steps is proportional to the  $p$ -th root of the  $L^{\frac{1}{p+1}}$  quasi-norm of the error density, while the number of uniform steps, with the same error, is proportional to the  $p$ -th root of the larger  $L^1$ -norm of the error density.

**Key words.** adaptive method, a posteriori error estimate, local error, computational complexity, mesh refinement, convergence rates, ordinary differential equations, step size control, variational principle, roundoff error.

**AMS subject classification.** 65L50, 65L70, 65Y20

---

<sup>†</sup>Institutionen för Numerisk Analys och Datalogi, Kungl. Tekniska Högskolan, S-100 44 Stockholm.

<sup>‡</sup>Corresponding author, szepessy@math.kth.se, Matematiska Institutionen, Kungl. Tekniska Högskolan, S-100 44 Stockholm, phone +46-8-790 7494, fax +46-8-790 0930.

<sup>§</sup>Universite Paris IX Dauphine, Place du Marechal de Lattre de Tassigny, F-75775 Paris CEDEX 16, France  
email: moon@nada.kth.se, szepessy@math.kth.se, rtempone@nada.kth.se, zouraris@nada.kth.se  
This work is supported by the EU-TMR project HCL # ERBFMRXCT960033, the Swedish Council for Engineering Science grant # 222-148, UdelaR and UdeM in Uruguay, the Swedish Network for Applied Mathematics, the Parallel and Scientific Computing Institute (PSCI) and the Swedish National Board for Industrial and Technical Development (NUTEK).

# 1 INTRODUCTION TO ADAPTIVE ODE METHODS

This paper studies a variational principle to estimate the global discretization error for ordinary differential equations of the form

$$\text{Global error} = \sum_{\text{time steps}} \text{local error} \cdot \text{weight} + \text{higher order error terms.} \quad (1.1)$$

Such estimates are fundamental for a priori and a posteriori analysis of numerical methods, and they are especially useful for adaptive mesh refinement algorithms.

Error estimates for differential equations, based on the local errors, can be derived by the classical error equation and linearization, cf. [21], [10], by Galerkin orthogonality using either local problems or the residual, cf. [2], [15] and by a variational principle, following Alekseev [1] and Gröbner [19] who introduced a variational principle to derive an error representation for perturbation errors in differential equations based on the residual of the perturbation.

The variational principle here uses local errors and has the advantage that the global error is a weighted sum of the local errors

$$\text{Global error} = \sum_{\text{time steps}} \text{local error} \cdot \text{weight.} \quad (1.2)$$

However in (1.2) both the true local errors and the weights are non computable, therefore the next idea is to transfer the expansion (1.2) to the form (1.1) with computable approximations of the local errors and the weights. The error estimate, based on the variational formulation, only requires the nodal values of the local errors. It is therefore applicable to all discretization methods. Note that the classical derivation of (1.1) based on the error equation does not include (1.2), since the global error is polluted by the linearization error between  $X$  and  $\bar{X}$ .

Consider a solution  $X : [0, T] \rightarrow \mathbb{R}^d$  of a differential equation, with flux  $a : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,

$$\begin{aligned} \frac{dX(t)}{dt} &= a(t, X(t)), \quad 0 < t \leq T, \\ X(0) &= X_0, \end{aligned} \quad (1.3)$$

and an approximation  $\bar{X}$  of (1.3) by any numerical method, satisfying the same initial condition

$$\bar{X}(0) = X(0) = X_0 \quad (1.4)$$

with time steps

$$0 = t_0 < \dots < t_N = T.$$

The following sections derive estimates of the form (1.1) for the global error

$$g(X(T)) - g(\bar{X}(T)) \quad (1.5)$$

with a given general function  $g : \mathbb{R}^d \rightarrow \mathbb{R}$ . The estimates will use the local error  $e$  defined by

$$e(t_n) \equiv \tilde{X}(t_n) - \bar{X}(t_n), \quad (1.6)$$

where the local exact solution  $\tilde{X}$  satisfies for each time step  $(t_{n-1}, t_n]$

$$\begin{aligned} \frac{d\tilde{X}(t)}{dt} &= a(t, \tilde{X}(t)), \quad t_{n-1} < t \leq t_n, \\ \tilde{X}(t_{n-1}) &= \bar{X}(t_{n-1}). \end{aligned} \quad (1.7)$$

Theorems 2.1 and (2.3, 2.4) prove error estimates of the form (1.2) and (1.1), respectively, with an approximation of the weight function, which solves a certain linear backward dual problem, obtained by linearizing the forward problem (1.3) around the approximate solution. It is possible to approximate the weight with lower accuracy and a coarser mesh than for the approximate solution  $\bar{X}$ . When the solution is well resolved, the work to determine the weight can therefore be smaller than the work to solve the differential equation (1.3). However, to solve the dual problem requires to store the approximate solution on the coarser mesh. On the other hand, some computer programs for numerical solution of ordinary differential equations store the solution at all time levels for other reasons, e.g. for post processing. The use of dual functions is standard in optimal control theory and in particular for adaptive mesh control for ordinary and partial differential equations, see [3], [6], [7], [16], [23], [24], [35].

A main application of the error estimate (1.1, 1.2) is in adaptive algorithms, where the aim is to choose the optimal mesh, cf. [35]. For a given bound on the global error, the number of time steps are minimized by choosing for all time steps

$$\text{local error} \cdot \text{weight} = \text{constant}. \tag{1.8}$$

Therefore the weights need to be determined to find the optimal mesh.

Despite the wide use of adaptive algorithms for differential equations and the well developed theory of a posteriori error estimates, surprisingly little is known theoretically on the behavior of adaptive mesh refinement algorithms. For constant step size  $\Delta t$ , approximations with error  $\mathcal{O}(\Delta t^p)$  require computational work with  $\mathcal{O}(1/\Delta t)$  operations. Analogously, for adaptive methods, it is natural to study the approximation error and the work as the tolerance parameter tends to zero. For a  $p$ -th order accurate method, the number of uniform steps to approximate with a given error turns out to be proportional to the  $p$ -th root of the  $L^1$ -norm of the error density, which is  $(\text{local error} \cdot \text{weight})/\Delta t^{p+1}$ , while the optimal number of adaptive steps is proportional to the  $p$ -th root of the  $L^{\frac{1}{p+1}}$  quasi-norm of the error density. Theorems 3.2, 3.3 and 3.4 in Section 3 prove that an adaptive algorithm stops with the optimal number of steps,  $N$ , up to a problem independent factor and, asymptotically as the tolerance parameter tends to zero, the global error is bounded by the tolerance times a problem independent factor, defined in the algorithm. The total number of time steps, including all refinement levels, can be bounded by the number of steps on the finest level times a problem independent factor, provided the tolerance in each refinement level is decreased by a constant factor to guarantee that the number of steps increase, see Theorem 3.6. With varying tolerance, the final stopping tolerance is of course a priori not precisely known. With constant tolerance, the total number of steps including all levels becomes  $\mathcal{O}(N \log N)$ .

The authors are not aware of results on convergence rates and asymptotic work related to Theorem 3.2, 3.3, 3.4 and 3.6 for other algorithms to solve ordinary differential equations. One reason for this is that most adaptive algorithms are based on making a combination of the absolute and the relative local errors approximately constant, ignoring the weights in (1.8), cf. [20], [33]. Although these algorithms in practise perform very well, a proof of the optimality of the mesh is lost and since the tolerance parameter measures the local error without the weight, there is by (1.1) in general no explicit relation of this tolerance parameter and the global error. Many such algorithms also lack proofs of convergence of the approximations. One exception is the work [25], which proves the convergence of ODE23 of MATLAB version 4.2 solving ordinary differential equations. Adaptivity based on the local errors, without the weights, has the clear advantage to avoid the additional storage and work to compute the weight at many time levels.

In the special case of integration, adaption is better understood and [18] shows that local error indicators give rigorous error bounds in an average probabilistic sense. Adaptive mesh refinement algorithms based on finite element approximations for linear coercive elliptic boundary value problems are shown to converge in energy norm, see [4] and [13]. The related adaptive wavelet algorithms, for linear coercive elliptic boundary value problems, are better understood by the recent breakthrough [8], where an adaptive wavelet method is constructed and analyzed with optimal convergence rate  $\mathcal{O}(N^{-s})$  in energy norm for an  $N$ -term wavelet approximation requiring close to optimal work  $\mathcal{O}(N \log N)$ . The work [32] introduces adaptive time stepping for weak approximation of stochastic differential equations in the spirit of Section 2 and 3.

The literature on information based complexity, cf. [31], [34], [36],[5], discuss the efficiency of adaptive versus non-adaptive methods. A central result by Bakhvalov and Smolyak proves that, using a fixed number of functional evaluations, there is for each adaptive method a non-adaptive method which has as small maximal error as the adaptive method for approximation of linear functionals,  $S : \mathcal{F} \rightarrow \mathbb{R}$ , such as e.g.  $Sf = \int_0^1 f(t)dt$ , with functions  $f$  in a convex symmetric subset  $\mathcal{F}$  of a normed linear function space. A symmetric set is a set which contains  $-f$ , if  $f$  is in the set. A precise statement of the theorem is in Remark 3.7. Starting from Bakhvalov and Smolyak's result there is a discussion when adaptive methods for integration and solution of ordinary and partial differential equations are useful, cf. the insightful review [31]. The study here differs from Bakhvalov and Smolyak's work in two essential assumptions: Section 3 and 4 prove that an adaptive algorithm applied to a fixed differential equation (and a fixed discretization method), (1.3), uses asymptotically close to the optimal number of time steps to approximate with a given error tolerance, (1.5), as the number of steps tends to infinity, while Bakhvalov and Smolyak's work analyzes discretization methods based on the maximal error in a convex function set, with a fixed number of steps. The performance of the algorithm in Section 3 and 4 is not characterized by convex function sets, as in [5]; on the contrary, applied to integration, i.e.  $a(t, x) = a(t)$  in (1.3), the estimate of the number of steps to approximate with error TOL in Section 3, using a  $p$ -th order accurate method, shows that adaptive integration is much more efficient than uniform steps, asymptotically as  $\text{TOL} \rightarrow 0$ , if

$$\|a^{(p)}\|_{L^{\frac{1}{p+1}}(0,T)} \ll \|a^{(p)}\|_{L^1(0,T)},$$

where  $a^{(p)} \equiv d^p a/dt^p$  is the error density (non adaptive methods with non uniform steps would require some additional a priori information to improve over uniform steps). In particular, the functions which can be adaptively integrated, with given asymptotic behavior of the error and number of steps, are characterized by the non convex set

$$\left\{ a \in C^p([0, T]) : \|a^{(p)}\|_{L^{\frac{1}{p+1}}} \leq c \right\} \quad (1.9)$$

for a constant  $c$ . In conclusion, the goal here is to solve a problem to a certain accuracy with minimal asymptotic work by using appropriate adaptive time steps. We do not address the related problem to adaptively determine the order of the method and to determine implicit/explicit alternatives. The closely related problem of efficient adaptive and non adaptive approximation of functions, measured in  $L^q$  norms, has been characterized by DeVore [11] using Besov spaces, see Remark 3.8.

The outline of the paper is as follows. Section 2 proves, by a variational principle, an error representation (1.2) and an error expansion (1.1). Section 3 describes and analyzes an adaptive algorithm. Finally, Section 4 presents numerical experiments based on the adaptive algorithm.

## 2 THE VARIATIONAL PRINCIPLE

Let  $X(s; t, y)$  denote the solution of (1.3) at time  $s$ , which at time  $t$  takes the value  $y$ , i.e.

$$\begin{aligned} \frac{dX}{ds}(s; t, y) &= a(s, X(s; t, y)), \quad t < s \leq T, \\ X(t; t, y) &= y. \end{aligned} \tag{2.1}$$

Define the function  $u : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$  by

$$u(t, y) \equiv g(X(T; t, y)), \quad t < T, \tag{2.2}$$

provided the differential equation (2.1) on  $(t, T)$  has a unique solution for all initial data  $y$  in  $\mathbb{R}^d$  and all  $t \in (0, T)$ . In the following theorem, the global approximation error for differential equations is represented in terms of the local errors and their weights, depending on the first variation of  $u$ . The generalization to partial differential equations is then possible with some convenient assumptions, see [29], [30].

**Theorem 2.1** *Assume that (2.1) has a unique continuous solution  $X$  for all initial data  $y \in \mathbb{R}^d$  and that the flux  $a(t, x)$  is differentiable in  $x$ , for all  $t \in (0, T)$ . Let  $e(t_n) \equiv \tilde{X}(t_n) - \bar{X}(t_n)$  denote the local error of an approximation,  $\bar{X}$ , of (1.3), satisfying (1.4). Then, for any differentiable function  $g : \mathbb{R}^d \rightarrow \mathbb{R}$ , the function  $u$  is defined by (2.2) and the global error is a weighted sum of the local error with the representation*

$$g(X(T)) - g(\bar{X}(T)) = \sum_{n=1}^N \left( e(t_n), \int_0^1 \Psi \left( t_n, \bar{X}(t_n) + se(t_n) \right) ds \right) \tag{2.3}$$

where  $(\cdot, \cdot)$  is the standard scalar product on  $\mathbb{R}^d$  and  $\Psi(t, y) \equiv \Psi_X(t) \in \mathbb{R}^d$  is the first variation of  $u$  in the sense that for all  $w \in \mathbb{R}^d$  and all sufficiently small  $\delta > 0$

$$u(t, y + \delta w) - u(t, y) = (\Psi_X(t), \delta w) + o(\delta).$$

The weight function  $\Psi_X$  satisfies for  $t < s < T$  the dual equation

$$\begin{aligned} -\frac{d\Psi_X(s)}{ds} &= (a')^*(s, X(s; t, y))\Psi_X(s), \\ \Psi_X(T) &= g'(X(T; t, y)), \end{aligned} \tag{2.4}$$

where  $(a')^*(s, x)$  is the transpose of the Jacobian matrix  $a'(s, x) \equiv \left\{ \frac{\partial a_i}{\partial x_j}(s, x) \right\} \in \mathbb{R}^{d \times d}$ , and  $X$  solves (2.1).

**Proof.** By the construction (2.2), the function  $u$  is constant along the characteristics  $\tilde{X}$ , i.e. for all  $t$  and  $\tau$

$$u(t, \tilde{X}(t)) = u(\tau, \tilde{X}(\tau)). \tag{2.5}$$

Therefore the initial condition for the local problem (1.7) shows that

$$u(t_n, \tilde{X}(t_n)) = u(t_{n-1}, \tilde{X}(t_{n-1})) = u(t_{n-1}, \bar{X}(t_{n-1})), \quad n = 1, \dots, N,$$

and consequently the initial condition (1.4) and (2.2) imply that

$$\begin{aligned}
\sum_{n=1}^N (u(t_n, \tilde{X}(t_n)) - u(t_n, \bar{X}(t_n))) &= u(0, \bar{X}(0)) - u(T, \bar{X}(T)) \\
&= u(0, X(0)) - u(T, \bar{X}(T)) \\
&= g(X(T)) - g(\bar{X}(T)).
\end{aligned} \tag{2.6}$$

The function  $U : [0, 1] \rightarrow \mathbb{R}$ , defined by

$$U(s) = u(t_n, s\tilde{X}(t_n) + (1-s)\bar{X}(t_n)),$$

and the equality

$$U(1) - U(0) = \int_0^1 U'(s) ds$$

show that each term in the sum (2.6) can be written

$$u(t_n, \tilde{X}(t_n)) - u(t_n, \bar{X}(t_n)) = \left( e(t_n), \int_0^1 \Psi(t_n, \bar{X}(t_n) + se(t_n)) ds \right),$$

which proves (2.3). The first variation  $\partial X(s; t, y)/\partial y$  exists, since  $a(s, x)$  is differentiable in  $x$ . The combination of the existence of the first variation of  $X$  and the assumption that  $g$  is differentiable, imply by (2.1-2.2) that  $\Psi$  exists. Finally, to verify that  $\Psi$  satisfies the dual equation (2.4), observe for any  $w \in \mathbb{R}^d$  and  $\delta \rightarrow 0$  that two solutions  $X^1$  and  $X^2$  of (2.1), with initial data  $X^1(t) = y \in \mathbb{R}^d$  and  $X^2(t) = y + \delta w$  satisfy

$$\begin{aligned}
0 &= \frac{d}{dt} (u(t, X^2(t)) - u(t, X^1(t))) \\
&= \frac{d}{dt} (\Psi_{X^1}, X^2(t) - X^1(t)) + o(\delta) \\
&= \left( \frac{d}{dt} \Psi_{X^1}, X^2(t) - X^1(t) \right) + \left( \Psi_{X^1}, \frac{d}{dt} X^2(t) - \frac{d}{dt} X^1(t) \right) + o(\delta) \\
&= \left( \delta \frac{d}{dt} \Psi_{X^1}, w \right) + (\delta \Psi_{X^1}, a'(t, X^1(t))w) + o(\delta),
\end{aligned}$$

which proves (2.4) in the limit  $\delta \rightarrow 0$ . □

Our next goal is to construct adaptive methods based on Theorem 2.1. The starting point for the adaptive method is an expansion of the representation (2.3) with leading order term in computable form. The derivation of this expansion is based on an approximation  $\bar{\Psi}$  of the weight  $\Psi$  and an approximation  $\bar{e}$  of the local error  $e$

$$\begin{aligned}
\sum_{n=1}^N (e(t_n), \Psi) &= \sum_{n=1}^N (\bar{e}(t_n), \bar{\Psi}) \\
&= \sum_{n=1}^N (e(t_n) - \bar{e}(t_n), \bar{\Psi}) + \sum_{n=1}^N (e(t_n), \Psi - \bar{\Psi}).
\end{aligned} \tag{2.7}$$

*Approximation of the weight.* The averaged weight function  $\Psi$  in Theorem 2.1, which is needed to determine the optimal step size in an adaptive method, can be computed by approximating (2.1) and (2.4). Therefore, any  $p$ -th order accurate approximation  $(\bar{X}, \bar{\Psi})$  of  $(X, \Psi)$ , which solves the systems of differential equations (1.3) and (2.4), satisfies

$$|\bar{\Psi}(t_n) - \Psi(t_n, \bar{X}(t_n))| = \mathcal{O}((\max \Delta t)^p) + \mathcal{O}\left(\frac{\epsilon}{\min \Delta t}\right) \quad (2.8)$$

where  $\epsilon$  is the machine roundoff unit and  $\Delta t_n = t_n - t_{n-1}$  with  $\max \Delta t \equiv \max_n \Delta t_n$  and  $\min \Delta t \equiv \min_n \Delta t_n$ . The effect of roundoff error is neglected in the formulation of Theorems 2.2 and 2.3 below. Instead a remark in the end of the section includes roundoff error in the error estimation.

A natural choice of approximation  $\bar{\Psi}$ , for a  $p$ -th order one step method  $\bar{X}$  written in the form

$$\bar{X}(t_n) = A(\bar{X}(t_{n-1}), \Delta t_n), \quad (2.9)$$

is

$$\begin{aligned} \bar{\Psi}_i(t_{n-1}) &= \sum_{j=1}^d \partial_{x_i} A_j(\bar{X}(t_{n-1}), \Delta t) \bar{\Psi}_j(t_n), \\ \bar{\Psi}_i(T) &= \partial_{x_i} g(\bar{X}(T)), \end{aligned} \quad (2.10)$$

which yields a  $p$ -th order accurate approximation  $(\bar{X}, \bar{\Psi})$  of  $(X, \Psi)$  and satisfies

$$\bar{\Psi}_i(t_{n-1}) = \partial_{x_i} g(\bar{X}(T; \bar{X}(t_{n-1}) = x)). \quad (2.11)$$

The relation (2.11) is the discrete version of the fact that  $\Psi(t)$  is the first variation of  $g(X(T))$  with respect to variation in the location of the path  $X(t)$  at time  $t$ , and (2.11) holds precisely when  $\bar{\Psi}$  is defined by (2.10). The Jacobian matrix  $\partial_{x_i} A_j(\bar{X}(t_{n-1})) = \frac{\partial \bar{X}_j(t_n)}{\partial \bar{X}_i(t_{n-1})}$  can be approximated by numerical differentiation of  $\bar{X}(t_n)$  with respect to  $\bar{X}(t_{n-1})$ , or alternatively the Jacobian can be evaluated explicitly for each method, e.g. to preserve a sparse structure. To conclude, we have the error estimate

**Theorem 2.2** *Suppose that (2.8) and the assumptions of Theorem 2.1 hold. Let  $\partial_{xx}u(t, x)$  in (2.2) be uniformly bounded for  $(t, x) \in [0, T] \times \mathbb{R}^d$ . Then the global approximation error for the differential equation (1.3) satisfies the estimate*

$$\begin{aligned} &g(X(T)) - g(\bar{X}(T)) \\ &= \sum_{n=1}^N (e(t_n), \bar{\Psi}(t_n) + \mathcal{O}(|e(t_n)|)) + \mathcal{O}((\max \Delta t)^p) \end{aligned} \quad (2.12)$$

where  $e(t_n) \equiv \tilde{X}(t_n) - \bar{X}(t_n)$  is the local error and  $(\bar{X}, \bar{\Psi})$  is a  $p$ -th order accurate approximation of the system (1.3) and (2.4).

**Proof.** A combination of Theorem 2.1, (2.8) and the boundedness of  $\partial_{xx}u \equiv \partial_x \Psi$  implies that

$$\begin{aligned} &\Psi(t_n, \bar{X}(t_n) + se(t_n)) - \bar{\Psi}(t_n) \\ &= (\Psi(t_n, \bar{X}(t_n) + se(t_n)) - \Psi(t_n, \bar{X})) + (\Psi(t_n, \bar{X}) - \bar{\Psi}(t_n)) \\ &= \mathcal{O}(e(t_n)) + \mathcal{O}((\max \Delta t)^p), \end{aligned}$$

which proves (2.12).  $\square$

*Approximation of the local error.* The next step in order to derive an error estimate based on computable quantities is to approximate the local error  $e = \tilde{X} - \bar{X}$  by replacing the exact local solution  $\tilde{X}$  by an approximation  $\bar{\bar{X}}$  of higher accuracy than  $\bar{X}$ , i.e. with smaller time steps or a higher order method in a higher precision. For smooth solutions  $X$ , the existence of the limits

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} (\Delta t)^{-(p+1)} (\tilde{X}(t_n) - \bar{X}(t_n)), \\ \lim_{\Delta t \rightarrow 0} (\Delta t)^{-(q+1)} (\tilde{X}(t_n) - \bar{\bar{X}}(t_n)), \end{aligned} \quad (2.13)$$

determines by Richardson extrapolation a constant  $\gamma$ , for  $q \geq p$  cf. [9], such that

$$e(t_n) = \tilde{X}(t_n) - \bar{X}(t_n) = \gamma \left( \bar{\bar{X}}(t_n) - \bar{X}(t_n) \right) + o(\Delta t^{p+1}). \quad (2.14)$$

For instance there holds:  $\gamma = 2^p / (2^p - 1)$  for  $\bar{\bar{X}}$  computed with the half mesh size and  $q = p$ ; and  $\gamma = 1$  for  $\bar{\bar{X}}$  computed with a higher order method  $q > p$ , see [20]. Let  $\Delta t(t) \equiv \Delta t_n$ ,  $t_{n-1} < t \leq t_n$ . The replacement of the exact local error with this approximate local error leads to

**Theorem 2.3** *Suppose that the limits (2.13) exist and that the assumptions of Theorem 2.1 and 2.2 hold. Then the global approximation error for the differential equation (1.3) satisfies the estimate*

$$g(X(T)) - g(\bar{X}(T)) = \sum_{n=1}^N (\bar{e}(t_n), \bar{\Psi}(t_n)) + \int_0^T o(\Delta t^p(t)) dt \quad (2.15)$$

where  $\bar{e}(t_n) \equiv \gamma \left( \bar{\bar{X}}(t_n) - \bar{X}(t_n) \right)$  is the approximation of local error in (2.14) and  $(\bar{X}, \bar{\Psi})$  is a  $p$ -th order accurate approximation of the system (1.3) and (2.4).

**Proof.** By substituting (2.14) in (2.12), we obtain

$$g(X(T)) - g(\bar{X}(T)) = \sum_{n=1}^N (\bar{e}(t_n), \bar{\Psi}(t_n)) + \mathcal{E}$$

where

$$\mathcal{E} \equiv \sum_{n=1}^N \left[ o(\Delta t^{p+1}) + \mathcal{O}(\Delta t^{p+1}) \cdot (\Delta t^{p+1} + (\max \Delta t)^p) \right] = \int_0^T o(\Delta t^p) dt \leq o((\max \Delta t)^p),$$

which proves theorem.  $\square$

Assume that  $\bar{X}^h$  and  $\bar{X}^H$  are approximations based on (2.9), where the step sizes  $\Delta_h t(t)$  and  $\Delta_H t(t)$  satisfy

$$\begin{aligned} \frac{\Delta_h t(t)}{\Delta_H t(t)} \text{ is independent of } t, \\ \max_t \Delta_h t(t) = h, \\ \max_t \Delta_H t(t) = H. \end{aligned} \quad (2.16)$$

Then the convergence assumption

$$\begin{aligned} g(X(T)) - g(\overline{X}^h(T)) &= ch^p + \alpha_h, \\ \alpha_h &= o(h^p), \end{aligned} \quad (2.17)$$

is meaningful for two positive constants  $c$  and  $p$ . An attractive alternative to the approximation (2.15) is to use (2.17) and apply Richardson extrapolation directly to the  $p$ -th order accurate approximations  $\overline{X}^h$  and  $\overline{X}^H$  of  $X$  to obtain

**Theorem 2.4** *Suppose that (2.16), (2.17) and the assumptions of Theorem 2.1 hold. Let the two  $p$ -th order accurate approximations  $\overline{X}^h$  and  $\overline{X}^H$  of  $X$ , be defined by*

$$\overline{X}^h(t_n) = A^h(\overline{X}^h(t_{n-1})), \quad \overline{X}^H(t_n) = A^H(\overline{X}^H(t_{n-1})) \quad (2.18)$$

following (2.9). Then the global approximation error for the differential equation (1.3) satisfies the estimate

$$\begin{aligned} g(X(T)) - g(\overline{X}(T)) &= \frac{1}{\left(\frac{H}{h}\right)^p - 1} \sum_{n=1}^N \left( A^h(\overline{X}^H(t_{n-1})) - A^H(\overline{X}^H(t_{n-1})), \bar{\Phi}(t_n) \right) + \alpha, \end{aligned} \quad (2.19)$$

$$\alpha \equiv \alpha_h + \frac{\alpha_h - \alpha_H}{\left(\frac{H}{h}\right)^p - 1} = o(h^p), \quad (2.20)$$

where the weight function  $\bar{\Phi}(t_n) \in \mathbb{R}^d$  is defined for  $n = N, \dots, 1$  and  $i = 1, \dots, d$  by the recursive equation

$$\bar{\Phi}_i(T) = \int_0^1 \partial_{x_i} g \left( s \overline{X}^h(T) + (1-s) \overline{X}^H(T) \right) ds, \quad (2.21)$$

$$\bar{\Phi}_i(t_{n-1}) = \left( \int_0^1 \partial_{x_i} A^h \left( s \overline{X}^h(t_{n-1}) + (1-s) \overline{X}^H(t_{n-1}) \right) ds, \bar{\Phi}(t_n) \right). \quad (2.22)$$

$$(2.23)$$

**Proof.** Using Richardson extrapolation, we get

$$g(X(T)) - g(\overline{X}(T)) = \frac{1}{\left(\frac{H}{h}\right)^p - 1} \left( g(\overline{X}^h(T)) - g(\overline{X}^H(T)) \right) + o(h^p). \quad (2.24)$$

Therefore it is sufficient to prove that the computable quantity,  $g(\overline{X}^h(T)) - g(\overline{X}^H(T))$ , has the representation

$$g(\overline{X}^h(T)) - g(\overline{X}^H(T)) = \sum_{n=1}^N \left( A^h(\overline{X}^H(t_{n-1})) - A^H(\overline{X}^H(t_{n-1})), \bar{\Phi}(t_n) \right) \quad (2.25)$$

which proves (2.19-2.20) together with (2.24). The initial conditions (2.21), (1.4) and telescoping cancellation show

$$\begin{aligned} g(\overline{X}^h(T)) - g(\overline{X}^H(T)) &= \left( \overline{X}^h(t_N) - \overline{X}^H(t_N), \bar{\Phi}(t_N) \right) \\ &= \sum_{n=1}^N \left[ \left( \overline{X}^h(t_n) - \overline{X}^H(t_n), \bar{\Phi}(t_n) \right) - \left( \overline{X}^h(t_{n-1}) - \overline{X}^H(t_{n-1}), \bar{\Phi}(t_{n-1}) \right) \right]. \end{aligned} \quad (2.26)$$

By the definitions (2.18), the right hand side of (2.26) can be separated into three parts

$$\begin{aligned} & \left( A^h(\overline{X}^h(t_{n-1})) - A^h(\overline{X}^H(t_{n-1})), \overline{\Phi}(t_n) \right), & \left( A^h(\overline{X}^H(t_{n-1})) - A^H(\overline{X}^H(t_{n-1})), \overline{\Phi}(t_n) \right), \\ & \text{and } - \left( \overline{X}^h(t_{n-1}) - \overline{X}^H(t_{n-1}), \overline{\Phi}(t_{n-1}) \right), \end{aligned}$$

where the first and the last parts are canceled out, since the first term can be written

$$\left( A^h(\overline{X}^h(t_{n-1})) - A^h(\overline{X}^H(t_{n-1})), \overline{\Phi}(t_n) \right) = \left( \overline{X}^h(t_{n-1}) - \overline{X}^H(t_{n-1}), \sum_{j=1}^d \Lambda_{.j} \overline{\Phi}_j(t_n) \right)$$

with  $\Lambda_{ij} \equiv \int_0^1 \partial_{x_i} A_j^h \left( s \overline{X}^h(t_{n-1}) + (1-s) \overline{X}^H(t_{n-1}) \right) ds$ , so that  $\sum_j \Lambda_{ij} \overline{\Phi}_j(t_n) = \overline{\Phi}_i(t_{n-1})$  by (2.21). Consequently (2.25) holds.  $\square$

*Approximation of roundoff error.* Theorems 2.2 and 2.3 can be modified to include error caused by roundoff due to finite precision arithmetic. Let  $\hat{e}(t_n)$  be the local roundoff error in one step of the method  $\overline{X}$ , i.e.

$$\hat{e}(t_n) \equiv \widehat{\overline{X}}(t_n) - \overline{X}(t_n),$$

where  $\widehat{\overline{X}}(t_n)$  is the exact arithmetic version of one step of the method  $\overline{X}$ , with the same initial data  $\overline{X}(t_{n-1})$  at time  $t_{n-1}$ . Here,  $\overline{X}$  is computed in finite precision arithmetic. Theorem 2.1 then implies that the part of the global error due to roundoff is the following weighted sum of the local roundoff error

$$\sum_{n=1}^N \left( \hat{e}(t_n), \int_0^1 \Psi(t_n, \overline{X}(t_n) + se(t_n)) ds \right).$$

If the local roundoff error  $\hat{e}(t_n)$  in one step dominates the local discretization error, i.e. if

$$|\hat{e}(t_n)| \geq |\gamma(\widehat{\overline{X}}(t_n) - \overline{X}(t_n))|, \quad (2.27)$$

then the refinement of the time step will not decrease the approximation error; instead a higher precision is needed. The local roundoff error  $\hat{e}$  can be estimated, analogously to the local discretization error, by approximating  $\widehat{\overline{X}}$  with higher precision than  $\overline{X}$ . Alternatively, the approximation

$$fl(fl(\overline{X}(t_n) - \overline{X}(t_{n-1})) - \Delta X(\overline{X}(t_{n-1}))) \simeq \hat{e}(t_n), \quad (2.28)$$

motivated by the compensated summation method by Kahan, cf. [22], is useful when the main roundoff error is caused by the recursive summation  $\overline{X}(t_{n-1}) + \Delta X(\overline{X}(t_{n-1}))$ . Here  $\Delta X(\overline{X}(t_{n-1}))$  is the explicit increment  $\overline{X}(t_n) - \overline{X}(t_{n-1})$ , of the method  $\overline{X}$ , and  $fl$  denotes the rounded operation. The analogous test to (2.27) can be applied also to the dual problem (2.4) for  $\Psi$ , to provide an accurate weight function.

### 3 AN ADAPTIVE ALGORITHM

This section discusses general properties of adaptive algorithms. First we show that in some sense the error representation (2.3) is the only useful global error indicator for adaptive mesh refinements. Then an adaptive algorithm is presented for problem (1.3). The algorithm chooses the number of time steps adaptively, by successively dividing and merging time steps, to bound an approximation of the global error, based on the local error and the variational principle described in Theorem 2.3. The main result is that this algorithm stops with the optimal number of steps, up to a multiplicative constant factor which is independent of the problem (1.3). The true global error is then bounded by the tolerance times a similar problem independent factor, asymptotically as the tolerance tends to zero.

To understand a possible uniqueness of the error representation (2.3), suppose that

$$g(X(T)) - g(\bar{X}(T)) = \sum_{i=1}^N r_i, \quad (3.1)$$

is an alternative error representation to (2.3). What properties are needed of an error representation to be useful for adaptive mesh refinements? A typical adaptive algorithm does two things iteratively:

- (1) if the error indicator is smaller than the given tolerance it stops; otherwise
- (2) the algorithm chooses where to refine the mesh and then makes an iterative step to (1).

Therefore the representation  $r_i$  must, in addition to estimate the global error (3.1) in (1), also give simple information where to refine to reach the optimal mesh. The only practical method seems to link the refinement of element  $i$  to the value of  $r_i$ . Then an ideal error representation satisfies:

- (i) the error contribution  $r_i$  depends only on  $\Delta t_i$ , not on  $\Delta t_j$  for  $i \neq j$ , and
- (ii)  $r_i = o(\Delta t_i)$ .

The following related conditions imply uniqueness

**Theorem 3.1** *Suppose that an error representation satisfies*

$$\text{the error representation (3.1) holds for all choices of step sizes,} \quad (3.2)$$

$$\text{the indicator } r_i \text{ depends only on } \Delta t_k, k = 1, \dots, i, \text{ and not on } \Delta t_j \text{ for } j > i, \quad (3.3)$$

$$\text{and the error indicators have a uniform bound } r_i = o(\Delta t_i). \quad (3.4)$$

Then  $r_i = (e(t_i), \int_0^1 \Psi(t_i, \bar{X}(t_i) + se(t_i)) ds)$ .

**Proof.** Take the limit  $\Delta t_i = 0$ ,  $i = 2, 3, \dots$ . Then by (3.2, 3.4) and Theorem 2.1

$$g(X(T)) - g(\bar{X}(T)) = r_1 = (e(t_1), \int_0^1 \Psi(t_1, \bar{X}(t_1) + se(t_1)) ds). \quad (3.5)$$

Next, let  $\Delta t_i = 0$ ,  $i = 3, 4, \dots$  and use (3.2, 3.3) to get

$$r_1 + r_2 = \sum_{i=1}^2 (e(t_i), \int_0^1 \Psi(t_i, \bar{X}(t_i) + se(t_i)) ds),$$

which together with (3.5) show also

$$r_2 = (e(t_2), \int_0^1 \Psi(t_2, \bar{X}(t_2) + se(t_2))) ds.$$

Continue this inductive argument to prove the theorem for all  $r_i$ .  $\square$

*Adaptive step size control.* To construct an adaptive algorithm, from an error estimate (2.15)

$$g(X(T)) - g(\bar{X}(T)) \simeq \sum_{i=1}^N \bar{r}_i$$

with the goal to bound the global error by a given tolerance parameter TOL using as few time steps  $N$  as possible, it is necessary to have a criterion how to choose the time step,  $\Delta t_i$ , depending on  $\bar{r}_i$ . Let us now motivate the optimal choice

$$\bar{r}_i = \text{constant} \tag{3.6}$$

i.e. local error  $\cdot$  weight = constant, cf. (1.8), for approximation methods which have no essential constraint on the step sizes, such as one step methods (2.9).

For given time steps  $0 = t_0 < \dots < t_N = T$ , let the piecewise constant mesh function  $\Delta t$  be determined by

$$\Delta t(\tau) \equiv \Delta t_i \equiv t_i - t_{i-1} \quad \text{for } \tau \in (t_{i-1}, t_i] \quad \text{and } i = 1, \dots, N.$$

Then the number of time steps that corresponds to a mesh  $\Delta t$ , for the interval  $[0, T]$ , can be defined by

$$N(\Delta t) \equiv \int_0^T \frac{1}{\Delta t(\tau)} d\tau. \tag{3.7}$$

Consider, for  $\tau \in (t_{i-1}, t_i)$  and  $i = 1, \dots, N$ , the piecewise constant function  $\rho \equiv r_i / \Delta t_i^{p+1}$ , which measures the density of the global error from (2.3)

$$\rho(\tau) \equiv \rho_i \equiv \frac{(e(t_i), \int_0^1 \Psi(t_i, \bar{X}(t_i) + se(t_i)) ds)}{\Delta t_i^{p+1}} \tag{3.8}$$

and its approximate counterpart  $\bar{r}_i \equiv |\bar{\rho}_i| \Delta t_i^{p+1}$ , obtained from (2.15) with

$$\bar{\rho}(\tau) \equiv \bar{\rho}_i \equiv \text{sign}(\bar{e}(t_i), \bar{\Psi}(t_i)) \max \left( \frac{|\bar{e}(t_i), \bar{\Psi}(t_i)|}{\Delta t_i^{p+1}}, \delta \right) \tag{3.9}$$

where  $\delta \equiv \sqrt{\text{TOL}}$  and  $\text{sign}(x) = x/|x|$  for  $x \in \mathbb{R} - \{0\}$  and  $\text{sign}(0) = 1$ . The constant  $\delta > 0$  is motivated by the wish that  $\max \Delta t \rightarrow 0$  as  $\text{TOL} \rightarrow 0$ . The two conditions  $\sum |\bar{\rho}_i| \Delta t_i^{p+1} \leq \text{TOL}$

and  $|\bar{\rho}_i|\Delta t_i^{p+1} = \text{constant}$ , for all  $i$ , give  $|\bar{\rho}_i|\Delta t_i^{p+1} \leq \frac{\text{TOL}}{N}$ , for all  $i$ . Assuming there is a constant  $C$  such that

$$|\bar{\rho}_i|\Delta t_i^{p+1} \leq \frac{C\text{TOL}}{N}, \quad \text{for all } i, \quad (3.10)$$

we conclude  $\max \Delta t^p \leq \frac{C\text{TOL}}{T\delta}$ , which indeed implies

$$\max \Delta t \leq \left( \frac{C\sqrt{\text{TOL}}}{T} \right)^{\frac{1}{p}} \rightarrow 0 \quad (3.11)$$

as  $\text{TOL} \rightarrow 0$ .

Let  $\mathcal{T} \equiv \{i : |\bar{\rho}_i| = \delta\}$  and set  $T_0 \equiv \sum_{i \in \mathcal{T}} \Delta t_i$ , which clearly satisfies  $0 \leq T_0 \leq T$ . Then Theorems 2.1 and 2.3 and the definitions of  $\rho$  and  $\bar{\rho}$  imply that

$$g(X(T)) - g(\bar{X}(T)) = \sum_{i=1}^N \rho_i \Delta t_i^{p+1} = \mathbf{E}_{\mathcal{T}} + \int_0^T o(\Delta t^p) dt + \mathcal{O}\left(\text{TOL} \frac{T_0}{T}\right) \quad (3.12)$$

where

$$\mathbf{E}_{\mathcal{T}} \equiv \sum_{i=1}^N \bar{\rho}_i \Delta t_i^{p+1} \quad (3.13)$$

is the computable approximation of the global error and the contribution from  $\delta > 0$  is bounded by

$$\sum_{i \in \mathcal{T}} \sqrt{\text{TOL}} \left( \frac{\sqrt{\text{TOL}}}{T} \right) \Delta t_i = \text{TOL} \frac{T_0}{T}.$$

The refinement criterion (3.6) is motivated by the requirement to minimize the number of steps  $N$  in (3.7) under the constraint

$$\sum_{i=1}^N |\bar{\rho}_i| \Delta t_i^{p+1} = \int_0^T |\bar{\rho}(\tau)| \Delta t^p(\tau) d\tau = \text{TOL}. \quad (3.14)$$

This yields, with a standard application of a Lagrange multiplier, the optimal time steps  $\Delta t^*$  satisfying  $|\bar{\rho}_i| \Delta t_i^{p+1} = \text{constant}$  and

$$\Delta t^* \equiv \frac{\text{TOL}^{\frac{1}{p}}}{|\bar{\rho}|^{\frac{1}{p+1}}} \left( \int_0^T |\bar{\rho}(\tau)|^{\frac{1}{p+1}} d\tau \right)^{-\frac{1}{p}}. \quad (3.15)$$

This condition is optimal only for positive density functions  $\bar{\rho}$ , since otherwise (3.13) and (3.14) may give  $\text{TOL} \gg \mathbf{E}_{\mathcal{T}}$ . To use the sign of the density in an optimal way is not considered in this work. The goal of the adaptive algorithm described below is to construct a partition  $\Delta t$  of  $[0, T]$  such that

$$s_2 \frac{\text{TOL}}{N} \leq |\bar{\rho}_i| \Delta t_i^{p+1} \leq s_1 \frac{\text{TOL}}{N}, \quad \forall i = 1, \dots, N \quad (3.16)$$

where  $s_1$  and  $s_2$  are given constants satisfying  $0 < s_2 < s_1$ . Condition (3.16) is an approximation of the optimal  $\bar{r}_i = \text{constant}$  and  $\sum \bar{r}_i \leq \text{TOL}$ . The remainder of this section analyzes in three theorems an adaptive algorithm based on (3.16) with respect to stopping, accuracy and efficiency.

To achieve (3.16), start with an initial partition  $\Delta t[1]$  and then specify iteratively a new partition  $\Delta t[k+1]$ , from  $\Delta t[k]$ , using the following dividing and merging strategy: for  $i = 1, 2, \dots, N[k]$  let  $\bar{r}_i[k] \equiv |\bar{\rho}_i[k]|(\Delta t_i[k])^{p+1}$  and

$$\text{if } \bar{r}_i[k] > s_1 \frac{\text{TOL}}{N}, \text{ then divide } \Delta t_i[k] \text{ into } M \text{ substeps} \quad (3.17)$$

$$\text{elseif } \max(\bar{r}_i[k], \bar{r}_{i+1}[k]) < s_2 \frac{\text{TOL}}{N}, \text{ then merge } \Delta t_i[k] \text{ and } \Delta t_{i+1}[k] \quad (3.18)$$

into one step, and increase  $i$  by 1,

$$\text{else let the new step be the same as the old.} \quad (3.19)$$

**endif**

Here  $M$  is a given integer greater than 1, which bounds the increment of the number of time steps from one iteration to the next. The following analysis, for fixed  $M$ , can easily be extended to bounded and varying  $M$ . From the above dividing and merging strategy, it is natural to use the following stopping criteria:

$$\text{if } \left( \bar{r}_i[k] \leq S_1 \frac{\text{TOL}}{N}, \quad \forall i = 1, \dots, N \right) \text{ and} \quad (3.20)$$

$$\left( \max(\bar{r}_i[k], \bar{r}_{i+1}[k]) \geq S_2 \frac{\text{TOL}}{N}, \quad \forall i = 1, \dots, N-1 \right) \quad (3.21)$$

**then** we stop.

Here  $S_1$  and  $S_2$  are given constants such that  $0 < S_2 < s_2 < s_1 < S_1$ . The combination of (3.12, 3.13) and (3.20) guarantees a given level of accuracy,  $|\mathbf{E}_T| \leq S_1 \text{TOL}$  and the lower bound (3.21) implies efficiency by almost optimal time steps. When almost all  $\bar{r}_i$  satisfy  $\bar{r}_i < s_1 \frac{\text{TOL}}{N}$ , the reduction of the error may be slow. Therefore the algorithm stops if  $\max_i \bar{r}_i \leq S_1 \frac{\text{TOL}}{N}$  and  $\min_i(\max(\bar{r}_i, \bar{r}_{i+1})) \geq S_2 \frac{\text{TOL}}{N}$  for  $S_2 < s_2 < s_1 < S_1$ .

What is the right choice for the constants  $S_2 < s_2 < s_1 < S_1$  of the dividing and merging strategy? Note that the dividing and merging process may do infinite loops at some time steps if the constant  $s_2$  is too close to  $s_1$ . Clearly, we want to avoid the case where the time step  $\Delta t(t)[k]$  is divided but in the next iteration,  $\Delta t(t)[k+1]$  is merged. To analyze this possible instability, the variation of the density  $\bar{\rho}$  at two consecutive iterations must be understood: since by (3.11),  $\text{TOL} \rightarrow 0+$  implies that  $\max \Delta t \rightarrow 0$ , there is a limit,  $\bar{\rho}$ , of  $\rho$  from (2.13). Similarly,  $\bar{\Psi} \rightarrow \Psi$  by (2.8) and  $\bar{e}/\Delta t^{p+1} - e/\Delta t^{p+1} \rightarrow 0$  by (2.13, 2.14) as  $\max \Delta t \rightarrow 0$ , thus  $|\bar{\rho}| \rightarrow |\rho|$ . A consequence of  $|\bar{\rho}| \rightarrow |\rho|$ , as  $\text{TOL} \rightarrow 0+$ , and (3.8, 3.9) is that for sufficiently small TOL there exists a constant  $c > 0$  such that for all  $t \in [0, T]$

$$c \leq \left| \frac{\bar{\rho}(t)[k+1]}{\bar{\rho}(t)[k]} \right| \leq c^{-1}, \quad (3.22)$$

which implies a related constraint,  $c_1 \leq |\bar{\rho}_i[k]/\bar{\rho}_{i+1}[k]| \leq c_1^{-1}$ , on the optimal mesh, see Remark 3.9.

*Stopping of the adaptive algorithm.* The right choice of the parameters  $S_2 < s_2 < s_1 < S_1$  is explained by

**Theorem 3.2 (Stopping)** *Suppose the assumptions of Theorem 2.3 hold and the adaptive algorithm uses the strategy (3.17, 3.18), (3.20, 3.21). Assume that (3.22) holds with  $c \geq 2^{-p}$ , and that*

$$s_2 \leq \frac{1}{2}cM^{-(p+1)}s_1, \quad (3.23)$$

$$S_2 < \frac{c}{2}s_2, \quad (3.24)$$

$$S_1 > \frac{M}{c}s_1. \quad (3.25)$$

*Then the adaptive algorithm stops by the stopping criteria (3.20, 3.21), after a finite number of operations.*

**Proof.** If  $s_2$  is too close to  $s_1$ , the algorithm may be unstable in the sense that a time step is first divided and then in the next iteration the step is merged, or similarly a step is first merged and then divided. To analyze this unstable situation assume that (3.22) holds. Merging two steps which have just been divided requires that

$$\bar{r}_i[k] > s_1 \frac{\text{TOL}}{N[k]}, \quad \text{and} \quad \max(\bar{r}_i[k+1], \bar{r}_{i+1}[k+1]) < s_2 \frac{\text{TOL}}{N[k+1]} \quad (3.26)$$

with  $\Delta t_i[k+1] = \Delta t_i[k]/M$ , so that by (3.22)

$$\begin{aligned} s_2 &> \frac{N[k+1] |\bar{\rho}[k+1]| \bar{r}[k]}{\text{TOL} |\bar{\rho}[k]| M^{p+1}} \\ &> \frac{N[k+1]}{N[k]} c \frac{s_1}{M^{p+1}} \\ &> \frac{1}{2} c \frac{s_1}{M^{p+1}}, \end{aligned} \quad (3.27)$$

since the number of steps cannot decrease with more than a factor  $2^{-1}$  in the next iteration. Similarly, dividing following merging requires  $s_2 > M^{-1}cs_12^{-(p+1)}$ , which is less demanding than the smaller (3.27). Therefore the condition (3.23) avoids the dividing-merging instability.

The next step is to understand the evolution of

$$\begin{aligned} r_{max}[k] &\equiv \max_i \bar{r}_i[k], \\ r_{min}[k] &\equiv \min_i \max(\bar{r}_i[k], \bar{r}_{i+1}[k]) \end{aligned}$$

for iterations  $k = 1, 2, \dots$ . It is advantageous if  $r_{max}$  decreases and  $r_{min}$  increases quickly to levels close to the bounds  $s_1\text{TOL}/N$  and  $s_2\text{TOL}/N$ , respectively. Indeed, there holds

$$r_{max}[k+1] < \frac{c^{-1}}{M^{p+1}} r_{max}[k], \quad (3.28)$$

provided  $r_{max}[k] > M^{p+1}s_1\text{TOL}/N[k]$ , and

$$r_{min}[k+1] > c2^{p+1}r_{min}[k], \quad (3.29)$$

provided  $r_{min}[k] < 2^{-(p+1)}s_2\text{TOL}/N[k]$ . In other words, the error indicators  $r_{max}$  and  $r_{min}$  decrease and increase, respectively, with a constant factor away from the bounds

$$r_{max} > M^{p+1}s_1\text{TOL}/N[k] \text{ and } r_{min} < 2^{-(p+1)}s_2\text{TOL}/N[k].$$

If  $r_{max}$  is close to the dividing bound, i.e. if

$$r_{max}[k] \leq M^{p+1}s_1\text{TOL}/N[k], \quad (3.30)$$

then in the next iteration  $r_{max}[k+1] \leq c^{-1}s_1\text{TOL}/N[k]$ , since a divided step cannot be merged by (3.23). The new dividing bound becomes

$$\frac{s_1\text{TOL}}{N[k+1]} \geq \frac{s_1\text{TOL}}{MN[k]} \geq \frac{c}{M}r_{max}[k+1],$$

and (3.30) remains satisfied for the later iterations provided  $c^{-1} \leq M^p$ .

Similarly, if  $r_{min}$  is close to the merging bound, i.e. if

$$r_{min}[k] \geq 2^{-(p+1)}s_2\text{TOL}/N[k], \quad (3.31)$$

then  $r_{min}[k+1] \geq cs_2\text{TOL}/N[k]$  and the new merging bound becomes

$$\frac{s_2\text{TOL}}{N[k+1]} \leq \frac{s_2\text{TOL}}{\frac{N[k]}{2}} \leq \frac{2}{c}r_{min}[k+1], \quad (3.32)$$

and (3.31) remains satisfied for the later iterations provided  $c \geq 2^{-p}$ .

Therefore  $r_{max}$  decreases with the factor (3.28) until  $r_{max}$  is close to the dividing region, where (3.30) holds, and afterwards  $r_{max}$  remains in the stopping region  $r_{max} < S_1\text{TOL}/N$ , provided  $S_1 > Ms_1/c$ . Similarly,  $r_{min}$  increase with the factor (3.29) until  $r_{min}$  is close to the merging region, where (3.31) holds, and afterwards  $r_{min}$  remains in the stopping region  $r_{min} > S_2\text{TOL}/N$ , provided  $S_2 < cs_2/2$ . When both  $r_{max}$  and  $r_{min}$  have entered their stopping regions, the algorithm stops.  $\square$

*Accuracy of the adaptive algorithm.* The adaptive algorithm guarantees that the estimate of the global error is bounded by a given error tolerance, TOL. The next question is whether the true global error is bounded by TOL asymptotically. Using the upper bound (3.20) of the error indicators and the convergence of  $\rho$  and  $\bar{\rho}$  in (3.8, 3.9), the global error has the estimate

**Theorem 3.3 (Accuracy)** *Suppose that the assumptions of Theorem 2.3 hold. Then the adaptive algorithm (3.17, 3.18), (3.20, 3.21) satisfies*

$$\limsup_{\text{TOL} \rightarrow 0^+} \text{TOL}^{-1}|g(X(T)) - g(\bar{X}(T))| \leq S_1. \quad (3.33)$$

**Proof.** When the adaptive algorithm stops, (3.12) and (3.20) imply

$$\begin{aligned} \text{TOL}^{-1}|g(X(T)) - g(\bar{X}(T))| &\leq \text{TOL}^{-1} \sum_{i=1}^N \Delta t_i^p \int_{t_{i-1}}^{t_i} |\rho(\tau)| d\tau \\ &\leq \text{TOL}^{-1} \left( S_1 \frac{\text{TOL}}{N} \right)^{\frac{p}{p+1}} \sum_{i=1}^N \int_{t_{i-1}}^{t_i} \frac{|\rho(\tau)|}{|\bar{\rho}(\tau)|^{\frac{p}{p+1}}} d\tau. \end{aligned} \quad (3.34)$$

Rewrite the inequality (3.20) as

$$|\bar{\rho}|^{\frac{1}{p+1}} \leq \left( S_1 \frac{\text{TOL}}{N} \right)^{\frac{1}{p+1}} \frac{1}{\Delta t_i},$$

integrate both sides and use the definition (3.7) to obtain

$$N^{-\frac{p}{p+1}} \leq (S_1 \text{TOL})^{\frac{1}{p+1}} \frac{1}{\int_0^T |\bar{\rho}(\tau)|^{\frac{1}{p+1}} d\tau}.$$

Apply this to the right hand side of (3.34) to get

$$\text{TOL}^{-1} |g(X(T)) - g(\bar{X}(T))| \leq S_1 \frac{\int_0^T |\rho(\tau)| / |\bar{\rho}(\tau)|^{\frac{p}{p+1}} d\tau}{\int_0^T |\bar{\rho}(\tau)|^{\frac{1}{p+1}} d\tau}. \quad (3.35)$$

Since by (3.10) and (3.11)  $\max \Delta t \rightarrow 0$  and consequently  $\rho$  and  $\bar{\rho}$  converges to  $\tilde{\rho}$  as  $\text{TOL} \rightarrow 0+$ , the fraction in (3.35) converges to 1 by the Lebesgue dominated convergence theorem, which proves (3.33).  $\square$

*Efficiency of the adaptive algorithm.* The next issue for the adaptive method is efficiency. We want to determine a partition with as few time steps as possible providing the desired accuracy. From the definition (3.7) and the optimality condition (3.15), the number of optimal adaptive steps,  $N^{\text{opt}}$ , satisfies

$$N^{\text{opt}} = \int_0^T \frac{1}{\Delta t^*(\tau)} d\tau = \frac{1}{\text{TOL}^{\frac{1}{p}}} \left( \int_0^T |\bar{\rho}[k](\tau)|^{\frac{1}{p+1}} d\tau \right)^{\frac{p+1}{p}},$$

i.e.

$$N^{\text{opt}} = \frac{1}{\text{TOL}^{\frac{1}{p}}} \|\bar{\rho}\|_{L^{\frac{1}{p+1}}}^{\frac{1}{p}}. \quad (3.36)$$

Here  $p > 0$  is the order of accuracy of the approximate solution and  $\|\cdot\|_{L^{\frac{1}{p+1}}}$  is the quasi-norm defined by  $\|f\|_{L^{\frac{1}{p+1}}} \equiv \left( \int_0^T |f(x)|^{\frac{1}{p+1}} dx \right)^{p+1}$ .

On the other hand, for the uniform steps  $\Delta t = \text{constant}$ , the number of steps,  $N^{\text{uni}}$ , to achieve  $\sum_{i=1}^N |\bar{\rho}_i| \Delta t_i^{p+1} = \text{TOL}$ , becomes

$$N^{\text{uni}} = \int_0^T \frac{1}{\Delta t(\tau)} d\tau = \frac{T}{\text{TOL}^{\frac{1}{p}}} \left( \int_0^T |\bar{\rho}[k](\tau)| d\tau \right)^{\frac{1}{p}},$$

i.e.

$$N^{\text{uni}} = \frac{T}{\text{TOL}^{\frac{1}{p}}} \|\bar{\rho}\|_{L^1}^{\frac{1}{p}}. \quad (3.37)$$

Hence, the number of steps is measured in the  $L^1$ -norm for a uniform method and the optimal number of steps is measured in the  $L^{\frac{1}{p+1}}$  quasi-norm for an adaptive method. Therefore an adaptive method uses fewer time steps than a uniform method, since by Jensen's inequality  $\|f\|_{L^{\frac{1}{p+1}}} \leq T^p \|f\|_{L^1}$ , see Remarks 3.7 and 3.8, (1.9) and Example 4.2.

The following theorem uses the lower bound (3.21) of the error indicators to show that the algorithm (3.17 - 3.21) generates a mesh which is optimal, up to a multiplicative constant  $C \simeq (2^{p+1}/S_2)^{1/p}$ .

**Theorem 3.4 (Efficiency)** *Suppose that the assumptions of Theorem 2.3 hold. Then there exists a constant  $C > 0$ , bounded by  $3(3/S_2)^{1/p}$ , such that the number of adaptive steps  $N$ , of the algorithm (3.17, 3.18), (3.20, 3.21), satisfies*

$$\text{TOL}^{\frac{1}{p}} N \leq C \|\bar{\rho}\|_{L^{\frac{1}{p+1}}}^{\frac{1}{p}}. \quad (3.38)$$

**Proof.** When the adaptive algorithm stops, by the stopping criteria (3.20, 3.21), the set of steps satisfies

$$\{\Delta t_i : i = 1, \dots, N\} = \mathcal{D} \cup \mathcal{M} \quad \text{and} \quad \mathcal{D} \cap \mathcal{M} = \emptyset$$

where

$$\begin{aligned} \mathcal{D} &:= \left\{ \Delta t_i : S_2 \frac{\text{TOL}}{N} \leq |\bar{\rho}_i| \Delta t_i^{p+1} \leq S_1 \frac{\text{TOL}}{N}, \quad i = 1, \dots, N \right\} \\ \mathcal{M} &:= \left\{ \Delta t_i : |\bar{\rho}_i| \Delta t_i^{p+1} < S_2 \frac{\text{TOL}}{N}, \quad i = 1, \dots, N \right\}. \end{aligned}$$

From the condition (3.21), we know that there is no successive pair of steps which belongs to the set  $\mathcal{M}$ . This means that the number of steps  $N_{\mathcal{M}}$  in  $\mathcal{M}$  is *at most* the same as the half of the number of steps,  $N$ , i.e.

$$N_{\mathcal{M}} \leq \left\lceil \frac{N}{2} \right\rceil.$$

Here  $\lceil a \rceil$  rounds the number  $a$  to the nearest integer greater than or equal to  $a$ . Denote the number of steps in  $\mathcal{D}$  by  $N_{\mathcal{D}}$ , so that

$$N = N_{\mathcal{D}} + N_{\mathcal{M}} \leq N_{\mathcal{D}} + \left\lceil \frac{N}{2} \right\rceil,$$

which implies

$$N \leq 2N_{\mathcal{D}} + 1 \leq 3N_{\mathcal{D}}. \quad (3.39)$$

The time steps  $\Delta t_i \in \mathcal{D}$  satisfy

$$\left( S_2 \frac{\text{TOL}}{N} \right)^{\frac{1}{p+1}} \frac{1}{\Delta t_i} \leq |\bar{\rho}|^{\frac{1}{p+1}}.$$

Integrating both sides and using the definition (3.7) and the fact (3.39) yields

$$\left( S_2 \frac{\text{TOL}}{N} \right)^{\frac{1}{p+1}} \frac{N}{3} \leq \left( S_2 \frac{\text{TOL}}{N} \right)^{\frac{1}{p+1}} N_{\mathcal{D}} \leq \int_0^T |\bar{\rho}(\tau)|^{\frac{1}{p+1}} d\tau. \quad (3.40)$$

Consequently,

$$\text{TOL}^{\frac{1}{p}} N \leq C \left( \int_0^T |\bar{\rho}(\tau)|^{\frac{1}{p+1}} d\tau \right)^{\frac{p+1}{p}}, \quad (3.41)$$

where  $C = (3^{p+1}/S_2)^{\frac{1}{p}}$ . A more precise use of the first inequality in (3.39) shows  $C \rightarrow (\frac{2^{p+1}}{S_2})^{\frac{1}{p}}$ , as  $N \rightarrow \infty$ .  $\square$

*The adaptive algorithm.* In this subsection we present a detailed implementation, called MSTZ, of the adaptive algorithm (3.17-3.21), including also a variant allowing the tolerance to decrease slightly as the mesh is refined. The varying tolerance is motivated by efficiency: the efficiency of the algorithm depends on the total work including all refinement levels. If the number of elements in each refinement iteration increase only very slowly, the total work becomes proportional to the product of the number of steps in the finest mesh times the number of refinement levels. The condition (3.15) shows that the number of dividing levels,  $J$ , satisfies

$$\min \Delta t = M^{-J} T / N[1] = \mathcal{O}(\text{TOL}^{1/p}). \quad (3.42)$$

A relation  $\min \Delta t = \mathcal{O}(\text{TOL}^\alpha)$ ,  $\alpha > 0$  still holds for many singular densities, as in Example 4.2. Therefore,  $J = \mathcal{O}(\frac{1}{p} \log(\text{TOL}^{-1})) \simeq \log N$ , so that the total number of steps for the algorithm (3.17-3.21) would be essentially bounded by

$$N \log N. \quad (3.43)$$

A more efficient refinement algorithm is obtained by successively decreasing the tolerance,  $\text{TOL}[k+1] < \text{TOL}[k]$ , in each refinement so that

$$\frac{N[k]}{N[k+1]} \leq \bar{c} < 1 \quad (3.44)$$

always holds. The condition (3.44) would imply that the total number of steps satisfy

$$\sum_{k=1}^J N[k] \leq \frac{N[J]}{1-\bar{c}}. \quad (3.45)$$

Therefore, a slightly decreasing tolerance may be more efficient than a constant tolerance, which yields the total work (3.43). Including the assumption

$$c' \leq \frac{\text{TOL}[k+1]}{\text{TOL}[k]} \leq 1 \quad (3.46)$$

and replacing  $c$  by  $c'$  everywhere in Theorem 3.2 directly generalizes Theorems 3.2, 3.3 and 3.4 to slightly varying tolerance, where TOL in (3.33) and (3.38) then denotes the final stopping tolerance. However, an unattractive consequence of varying tolerance is that the stopping tolerance becomes a priori uncertain, see Remark 3.5 and Theorem 3.6.

Let us now introduce some useful notation and then describe the algorithm MSTZ in detail. Our dividing and merging strategy (3.17, 3.18) is applied iteratively until the approximate solution is sufficiently resolved, in other words, until the approximate error density  $\bar{\rho}$  and the time steps satisfy the stopping criteria (3.20, 3.21). To check this, first we set a global error indicator,  $\mathbf{G}$ , to 0 and for each refinement we change  $\mathbf{G}$  to 1 if  $\bar{\rho}$  satisfies (3.20, 3.21). Similarly to detect the roundoff error, first set the roundoff error indicator,  $\mathbf{R}$ , to 0 then for each refinement change  $\mathbf{R}$  to 1 if roundoff error is significant. Now we are ready to define the adaptive algorithm MSTZ:

**Initialization** The user chooses

1. an initial error tolerance, TOL,

2. a number,  $N[1]$ , of initial uniform steps  $\Delta t[1]$  for  $[0, T]$ ,
3. an integer,  $M$ , which bounds the increment of the number of time steps,
4. a number,  $s_1$  in (3.17) and a rough estimate of  $c$  in (3.22) to compute  $s_2, S_1, S_2$  using (3.23, 3.24, 3.25), and
5. a factor  $\bar{c}$  to increase the number of steps in (3.44).

Set the iteration number  $k$ , the roundoff error indicator,  $\mathbf{R}$ , and the global error indicator,  $\mathbf{G}$ , to 0.

**Step 1** Increment the iteration number  $k$  by 1. For  $n = 1, \dots, N[k]$ , compute the approximation  $\bar{X}(t_n)$  of (1.3) using a  $p$ -th order accurate numerical method (2.9), and to obtain the local error, compute the approximate local exact solution  $\bar{\bar{X}}(t_n)$  of (1.7) using a higher accuracy than for  $\bar{X}(t_n)$ . Compute the approximation of the local error  $\bar{e}(t_n)$  by (2.14) and the approximate weight  $\bar{\Psi}(t_n)$ , for  $n = N[k], \dots, 1$ , using the  $p$ -th order accurate method (2.10).

**Step 2** Determine  $\mathbf{R}$  by checking (2.27), using (2.28) and  $\mathbf{G}$  by checking (3.20, 3.21) and

**if**  $\mathbf{R} = 1$ , go to **Step 3**,

**elseif**  $\mathbf{G} = 1$ , go to **Step 4**,

**endif**

**do** for all time steps  $i = 1, \dots, N[k]$

**if**  $\left( \bar{r}_i[k] > s_1 \frac{\text{TOL}}{N[k]} \right)$ , then  $\Delta t(t)[k+1] = \frac{\Delta t_i[k]}{M}$ ,  $t_{i-1}[k] < t \leq t_i[k]$ ,

**elseif**  $\left( \max(\bar{r}_i[k], \bar{r}_{i+1}[k]) < s_2 \frac{\text{TOL}}{N[k]} \right)$

then  $\Delta t(t)[k+1] = \Delta t_i[k] + \Delta t_{i+1}[k]$ ,  $t_{i-1}[k] < t \leq t_{i+1}[k]$ , and increase  $i$  by 1,

**else**  $\Delta t(t)[k+1] = \Delta t_i[k]$ ,  $t_{i-1}[k] < t \leq t_i[k]$ ,

**endif**

**enddo**

**go to Step 1.**

**Step 3** Terminate the program and report the results since roundoff error is detected.

**Step 4** Stop the program since the global error is bounded by the given error tolerance, TOL.

**Remark 3.5** *A diminished tolerance is useful if there are few steps with their error indicators,  $\bar{r}_i$ , in the set  $(s_1 \text{TOL}/N, \infty)$ . Modify the algorithm by adding the command “Set  $\mathbf{V} = 0$ ” in the end of **Step 1** and replace the statement “**go to Step 1**” after **enddo** in the end of Step 2 by:*

**if**  $(N[k]/N[k+1] > \bar{c} \ \& \ \mathbf{V} = 0)$ , then  $\text{TOL} \equiv \text{TOL}[k](1 - \frac{\bar{c}^{-1}-1}{M-1})$ ,  $\mathbf{V} = 1$  and go to **Step 2**,

**else** go to **Step 1**.

**endif**

Assume that the set  $(c's_1 TOL/N, s_1 TOL/N]$  contains a fraction  $c''N$  of the steps, where  $M^{-p} < c' < 1$ ; for instance, if the error indicators,  $\bar{r}_i$ , are uniformly distributed in  $[s_2 TOL/N, s_1 TOL/N]$ , with a negligible part outside of this set, there holds  $c'' = \frac{s_1(1-c')}{s_1-s_2}$ , which for negligible  $s_2$  yields  $\bar{c} = \frac{1}{1+c''(M-1)} = \frac{1}{1+(1-c')(M-1)}$  and motivates  $c' = 1 - \frac{\bar{c}^{-1}-1}{M-1}$  in the algorithm. A refinement approximately maps the set  $(c's_1 TOL/N, s_1 TOL/N]$  to  $(c's_1 TOL/(NM^{p+1}), s_1 TOL/(NM^{p+1})]$ . Then the next refinement continues with essentially a similar distribution of the error indicators, provided  $c'$  is not too small. When the algorithm stops, the final tolerance satisfies  $TOL[0] \geq TOL[J] \geq TOL[0](c')^J = TOL^{1+\mathcal{O}(\frac{\log c'}{p})}$ , which for  $c'$  close to 1 is only a slight change.

Let us now show that the total number of steps can be bounded by a constant times the number of steps in the finest mesh. Its proof uses that the tolerance decreases sufficiently, which simplifies the analysis. A more refined study, with less assumptions on the tolerance, following the idea in Remark 3.5 would need deeper understanding of the distribution of the error indicators  $\bar{r}_i$ .

**Theorem 3.6** *The total number of steps satisfies the bound  $\sum_{k=1}^J N[k] = \mathcal{O}(N[J])$ , for a variant of MSTZ where all levels have decreasing tolerance  $TOL[k+1] = TOL[k]c'$  satisfying  $0 < c' < c$ , provided all initial steps are divided and (3.22) and the conditions in Theorem 3.2 hold with  $c$  replaced by  $c'$ .*

**Proof.** Let us first verify that the algorithm will not do any merging if all initial steps are divided. For a step to be merged it has to enter into the merging region at a previous level. Let  $\mathcal{N}_0[k] \equiv \{i : s_2 TOL[k]/N[k] \leq \bar{r}_i[k] \leq s_1 TOL[k]/N[k]\}$  and assume  $s_2 TOL[k]/N[k] \leq \min_n \bar{r}_n[k] \leq s_1 TOL[k]/N[k]$ , then  $N[k+1] > N[k]$ . Consequently  $\bar{r}_n[k+1] \geq c\bar{r}_n[k] > s_2 TOL[k+1]/N[k+1]$  for  $n \in \mathcal{N}_0[k]$ , so that the error indicator cannot enter into the merging region and therefore there is no merging.

Since the algorithm will not do any merging, we have  $N[k] < N[k+1]$  for all  $k$ . Assume that

$$\frac{N[k]}{N[k+1]} > \bar{c}, \quad k = K, \dots, K+m, \quad (3.47)$$

where  $m$  and  $\bar{c}$  satisfy

$$\left(\frac{c'}{c}\right)^m < \frac{s_2}{s_1}, \quad (3.48)$$

$$1 < \bar{c}^{-1} < M^{1/(m+1)}, \quad (3.49)$$

and let  $N_0[k] \equiv \#\mathcal{N}_0[k]$  and  $N_+ \equiv N - N_0$ . The condition (3.47) shows that the number of divided steps,  $N_+[k]$ , satisfies

$$N_+[k] < \frac{\bar{c}^{-1}-1}{M-1} N[k]. \quad (3.50)$$

The tolerance decreases, so that after  $m$  levels the dividing barrier is

$$s_1 TOL[K+m]/N[K+m] < (c')^m s_1 TOL[K]/N[K].$$

All elements in  $\mathcal{N}_0[K]$  must have been divided after  $m$  levels, since if they have not all been divided some error indicators are larger than  $c^m s_2 \text{TOL}[K]/N[K]$  and condition (3.48) gives the contradiction

$$s_1 \frac{\text{TOL}[K+m]}{N[K+m]} < (c')^m s_1 \frac{\text{TOL}[K]}{N[K]} < c^m s_2 \frac{\text{TOL}[K]}{N[K]}.$$

Dividing of all steps in  $\mathcal{N}_0[K]$  shows that  $N_0[K]$  must be smaller than the sum of divided steps

$$N_0[K] \leq \sum_{j=1}^m N_+[K+j] \tag{3.51}$$

which also leads to a contradiction, since by (3.47) and (3.50)

$$N_0[K] > \frac{M - \bar{c}^{-1}}{M - 1} N[K]$$

and

$$N_+[K+j] < \frac{\bar{c}^{-1} - 1}{M - 1} \bar{c}^{-j} N[K],$$

so that by the assumption (3.49)

$$N_0 - \sum_{j=1}^m N_+[K+j] > \frac{M - \bar{c}^{-m-1}}{M - 1} N[K] > 0,$$

which contradicts (3.51). Hence, the number of consecutive levels, where  $N[k]/N[k+1] > \bar{c}$ , must be smaller than  $m$  and therefore

$$\sum_{k=1}^J N[k] \leq \frac{mN[J]}{1 - \bar{c}} = \mathcal{O}(N[J]).$$

□

**Remark 3.7** *The discussion on the non convex sets (1.9) in the introduction is inspired by the work [31], which includes an elegant proof of Bakhvalov and Smolyak's result following [5]: Assume that  $\mathcal{F}$  is a convex symmetric subset of a normed linear function space and that  $S : \mathcal{F} \rightarrow \mathbb{R}$  is a linear functional with an approximation  $S_N(f) = \phi(L_1(f), \dots, L_N(f))$ , based on  $N$  linear functionals  $L_k : \mathcal{F} \rightarrow \mathbb{R}$ , which may depend on  $f \in \mathcal{F}$ , and a linear or nonlinear mapping  $\phi : \mathbb{R}^N \rightarrow \mathbb{R}$ . Let the worst case error for  $S_N$  be defined by*

$$\Delta_{\max}(S_N) \equiv \sup_{f \in \mathcal{F}} |S(f) - S_N(f)|,$$

*and assume that  $S_N$  uses the linear functionals  $\{L_k^0\}_{k=1}^N$  for  $f \equiv 0 \in \mathcal{F}$ . Then there exists  $a = (a_1, \dots, a_N) \in \mathbb{R}^N$  and a linear non-adaptive method*

$$S_N^*(f) \equiv \sum_{k=1}^N a_k L_k^0(f),$$

such that

$$\Delta_{\max}(S_N^*) \leq \Delta_{\max}(S_N).$$

An example is computation of integrals  $S(f) = \int_0^1 f(t)dt$  with the approximation  $S_N(f)$  based on  $N$  point values  $L_k(f) = f(t_k)$  at mesh points  $t_k$ ,  $k = 1, \dots, N$ . The method is adaptive if  $t_k$  depends on  $f$  and non adaptive otherwise.

**Remark 3.8** Let us consider integration by the first order accurate Euler method. Then the integration error is the same as the  $L^1$  approximation error by piecewise constant functions. DeVore points out in [11] that this  $L^1$  approximation error of a function  $f$ , with  $N$  non-adaptive steps, is  $\mathcal{O}(N^{-\alpha})$  provided  $f$  belongs to the Besov space  $B_\infty^\alpha(L^1[0, T])$ ,  $\alpha \leq 1$ . With  $N$  adaptive steps the error is  $\mathcal{O}(N^{-\alpha'})$  provided  $f$  belongs to the Besov space  $B_\infty^{\alpha'}(L^\gamma[0, T])$ ,  $\alpha' \leq 1$ , for some  $\gamma > (\alpha' + 1)^{-1}$ . For  $\alpha' \rightarrow 1-$ , this explains that adaptive integration is better when  $\|f'\|_{L^{\frac{1}{2}}(0,1)} \ll \|f'\|_{L^1(0,1)}$ , cf. (3.36, 3.37).

**Remark 3.9** The error density condition (3.22) also implies constraints on the optimal mesh, for instance  $M = 2$  and the assumption  $\frac{1}{2}(\bar{\rho}_i[k] + \bar{\rho}_{i+1}[k]) = \bar{\rho}(t_i)[k - 1]$  shows that

$$2c - 1 \leq \left| \frac{\bar{\rho}_{i+1}[k]}{\bar{\rho}_i[k]} \right| \leq 2c^{-1} - 1. \quad (3.52)$$

**Remark 3.10** Here the time steps are chosen to be the same for all components of the solution. Logg [26] uses the more efficient and flexible choice of independent steps for each component of the solution. The error estimates (2.15) and (2.19) would also be applicable to such multi adaptive time steps  $\Delta t_{n,i}$  by replacing  $\Delta t_n^{p+1} \rho_n$  with  $\sum_n \sum_i \Delta t_{n,i}^{p+1} \rho_{n,i}$ , where  $\rho_{n,i} \equiv \bar{e}_i(t_n) \bar{\Psi}_i(t_n) / \Delta t_{n,i}^{p+1}$ .

## 4 NUMERICAL EXPERIMENTS

This section presents numerical experiments from the implementation of the adaptive algorithm MSTZ in Section 3 using the MATLAB version 5.3 software package. To study its performance, we choose the Lorenz problem and a simple problem with a singularity. For the initialization of MSTZ, we set  $s_1 = 2$  in (3.17) and from the conditions of Theorem 3.2 we let  $s_2 = s_1 / (20 M^{p+1})$ ,  $S_1 = 2M s_1$ ,  $S_2 = s_2 / (2M)$  and  $M = 2$ . We investigate the performance of the algorithm MSTZ and compare the approximation with the adaptive algorithm ODE45 in MATLAB and with a constant step size algorithm, denoted Uniform. In particular we study the quality of the error estimate, by comparing the ratio between the exact error and the approximate error in (3.13), defined by

$$\Gamma \equiv \frac{E_T}{|g(X(T)) - g(\bar{X}(T))|}. \quad (4.1)$$

We also compare the final number of time steps,  $N_T$ , and the total number of time steps, defined by

$$N^{\text{tot}} \equiv \sum_{k=1}^J N[k], \quad (4.2)$$

where  $J$  is the total number of refinement levels.

**Example 4.1** Consider the well-known Lorenz system, which is the three dimensional system of ordinary differential equations,

$$\begin{aligned} a_1(t, x) &= -\sigma x_1 + \sigma x_2, \\ a_2(t, x) &= rx_1 - x_2 - x_1x_3, \quad 0 \leq t \leq T, \quad x \in \mathbb{R}^3 \\ a_3(t, x) &= x_1x_2 - bx_3 \end{aligned} \tag{4.3}$$

where  $\sigma, r$  and  $b$  are the given positive constants and  $(x_1(0), x_2(0), x_3(0))$  are the given initial data.

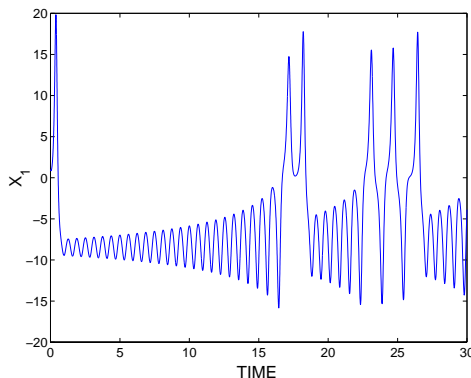


Figure 1: (Example 4.1) Approximate  $x_1$ -component from MSTZ with TOL =  $10^{-1}$ .

The Lorenz system was introduced to show the limitation of large time prediction for a simplified model of weather forecast, see [27]. In our experiments, the coefficient values are  $\sigma = 10, b = 8/3$  and  $r = 28$  and the initial value is  $X(0) = (1, 0, 0)$ , cf. [7] and [17]. The computed function is  $g(x) = x_1$ , i.e. we study the global error  $|X_1(T) - \bar{X}_1(T)|$  at the final time  $T = 30$ .

**MATLAB** uses the fixed IEEE floating-point arithmetic with relative accuracy,  $eps = 2^{-52}$ , which is approximately  $2.22 \times 10^{-16}$ . Hence if we choose TOL too small, the approximation is affected with roundoff error. A computation with a **Fortran** implementation of MSTZ in quadruple precision gives us the approximate value  $g_1 \simeq -3.892637$ , with TOL =  $10^{-7}$ . **ODE45**, which is a built in function in **MATLAB**, is based on an explicit Runge-Kutta (5,4) formula, the Dormand-Prince pair, see [12]. In order to compare with **ODE45**, the program **MSTZ** also uses the same 5-th order explicit Runge-Kutta method (Dormand-Prince method) to compute  $\bar{X}(t_{n+1})$ ,  $\bar{\Psi}(t_n)$  and  $\bar{X}(t_{n+1})$ . The approximate local “exact” solution  $\bar{X}$  is computed with the half mesh size ( i.e.  $\gamma = 2^5/(2^5 - 1)$  in (2.14) ). The program **Uniform** uses a constant step size,  $\Delta t = constant$ , based on the same 5-th order explicit Runge-Kutta method.

Table 1 compares adaptive steps of **MSTZ**, **ODE45** and constant steps of **Uniform**. The program **MSTZ** obtains the reliable approximations  $g(\bar{X}(T)) = -3.8834$  with  $\Gamma = 0.9908$  and  $g(\bar{X}(T)) = -3.8900$  with  $\Gamma = 0.9971$  for  $N[1] = 300$  and TOL =  $10^{-1}, 10^{-2}$  respectively, see Figure 1. On the other hand, the program **ODE45** yields  $g(\bar{X}(T)) \simeq -3.8497$  for the case TOL =  $10^{-10}$  which

means that a combination of the relative local error tolerance of  $10^{-10}$  and absolute local error tolerance of  $10^{-10}$  for all  $x$  components are chosen. Similarly ODE45 yields  $g(\bar{X}(T)) \simeq -3.8882$  for  $\text{TOL} = 10^{-11}$ .

In short, the algorithm MSTZ achieves higher accuracy with half the final number of time steps compared to Uniform and with one fifth of the final number of steps compared to ODE45. The total number of time steps, including all refinement levels, of MSTZ is  $N^{\text{tot}} = 20226$  with 7 refinement levels and  $N^{\text{tot}} = 33544$  with 8 refinement levels for  $\text{TOL} = 10^{-1}$ ,  $10^{-2}$  respectively, which are even smaller than the final numbers of time steps of ODE45. With  $\text{TOL}$  decreasing a factor of 10 from  $10^{-1}$  to  $10^{-10}$ , the total number of steps for ODE45 is 91554 (for  $N_T = 33769$ ).

Figure 2 shows a comparison of the mesh function of MSTZ with  $\text{TOL} = 10^{-1}$  and ODE45 with  $\text{TOL} = 10^{-10}$  using a base 10 logarithmic scale for both the vertical-axis and the horizontal axis. The minimum value of  $\Delta t$  of MSTZ is 0.0016 which is 22 times larger than the minimum of ODE45.

MSTZ	Error	0.01	0.003
	$N_T$	6324	9320
Uniform	Error	0.02	0.004
	$N_T$	12000	17000
ODE45	Error	0.04	0.004
	$N_T$	33769	53481

Table 1: Example 4.1: Comparisons of the final number of steps,  $N_T$ , with the global error,  $\text{Error} \equiv |g_1 - g(\bar{X}(T))|$ , using a 5-th order explicit Runge-Kutta method with adaptive steps for MSTZ or ODE45 and uniform steps for Uniform.

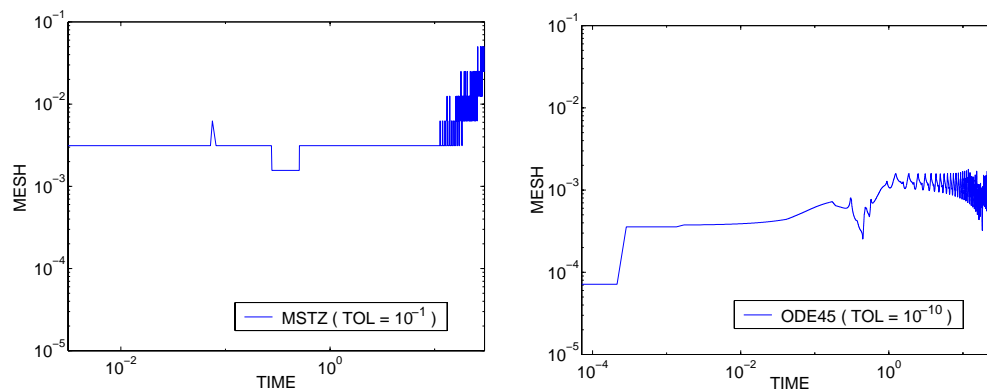


Figure 2: (Example 4.1) Comparison of the mesh functions of MSTZ and ODE45.

**Example 4.2** Consider (1.3) with

$$a(t, x) = \frac{x}{\sqrt{|t - \omega|}}, \quad 0 \leq t \leq T, \quad x \in \mathbb{R} \quad (4.4)$$

and  $X(0) = e^{-2\sqrt{\omega}}$ . The exact solution is

$$X(t) = e^{\text{sign}(t-\omega) 2\sqrt{|t-\omega|}}$$

where  $\omega \in [0, T]$  is a constant and let  $g(x) = x$ .

The function  $a$  in Example 4.2 has a singularity at  $t = \omega$ . The corresponding error density (3.8),  $\rho_p \equiv \rho$  for a  $p$ -th order method, satisfies  $\|\rho_p\|_{L^1} = \infty$  for  $p \geq 1$  and interpolation between the first order  $\rho_1$  and the zero order  $\rho_0 \equiv a$  shows  $\|\rho_p\|_{L^1} < \infty$  for  $p < 1/2$ , so that by (3.37) the number of uniform steps becomes  $N^{\text{uni}} \sim \text{TOL}^{-2}$ . In contrast, the convergence rate for adaptive approximation remains by (3.36) optimal  $N = \mathcal{O}(\text{TOL}^{-1/p})$ , since  $\|\rho_p\|_{L^{\frac{1}{p+1}}} < \infty$  for  $p > 0$ .

Consider the case  $\omega = 5/3$  with  $T = 4$ . **Uniform** requires 524288 final time steps for  $|g(X(T)) - g(\bar{X}(T))| = 0.029934$ , but on the other hand the program **MSTZ** requires only 36 final time steps and  $N^{\text{tot}} = 510$  to obtain  $|g(X(T)) - g(\bar{X}(T))| = 0.010059$  using a 5-th order explicit Runge-Kutta method with  $\text{TOL} = 10^{-1}$  and  $N[1] = 2^5$ . The uniform method cannot achieve higher accuracy than 0.029934 due to the roundoff error, however **MSTZ** obtains  $|g(X(T)) - g(\bar{X}(T))| = 2.4578 \times 10^{-5}$  using  $N_T = 125$  and  $N^{\text{tot}} = 3882$ . Figure 3 shows that **MSTZ** detects the singularity at  $t = 5/3$  correctly and merges the mesh efficiently because the mesh function  $\Delta t$  satisfies  $\Delta t[1]/4 \leq \Delta t \leq \Delta t[1]$  except near the singularity, using  $\text{TOL} = 10^{-4}$  and  $N[1] = 2^5$ . The total number of steps for **ODE45** with  $\text{TOL}$  decreasing a factor of 10 from  $10^{-1}$  to  $10^{-5}$  is 481 (for  $N_T = 209$  with final error 0.0211). Similarly, decreasing tolerance from  $10^{-4}$  to  $10^{-8}$  gives 1809 total **ODE45** steps ( for  $N_T = 713$  with final error  $4.8 \times 10^{-6}$ ).

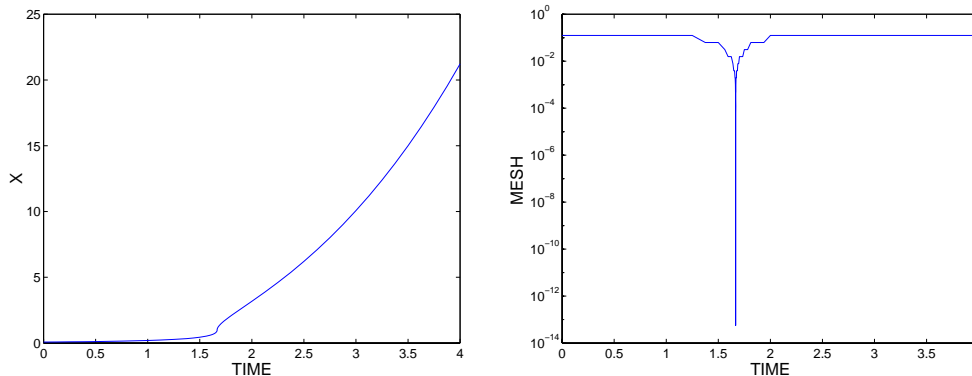


Figure 3: (Example 4.2) Approximate solution (left) and mesh function (right) of **MSTZ** using a 5-th order explicit Runge-Kutta method with  $\text{TOL} = 10^{-4}$  and  $\omega = 5/3$ .

**Remark 4.3** If a time node, say  $t_{m+1}$ , hits the singularity at  $t = \omega$ , the approximation  $\bar{X}(t_{m+1})$  becomes an infinite number. One remedy for this is to change the time steps, i.e  $t_m := t_m + \alpha$  and  $t_{m+1} := t_{m+1} + \beta$  where  $\alpha$  and  $\beta$  are sufficiently small numbers, e.g.  $\Delta t_m/M$ , and then recompute  $\bar{X}(t_m)$  and  $\bar{X}(t_{m+1})$ . Using this practical trick, we solve Example 4.2 for  $w = 1$  and  $T = 4$  and we get  $|g(X(T)) - g(\bar{X}(T))| = 1.7598 \times 10^{-4}$  with 81 final time steps and 1728 total time steps using a 5-th order explicit Runge-Kutta method and  $\text{TOL} = 10^{-3}$ ,  $N[1] = 40$ .

## References

- [1] V.M. Alekseev, An estimate for the perturbations of the solutions of ordinary differential equations. II, *Vestnik Moskov. Univ. Ser. I Mat. Mech.*, **3** (1961), 3-10, russian.
- [2] M. Ainsworth and J. T. Oden, A posteriori error estimation in finite element analysis, *Comput. Methods Appl. Mech. Engrg.*, **142** (1997), 1-88.
- [3] I. Babuška, A. Miller and M. Vogelius, Adaptive methods and error estimation for elliptic problems of structural mechanics, in *Adaptive computational methods for partial differential equations* (College Park, Md., 1983), SIAM, Philadelphia, Pa., (1983), 57-73.
- [4] I. Babuška, and M. Vogelius, Feedback and adaptive finite element solution of one-dimensional boundary value problems, *Numer. Math.* **44** (1984), no. 1, 75-102.
- [5] N.S. Bakhvalov, On the optimality of linear methods for operator approximation in convex classes of functions, *USSR Comput. Math. and Math. Phys.*, **11** (1971), 244-249.
- [6] K. Becker and R. Rannacher, A feed-back approach to error control in finite element methods: basic analysis and examples, *East-West J. Numer. Math.*, **4** (1996), no. 4, 237-264.
- [7] K. Böttcher and R. Rannacher, Adaptive error control in solving ordinary differential equations by discontinuous Galerkin method, preprint, (1996).
- [8] A. Cohen, W. Dahmen and R. DeVore, Adaptive wavelet methods for elliptic operator equations: convergence rates, *Math. Comp.*, **70** (2001), no. 233, 25-75
- [9] G. Dahlquist and Å. Björk, *Numerical Methods*, Prentice-Hall, 1974.
- [10] G. Dahlquist and Å. Björk, *Numerical Mathematics*, <http://www.mai.liu.se/~akbjo/NMbook.html>
- [11] R. A. DeVore, Nonlinear approximation, *Acta Numerica*, (1998), 51-150.
- [12] J.R. Dormand and P.J. Prince, A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, **6** (1980), no. 1, 19-26.
- [13] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, *SIAM J. Numer. Anal.* **33** (1996), no. 3, 1106-1124.
- [14] J.R. Dormand and P.J. Prince, A family of embedded Runge-Kutta formulae, *J. Comp. Appl. Math.* **6** (1980), no. 1, 19-26.
- [15] K. Eriksson, D. Estep, P. Hansbo and C. Johnson, Introduction to adaptive methods for differential equations, *Acta Numerica*, (1995), 105-158.
- [16] D. Estep, A posteriori error bounds and global error control for approximation of ordinary differential equations, *SIAM J. Numer. Anal.*, **32** (1995), 1-48.
- [17] D. Estep and C. Johnson, The pointwise computability of the Lorenz system, *Math. Models Methods Appl. Sci.*, **8** (1998), 1277-1305.

- [18] F. Gao, Probabilistic analysis of numerical integration algorithms, *J. Complexity* **7** (1991), no. 1, 58-69.
- [19] W. Gröbner, *Die Lie-Reihen und ihre Anwendungen*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1967.
- [20] E. Harrier, S.P. Norsett and G. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, 1993.
- [21] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley & Sons, Inc., 1962.
- [22] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, 1996.
- [23] 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), 908-926.
- [24] C. Johnson and A. Szepessy, Adaptive finite element methods for conservation laws based on a posteriori error estimates, *Comm. Pure Appl. Math.*, **48** (1995), 199-234.
- [25] H. Lamba and A. M. Stuart, Convergence results for the MATLAB ODE23 routine, *BIT*, **38** (1998), No. 4, 751-780.
- [26] A. Logg, A multi-adaptive ODE solver, *preprint*,  
<http://www.md.chalmers.se/Centres/Phi/preprints/index.html>
- [27] E. N. Lorenz, Deterministic non-periodic flows, *J. Atmos. Sci.*, **20** (1963), 130-141.
- [28] MATLAB Help Desk, MATLAB, 1999.
- [29] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, Adaptive approximation of partial differential equations based on global and local errors, *preprint*,  
<http://www.nada.kth.se/~szepessy/pde.ps>
- [30] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, Hyperbolic differential equations and adaptive numerics, in *Theory and numerics of differential equations, Durham 2000* (Eds. J.F. Blowey, J.P. Coleman and A.W. Craig), Springer Verlag, in press.
- [31] E. Novak, On the power of adaption, *J. Complexity*, **12** (1996), 199-237.
- [32] A. Szepessy, R. Tempone and G. E. Zouraris, Adaptive weak approximation of stochastic differential equations, TRITA-NA-9912, NADA, KTH, Sweden, (1999).  
<http://www.nada.kth.se/~szepessy/sdew.ps>
- [33] G. Söderlind, Automatic control and adaptive time-stepping ANODE01 Proceedings, Numerical Algorithms.
- [34] J.F. Traub and A.G. Werschulz, *Complexity and Information*, Cambridge University Press, Cambridge, 1998.

- [35] T. Utumi, R. Takaki and T. Kawai, Optimal time step control for the numerical solution of ordinary differential equations, *SIAM J. Numer. Anal.*, **33** (1996), 1644-1653.
- [36] A. G. Werschulz, *The Computational Complexity of Differential and Integral Equations, An Information-Based Approach*, Oxford Mathematical Monographs. Oxford Science Publications. *The Clarendon Press, Oxford University Press, New York*, 1991.