Title

A posteriori error estimation and global error control for ordinary differential equations by the adjoint method

Permalink

https://escholarship.org/uc/item/3c59x9vs

Journal

SIAM Journal on Scientific Computing, 26(2)

ISSN

1064-8275

Authors

Cao, Y
Petzold, L

Publication Date

2004

SIAM J. Sci. Comput.
Vol. 26, No. 2, pp. 359–374

A POSTERIORI ERROR ESTIMATION AND GLOBAL ERROR CONTROL FOR ORDINARY DIFFERENTIAL EQUATIONS BY THE ADJOINT METHOD[*]

YANG CAO[†] AND LINDA PETZOLD[†]

Abstract. In this paper we propose a general method for a posteriori error estimation in the solution of initial value problems in ordinary differential equations (ODEs). With the help of adjoint sensitivity software, this method can be implemented efficiently. It provides a condition estimate for the ODE system. We also propose an algorithm for global error control, based on the condition of the system and the perturbation due to the numerical approximation.

Key words. adjoint method, ordinary differential equation, global error control

AMS subject classifications. 65F35, 65L05, 65L50, 65L60

DOI. 10.1137/S1064827503420969

1. Introduction.

Error is one of the major concerns in numerical analysis. There are many sources of errors, including roundoff error, truncation error, error in the data, and uncertainty in the model. Most of these errors cannot be determined a priori. An important task for numerical analysis is to build up reliable and efficient algorithms. It is desirable to be able to estimate or control the error in the final solution within a tolerance when we solve a problem. An error estimate can help us to determine whether or not we can trust the solution or even the model itself. Particularly for ODEs, the error estimate is also very important for adaptive error control. In this paper, we propose a general method for a posteriori error estimation in the solution of the ODE initial value problem

$$
\left\{\begin{array}{l} \dot{x} = f(x,t), \quad 0 < t \leq T, \\ x(0) = x_0, \end{array}\right. \tag{1}
$$

where $x \in \mathbb{R}^n$. This method is based on a condition analysis of system (1) and a perturbation analysis of the particular numerical discretization method. With this error estimation method, we also propose a global error control algorithm which can be combined with existing local error control schemes.

Error estimation and error control have been important topics from the very first numerical analysis of ODEs (see [9, 16, 20, 29]). Algorithms for the estimation and control of local error have been studied quite deeply and implemented in ODE solvers like DASSL and DASPK [4], VODE [5], DIFSUB [14], and RADAU5 [17]. Many methods for estimating global errors have also been proposed; see, for example, [29, 31, 32]. A good review of basic methods for estimation of the global error can be found in [31]. Most of these estimates are based on a priori methods and asymptotic analysis in the limit of small stepsize. A simple method of this type for computing the global error is to compute the solution with two different formulas or stepsizes. The difference serves as the asymptotic estimate of the global error. However, this type of estimate is not ideal because it is neither rigorous nor efficient.

[*]Received by the editors March 5, 2003; accepted for publication (in revised form) March 18, 2004; published electronically December 22, 2004. This work was supported by grants NSF/ITR ACI-0086061 and DOE DE-F603-00ER25430.

http://www.siam.org/journals/sisc/26-2/42096.html

[†]Department of Computer Science, University of California, Santa Barbara, Santa Barbara, CA 93106 (ycao@cs.ucsb.edu, petzold@engineering.ucsb.edu).

An error *bound* similar to what is obtained in linear algebra using the matrix condition number would be desirable if the additional computations were not too expensive. Recently, much work has been done towards this end for ODEs and partial differential equations (PDEs) (see [9, 10, 11, 12, 13, 15, 21, 26, 27]). An extensive review is given in [12] of the theoretical framework of a posteriori error estimation, which includes a discussion of implementation, particularly for reaction-diffusion equations. The analysis is based on finite elements and formulating the error through the dual (adjoint) problem. Goal-oriented a posteriori error estimation for a single derived function has been proposed for error estimation and adaptive error control for PDEs (see [1, 2, 3, 15, 19]) and ODEs (see [10, 21, 26, 27]). Implementation details for adaptive stepsize control for ODEs can also be found in [11, 12, 13, 25]. This method has been very successful for a single derived function (the "goal"). For the general case, the "choice of data" (initial condition) for the dual problem [12] has remained an open problem. A brief discussion on this issue has been given in [12, section 4.2]. One approach is to use random initial data. Another approach is to try to compute an approximation to the direction of the error and use that as initial data for the dual problem. It is reported in [12] that *"we experimented with this approach using several different approximations based on heuristic reasoning and in almost every case obtained worse results than those obtained using random initial data for the dual problem."* This is an interesting observation. It suggests that probability theory could play a role here.

There are two main contributions of this paper. The first is to show how a posteriori error estimates can be relatively easily implemented within the framework of existing ODE solvers with the help of adjoint sensitivity software. We derive the error estimate using the adjoint system for the sensitivity analysis of ODEs [6, 7]. This adjoint system is equivalent to the dual system in [12]. We consider the numerical discretization error as a perturbation to the ODE system. The error in the solution has two parts: the error at each time step due to the discretization and the propagation of these errors which is measured by the condition. We begin by taking the numerical solution as the solution to a perturbed ODE system. According to [28], the condition of the ODE system is the sensitivity of the solution with respect to these perturbations. Thus the condition problem becomes a sensitivity problem. With the help of adjoint sensitivity analysis, we solve the sensitivity problem and estimate the condition of the system. We present the corresponding perturbation analysis for the backward differentiation formula (BDF) method and obtain the error estimate. This approach is not limited to BDF methods.

The error estimate using the analysis of the numerical solution as the solution to a perturbed ODE system was first proposed in [14] and has been further developed in [18]. But at that time it was not deemed to be practical. The difficulty is in the estimation of the condition. The perturbations can arise at any time during the solution of the system and thus has an infinite dimension. Therefore the sensitivity will have an infinite dimension and is difficult, if not impossible, to compute. Thus we must *estimate* the sensitivity. We consider the condition problem as a sensitivity problem. Since there are a large number of parameters (perturbations), the adjoint method is an attractive choice. A detailed analysis of the adjoint sensitivity method for differential-algebraic equations (DAEs) can be found in [6, 7]. The advantage of the adjoint method is that when the number of parameters increases, the computational cost does not increase much once we have the adjoint solution. Furthermore, the adjoint problems are linear and can be computed in parallel.

The second main contribution of this paper is to present a method to give a probabilistic error estimate for the solution or for a multidimensional derived function. The adjoint sensitivity method applies best to the case of a small-dimensional output space and a large-dimensional parameter space. But the error estimation problem may have both a large-dimensional output space and a large-dimensional parameter space. To resolve this, instead of requiring the error estimate to be accurate, we require it to be accurate *with a high probability.* To accomplish this, we use the small sample statistical method which was originally proposed in [22, 23] and was later applied to condition estimation in numerical linear algebra [8]. The small sample statistical method provides us with a methodology for choosing the initial condition for the adjoint system which is applicable to general ODE systems and is justified by theory. It confirms as well as justifies Estep's observation regarding the use of random initial data.

Finally, we combine local error control with the condition estimate to obtain a method for global error control. The same approach can also be applied to obtain a goal-oriented (subspace) error estimate. This method is not based on any particular class of discretization formula. It can be applied easily to all the known one-step and multistep methods, such as BDF, Adams, and Runge–Kutta methods. The overall strategy for global error control is similar to the approaches in [12, 26, 27, 33]. In this paper we focus mainly on the BDF method, where adjoint sensitivity software [6, 7] is readily available.

We note that it may be possible to use an approach similar to the one outlined here, based on adjoint sensitivity analysis in combination with the small sample statistical method, for global error estimation and control for some time-dependent PDE systems. However, the computation of the sensitivities presents many challenges [24].

In summary, our new method is the combination of two powerful methods: the adjoint sensitivity method and software for ODEs and DAEs [6, 7], and the small sample statistical method for estimation of condition [22, 23]. This paper is organized as follows. In section 2 the condition number and error estimation method is proposed. An asymptotic analysis of the perturbation due to the numerical approximation is given in section 3 for BDF methods. Section 4 describes our strategy for global error control. Numerical experiments are presented in section 5.

2. A posteriori error estimation for ODEs.

Before proceeding, we establish some notation. In the remainder of this paper we will be using the following norms. Unless specified otherwise, for the vector norm we will use the 2-norm. Thus we will omit the subscript,

$$\|v\| = \|v\|_2 = \left(\sum v_i^2\right)^{\frac{1}{2}}.$$

For the norm of a function $f \in C([0,T])$, the L_1-norm, L_2-norm, and ∞-norm will all be used:

$$\|f\|_{L_1} = \int_0^T \|f(t)\| dt,$$

$$\|f\|_{L_2} = \left(\int_0^T \|f(t)\|^2 dt\right)^{\frac{1}{2}},$$

$$\|f\|_\infty = \max_{0\le t\le T} \|f(t)\|.$$

The error in the solution comes from two sources: the error propagated by the ODE system and the error introduced by the numerical discretization at each time step. We will begin by focusing on the propagation of errors by the ODE system. Consider the time-varying linear ODE

$$\text{(2)}\qquad \begin{cases} \quad \dot{x} = A(t)x + h(t), \\ x(0) = x_0 \end{cases}$$

and the corresponding perturbed ODE

$$\text{(3)}\qquad \begin{cases} \quad \dot{\tilde{x}} = A(t)\tilde{x} + h(t) + r_1(t), \\ \tilde{x}(0) = x_0 + r_2, \end{cases}$$

where $r_1(t)$ and r_2 are the perturbations. The perturbations may arise from roundoff error, truncation error, error due to termination of the Newton iteration if using an implicit method, or uncertainty in the model. We will leave the discussion on the estimation of r_1 to section 3.

The main objective of our a posteriori error estimate is to estimate the error $\|x(T)-\tilde{x}(T)\|$ at the end point T. Without any assumptions about the perturbations, this estimate is nonsense. Different applications give rise to different assumptions. For the purpose of estimating the global error in the numerical solution of ODEs, we assume that the perturbations are bounded as follows:

$$\|r_1\|_\infty < \epsilon, \quad \|r_2\| < \epsilon.$$

2.1. Error estimation for a scalar derived function. We first consider estimating the error of a scalar derived function $g = g(x(T))$ at the end time T. Let $e = x - \tilde{x}$, and define $\delta = \|e(T)\|$. Then we have

$$\Delta g = g(x(T)) - g(\tilde{x}(T)) = g_x(x(T))(e(T)) + O(\delta^2).$$

The error e satisfies

$$\text{(4)}\qquad \begin{cases} \quad \dot{e} = A(t)e + r_1(t), \\ x(0) = r_2. \end{cases}$$

Thus

$$e(t) = \int_0^t \Phi(t)\Phi^{-1}(s)r_1(s)ds + \Phi(T)r_2,$$

where Φ is the fundamental solution matrix, which satisfies

$$\begin{aligned} \dot{\Phi} &= A(t)\Phi, \\ \Phi(0) &= I. \end{aligned}$$

Let $l = g_x^T(x(T))$. Then the error in the derived function g satisfies

$$\begin{aligned} \Delta g &= l^T e(T) + O(\delta^2) \\ &= \int_0^T l^T \Phi(T)\Phi^{-1}(s)r_1(s)ds + l^T\Phi(T)r_2 + O(\delta^2). \end{aligned}$$

Solving the adjoint equation

$$\text{(5)}\qquad \begin{cases} \quad \dot{\lambda} = -A^T(t)\lambda, \\ \lambda(T) = l, \end{cases}$$

we obtain

$$\lambda^T(s) = l^T \Phi(T)\Phi^{-1}(s)$$

and

$$\lambda^T(0) = l^T \Phi(T).$$

Thus

$$\Delta g = \int_0^T \lambda^T(s) r_1(s) ds + \lambda^T(0) r_2 + O(\delta^2) \tag{6}$$

and

$$\begin{aligned} |\Delta g| &\leq \|\lambda\|_{L_1} \|r_1\|_\infty + \|\lambda(0)\| \|r_2\| + O(\delta^2) \\ &< (\|\lambda\|_{L_1} + \|\lambda(0)\|)\epsilon + O(\delta^2). \end{aligned} \tag{7}$$

We note that (7) is a rather loose estimate because it takes the norm and ignores the effects of cancellation of errors. Equation (6) is more general and will be the formula we use in global error control. Formula (7) provides a simple formula in the form of *error = condition number ∗ perturbation to data*. The term $(\|\lambda\|_{L_1} + \|\lambda(0)\|)$ plays a role which is similar to the role that the matrix condition number plays for linear systems. $\|\lambda(0)\|$ is the condition number with respect to perturbations in the initial values, and $\|\lambda\|_{L_1}$ is the condition number with respect to perturbations of the differential equation. We define

$$K(\lambda) = \|\lambda(t)\|_{L_1} + \|\lambda(0)\|. \tag{8}$$

Ignoring the high-order term, we have $\|\Delta g\| < K\epsilon$. Further details on this form of the condition number for ODEs can be found in [26]. Note that λ is a function determined by the differential equation and the derived function. It can be computed by adjoint sensitivity software. Details on the computation of λ can be found in [6, 7, 26, 27].

2.2. Error estimation for the solution. Our objective is to estimate $\|\Delta x(T)\|$. $x(T)$ is a vector, whereas in the last subsection g was a scalar function. A direct generalization of the method in section 2.1 would be to define every component of X as a scalar derived function. But in that way we would have to solve a large adjoint system. The computational cost would make this method impractical. Alternatively, if we know $\Delta x(T)$, we may define $g(x) = [\frac{\Delta x(T)}{\|\Delta x(T)\|}]^T x$. Then $\|\Delta g(x(T))\| = \|\Delta x(T)\|$. Unfortunately, however, we do not have $\Delta x(T)$. Thus we cannot directly apply the method of section 2.1. However, if we allow our estimate to have a moderate relative error, we can use the small sample statistical method in combination with the estimate for scalar functions derived in the previous subsection.

The small sample statistical method was originally proposed in [23]. It suits the requirements of error estimation very well. Here we give the basic idea; details can be found in [23].

For any vector $l \in \mathbb{R}^n$, if z is selected uniformly and randomly from the unit sphere S_{n-1} in n dimensions, the expected value of $|l^T z|$ is given by

$$E(|l^T z|) = \|l\| E_n,$$

where $E_1 = 1$, $E_2 = \frac{2}{\pi}$, and for $n > 2$,

$$E_n = \frac{1 \cdot 3 \cdot 5 \cdots (n-2)}{2 \cdot 4 \cdot 6 \cdots (n-1)} \quad \text{for } n \text{ odd},$$

$$E_n = \frac{2}{\pi} \cdot \frac{2 \cdot 4 \cdot 6 \cdots (n-2)}{1 \cdot 3 \cdot 5 \cdots (n-1)} \quad \text{for } n \text{ even}.$$

E_n can be estimated by $\sqrt{\frac{2}{\pi(n-\frac{1}{2})}}$. We use $\xi = \frac{|l^T z|}{E_n}$ to estimate $\|l\|$. The estimate satisfies

$$\Pr\left(\frac{\|l\|}{w} \leq \xi \leq w\|l\|\right) \geq 1 - \frac{2}{\pi w} + O\left(\frac{1}{w^2}\right), \tag{9}$$

where Pr() denotes the probability. The bound does not depend on the vector l. The proof for this result is based on the inner product distributions on S_{n-1}. Taking an orthogonal rotation of l and z, we may select a coordinate system in which l is just a scalar multiple of the vector $e^{(1)} = (1, 0, \ldots, 0)^T$. This means that $l^T z = \|l\||z_1|$, where $z = (z_1, \ldots, z_n)^T$. Inequality (9) is then obtained by the analysis of the distribution of $|z_1|$ when z is uniformly and randomly distributed over the unit sphere S_{n-1} in $\mathbb{R}^n$.

For a more accurate estimate, we can use more orthogonal random vectors. Suppose we have k orthogonal random vectors $z_1, z_2, \ldots, z_k$. Let

$$\eta_i = |l^T z_i|.$$

Then the estimate for $\|l\|$ is given by

$$\xi(k) = \frac{E_k}{E_n}\sqrt{\eta_1^2 + \cdots + \eta_k^2}. \tag{10}$$

Usually at most two or three random vectors are required in practice. The corresponding probabilities satisfy (see [23])

$$\Pr\left(\frac{\|l\|}{w} \leq \xi(2) \leq w\|l\|\right) \approx 1 - \frac{\pi}{4w^2},$$

$$\Pr\left(\frac{\|l\|}{w} \leq \xi(3) \leq w\|l\|\right) \approx 1 - \frac{32}{3\pi^2 w^3}.$$

Table 1 gives the lower bounds on the probability that $\xi(k)$ equals $\|l\|$ to within a factor of w. We comment that usually for the estimation of the error, only the magnitude is of concern. For example, with $w = 10$, two random vectors can achieve 99% probability of accuracy to a factor of 10.

To estimate $\|e(T)\|$, we first select a random vector z uniformly from the unit sphere S_{n-1}. Then define $g(x) = z^T x$. Solving for λ_z from

$$\begin{cases} \dot{\lambda}_z = -A^T(t)\lambda_z, \\ \lambda_z(T) = z, \end{cases} \tag{11}$$

the condition estimate is given by $\frac{1}{E_n}K(\lambda_z)$. For the error estimate, we have

$$\|e(T)\| \approx \frac{1}{E_n}|z^T e(T)| \leq \frac{1}{E_n}K(\lambda_z)\epsilon, \tag{12}$$

Table 1
Lower bounds on the probability that $\xi(k)$ equals $\|l\|$ to within a factor of w.

w	Lower bound for $\xi(1)$	Lower bound for $\xi(2)$	Lower bound for $\xi(3)$	Lower bound for $\xi(4)$
3	0.7736	0.9156	0.9632	0.9831
5	0.8732	0.9691	0.9916	0.9976
10	0.9364	0.9922	0.9989	0.9998
100	0.9936	0.9999	$1-10^{-5}$	$1-10^{-7}$

where $\approx$ means a high probability for an accuracy to a factor of 10. For two orthogonal random vectors z_1 and z_2, we solve the adjoint equation (11) twice to obtain λ_{z_1} and λ_{z_2}. Then the condition number is given by $\frac{E_2}{E_n}(K(\lambda_{z_1})^2+K(\lambda_{z_2})^2)^{\frac{1}{2}}$. Two orthogonal random vectors yield an error estimate which is correct to within a factor of 10 with 99% probability.

Note that so far we have discussed error estimation only for linear ODE systems. In this case, the coefficients of the adjoint system do not depend on the solution of the original system. Thus we can solve the adjoint system separately. This is not true for nonlinear ODE systems.

2.3. A posteriori error estimation for nonlinear ODEs. When the system is nonlinear, the adjoint ODE depends on the solution to the original ODE system. This requires numerically solving the ODE and storing an approximation to the solution so that it can be used later to determine the adjoint ODE. The reader is referred to [7, 26] for details on how this can be accomplished efficiently. We note that if the original ODE is not solved to sufficient accuracy, the adjoint solution and hence the error estimate cannot be trusted. Hence the entire process may have to be iterated several times for the bound to be absolutely trustworthy. Then it becomes an adaptive algorithm [26, 27]. The additional storage is a drawback of this method.

Consider the ODE system

$$\left\{\begin{aligned} \dot{x} &= f(x,t),\\ x(0) &= x_0 \end{aligned}\right. \tag{13}$$

and the perturbed system

$$\left\{\begin{aligned} \dot{\tilde{x}} &= f(\tilde{x},t)+r_1(t),\\ x(0) &= x_0+r_2, \end{aligned}\right. \tag{14}$$

where $\|r_1\|_\infty < \epsilon$ and $\|r_2\| < \epsilon$. As before, we define $e = x - \tilde{x}$. Then

$$\dot{e} = f(x,t) - f(\tilde{x},t) + r_1(t) = J(\tilde{x},t)e + r_3(x,\tilde{x},t) + r_1(t) \tag{15}$$

with initial condition $e(0) = r_2$, where $J(\tilde{x},t)$ is the Jacobian of f at $\tilde{x}$, and $r_3(x,\tilde{x},t) = f(x,t) - f(\tilde{x},t) - J(\tilde{x},t)(x-\tilde{x})$. Assuming $\tilde{x}(t)$ is close to $x(t)$, we have $r_3(x,\tilde{x},t) = O(\|x-\tilde{x}\|^2)$. Here we will assume that the stepsizes have been chosen so that the numerical solution is close to the true solution. Then $\|r_3(x,\tilde{x},t)\|_\infty < \epsilon$. The algorithm of section 2.2 is then applied in the nonlinear case as follows.

Let the matrix function $\Phi(t)$ satisfy

$$\left\{\begin{aligned} \dot{\Phi} &= J(\tilde{x},t)\Phi,\\ \Phi(0) &= I. \end{aligned}\right. \tag{16}$$

Then $e(T) = \int_0^T \Phi(T)\Phi^{-1}(s)(r_3(x,\tilde{x},s) + r_1(t))ds + \Phi(T)r_2$. Thus

$$l^T e(T) = \int_0^T l^T \Phi(T)\Phi^{-1}(s)(r_3(x,\tilde{x},s) + r_1(t))ds + l^T \Phi(T)r_2. \tag{17}$$

The adjoint ODE system is given by

$$\begin{cases} \dot{\lambda} = -J(\tilde{x},t)^T \lambda, \\ \lambda(T) = l. \end{cases} \tag{18}$$

The adjoint solution satisfies $\lambda^T(s) = l^T\Phi(T)\Phi^{-1}(s)$ and $\lambda^T(0) = l^T\Phi(T)$. Thus

$$l^T e(T) = \int_0^T \lambda^T(s)(r_3(x,\tilde{x},s) + r_1(s))ds + \lambda^T(0)r_2,$$

yielding $|l^T e(T)| \le 2K(\lambda)\epsilon$. To estimate $\|e(T)\|$, we use the small sample statistical method as outlined in the previous subsection.

Remark 2.1. The key assumption here is that $\|r_3(x,\tilde{x},t)\|_\infty < \epsilon$. Such an assumption is necessary. For example, if the perturbed system were unrelated to the original system, then there would be no way to give a reasonable estimate.

Remark 2.2. As pointed out in [12], an important problem for error estimation is to choose a proper initial value for the adjoint system. The discussion in [12] leads to a probabilistic lower bound for the estimate. Our method generates probabilistic upper and lower bounds. The quality of the error estimation can be improved by using more random vectors. But the analysis in section 2.2 shows that it is usually sufficient to use only two or three random vectors. We should point out that for some special problems, such as reaction-diffusion equations, the error tends to point in the direction of the lowest few modes. Utilizing that information can lead to a better error estimation as pointed out in [12].

3. The perturbation due to the numerical approximation.

The key assumption in our estimated bound on the global error from section 2 was that $\|r_1\|_\infty$, $\|r_2\|$, and $\|r_3\|_\infty$ are bounded by ϵ.[1] Recall that r_2 is the error of the initial value, and r_3 is bounded by $O(\|\tilde{x} - x\|^2)$. These perturbations can be reasonably expected to be small. Thus our focus will be on the estimate of $\|r_1\|_\infty$. $r_1(t)$ was called "the perturbation due to the numerical approximation" by C. W. Gear in his famous book [14, sections 1.3.5 and 4.5.1]. In [14], Gear analyzed the situation of explicit Runge–Kutta methods. His analysis can be easily generalized to implicit Runge–Kutta methods. Here we focus our discussion on the analysis of multistep methods. We give an analysis for the variable-coefficient form of BDF methods in the following. The bound for other multistep methods can be similarly derived.

THEOREM 3.1. *Suppose the function f is sufficiently smooth and satisfies the Lipschitz condition $\|f(x) - f(y)\| \le L\|x - y\|$.*[2] *Then the perturbation due to the k-step BDF method is bounded by $\frac{R}{h_{min}} + O(h^k)$, where h_{min} and h both denote the stepsize in the case of the fixed stepsize method, or $h_{min} = \min_i h_i$, $h = \max_i h_i$ in the case of a variable stepsize method, and R is a bound on the roundoff error and the error due to the Newton iteration.*

[1] We note that a similar analysis is given in [10, 12] for the discontinuous Galerkin method, where the errors are considered as the residual errors.

[2] This assumption is very tight. However, for the purpose of keeping the exposition simple, we will use it here.

Proof. The numerical solution is given by constructing the polynomial which interpolates x_{n-i} at the last k values and satisfies the ODE at t_n,

$$\begin{aligned} p(t_{n-i}) &= x_{n-i}, \quad 1 \le i \le k, \\ p'(t_n) &= f(p(t_n), t_n), \end{aligned} \tag{19}$$

and setting

$$x_n = p(t_n) + \delta_n,$$

where $\|\delta_n\| \le R$. Define a continuous form of the perturbed solution in $[t_{n-1}, t_n]$ by $z(t) = p(t) + \frac{t-t_{n-1}}{t_n - t_{n-1}}\delta_n$. Then $z(t)$ is differentiable and agrees with the numerical solution at the mesh points $z(t_n) = x_n$. We now estimate the residual $r(t) = z'(t) - f(z(t), t)$.

Define the local solution which satisfies

$$u'(t) = f(u(t), t), \quad u(t_{n-k}) = x_{n-k}, \quad t \in [t_{n-k}, t_n]. \tag{20}$$

From the classical results we have $x_{n-i} - u(t_{n-i}) = O(h^{k+1})$ for $i = 0, \ldots, k$. Consider two kth-order polynomials which satisfy

$$\begin{aligned} q(t_{n-i}) &= x_{n-i}, \quad 1 \le i \le k, \\ q(t_n) &= x_n \end{aligned}$$

and

$$\begin{aligned} s(t_{n-i}) &= u(t_{n-i}), \quad 1 \le i \le k, \\ s(t_n) &= u(t_n). \end{aligned}$$

It is easy to see that $q(t) = p(t)$ since they coincide at $k+1$ points. The polynomials $q(t)$ and $s(t)$ are interpolated at the same mesh points but with different values which satisfy $x_{n-i} - u(t_{n-i}) = O(h^{k+1})$. Using the Lagrangian form, we have

$$q(t) - s(t) = O(h^{k+1}), \quad q'(t) - s'(t) = O(h^k). \tag{21}$$

For the residual $r(t)$, we have

$$\begin{aligned} r(t) &= z'(t) - f(z(t), t) - (u'(t) - f(u(t), t)) \\ &= z'(t) - f(z(t), t) - (s'(t) - f(s(t), t)) \\ &\quad + [(s'(t) - f(s(t), t)) - (u'(t) - f(u(t), t))]. \end{aligned}$$

Define $d_1(t) = p(t) - s(t)$ and $d_2(t) = s(t) - u(t)$. Then $r(t)$ is given by

$$r(t) = d_1'(t) + d_2'(t) - (f(z(t), t) - f(s(t), t)) + (f(s(t), t) - f(u(t), t)) + \frac{\delta_n}{h_n}.$$

By the Lipschitz condition, $\|f(z(t), t) - f(s(t), t)\| \le L\|d_1(t)\|$, $\|f(s(t), t) - f(u(t), t)\| \le L\|d_2(t)\|$. Thus we need to estimate $d_1(t)$, $d_1'(t)$, $d_2(t)$, and $d_2'(t)$. By (21), we have $d_1(t) = O(h^{k+1})$ and $d_1'(t) = O(h^k)$. For $d_2(t)$ and $d_2'(t)$, using the divided difference notation, $u[t_l] = u(t_l)$,

$$u[t_l, \ldots, t_{l+i}] = \frac{u[t_{l+1}, \ldots, t_{l+i}] - u[t_l, \ldots, t_{l+i-1}]}{t_{l+i} - t_l},$$

and Newton's form of the interpolating polynomial,

$$s(t) = u[t_n] + u[t_n, t_{n-1}](t - t_n) + \cdots + u[t_n, \ldots, t_{n-k}] \prod_{i=0}^{k} (t - t_{n-i}),$$

the interpolation error is given by

$$s(t) - u(t) = u[t, t_n, \ldots, t_{n-k}] \prod_{i=0}^{k} (t - t_{n-i}).$$

It is easy to show that $x[t, t_n, \ldots, t_{n-k}] \leq C \max_{\tau \in [t_{n-k}, t_n]} \|x^{(k+1)}(\tau)\|$ and $\frac{d(x[t,t_n,\ldots,t_{n-k}])}{dt} = C \max_{\tau \in [t_{n-k}, t_n]} \|x^{(k+2)}(\tau)\|$, where C is a constant and $x^{(i)}(t) = \frac{d^i x(t)}{dt^i}$. Thus we have

$$\|d_2(t)\| = O(h^{k+1}), \quad \|d_2'(t)\| = O(h^k). \tag{22}$$

The magnitude of the residual $r(t) = z'(t) - f(z(t), t)$ is then estimated by

$$\begin{aligned} \|r(t)\| &\leq \|d_1'(t)\| + \|d_2'(t)\| + L(\|d_1(t)\| + \|d_2(t)\|) + R/h_{min} \\ &= R/h_{min} + O(h^k). \qquad \square \end{aligned}$$

Remark 3.1. Assuming that in each step the errors due to roundoff and the Newton iteration are small, we can neglect R. The global error is then given by

$$\|e(T)\| = K(\lambda(t)) \|r_1(t)\|_\infty \leq CK(\lambda(t))h^k. \tag{23}$$

This is another formula for the famous result *convergence = stability + consistency* and has been given in [12] for the discontinuous Galerkin method. Traditionally, the *stability* constant is formulated in terms of e^{LT}, where L is the Lipschitz constant for f, and is well known to be too pessimistic. $K(\lambda(t))$ gives a more realistic estimate of the condition of the system and thus makes global error control practically possible.

Remark 3.2. The leading term in $O(h^k)$ is an important concern for error control. We need a computable number instead of the big O. From the above proof, we can see that the leading term arises from $d_1'(t)$ and $d_2'(t)$. For $d_1'(t)$, the leading term arises from the difference of $p(t)$ and $s(t)$ at the mesh points. For a constant stepsize method, the classical local error estimate yields $d_1(t_{n-i}) = [I - h\beta_0 J]^{-1} C_{k+1} x^{(k+1)} h^{k+1}$, where β_0 is the coefficient of f_n in the BDF method. In most practical implementations, $[I - h\beta_0 J]^{-1}$ is neglected, and $x^{(k+1)}$ is estimated during the computation. Let $C_0 = C_{k+1} x^{(k+1)}$. Classical local error control chooses h so that $C_0 h^{k+1} < TOL$. We can use C_0 to construct a leading term estimate for $d_1'(t)$. Rewriting the Lagrangian form from which (21) was derived,

$$d_1(t) = \sum_i \prod_{j \neq i} \frac{t - t_{n-j}}{t_{n-i} - t_{n-j}} d_1(t_{n-i}),$$

where $d_1(t_{n-i})$ is estimated by $C_0 h^{k+1}$. Thus the leading term of $d_1'(t)$ is estimated by

$$d_1'(t) \approx C_0 \sum_i \sum_{j \neq i} \prod_{m \neq i,j} \frac{t - t_{n-m}}{t_{n-i} - t_{n-m}} \frac{h^{k+1}}{t_{n-i} - t_{n-j}}. \tag{24}$$

For $d_2'(t)$, the leading term arises from the interpolation error. Using the Newton form of the interpolation polynomial, we can write $d_2'(t)$ as

$$
\begin{aligned}
d_2'(t) &= u[t, t_n, \dots, t_{n-k}] \frac{d}{dt} \prod_{i=0}^{k} (t - t_{n-i}) + O(h^{k+1}) \\
&= u[t, t_n, \dots, t_{n-k}] \sum_i \prod_{j \neq i} (t - t_{n-i}) + O(h^{k+1}),
\end{aligned}
\tag{25}
$$

where $u[t, t_n, \dots, t_{n-k}]$ is estimated by $\frac{x^{(k+1)}(\tau)}{(k+1)!}$. Thus the leading term of $d_2'(t)$ can be estimated by

$$
d_2'(t) \approx \sum_{i=0}^{k} \frac{1}{i+1} x^{(k+1)}(t) h^k,
\tag{26}
$$

where $x^{(k+1)}(t)$ is estimated during the computation. In the next section, we denote the coefficient of the leading term by C.

4. Control of the global error.

Global error control is expensive, but sometimes it is necessary. Traditionally, ODE solvers such as DASSL and DASPK [4], VODE [5], DIFSUB [14], and RADAU5 [17] have controlled the local error. Because of the expense, few codes have made an attempt to estimate or control the global error. From the analysis in sections 2 and 3, we know that the condition number tells us the stability of the ODE system. If the condition number is moderate, we will end up with a small global error if we control the local error well. But if the condition number is large, we could still have a large global error even if we control the local error within a reasonable tolerance. A natural way to deal with this is to tighten the local error tolerance to obtain a better solution. But we have no idea what the new local error tolerance should be. The condition number provides a way to estimate that.

Once we have obtained the adjoint solution $\lambda(t)$, we can control the global error by varying the local error tolerance. The simplest approach is to take $K = K(\lambda(t))$ and use the estimate

$$
GTOL \leq K \cdot LTOL,
\tag{27}
$$

where $GTOL$ is the global error tolerance and $LTOL$ is the tolerance which bounds $r_1(t)$. For BDF methods, according to the previous section, we have the estimate $r_1(t_n) \approx C h_{n+1}^k$. Thus if we are given $GTOL$, we can let $LTOL = \frac{GTOL}{K}$. To control the global error, we choose the stepsize so that $C h_{n+1}^k \leq LTOL$. Thus we choose

$$
h_{n+1} \leq \left(\frac{GTOL}{KC} \right)^{\frac{1}{k}}.
\tag{28}
$$

We should comment on the difference between this global error control and the original local error control. Denote $ltol$ as the tolerance for the local error control. Local error control in a BDF code like DASSL [4] chooses h_{n+1} such that $C_0 h_{n+1}^{k+1} \leq ltol$. Neglecting the difference between C_0 and C, we may conclude that global error control is similar to taking local error control with a local tolerance $ltol = LTOL * h_{n+1}$. This criterion is not totally new. It was called "the criterion of error per unit step" in [30, p. 97]. In the code ODE [30] which uses the Adams method, this criterion was used.

This simple strategy does not use all the information we have already obtained. If $\lambda(t)$ varies with time, it would be more efficient to choose a larger $LTOL$ when $\lambda(t)$ is smaller. It can be seen from the error estimate (6),

$$\Delta g = \int_0^T \lambda(s) r_1(s) ds + \lambda(0) r_2,$$

that when $\lambda(s)$ is smaller, $r_1(s)$ can be larger. In this situation the final result is not very sensitive to the local error at time s; thus we can loosen the error tolerance locally. Supposing that $r_1(t_n+\tau) = Ch_{n+1}^k$,[3] we choose the next stepsize to satisfy

$$\int_0^{h_{n+1}} \lambda(t_n+\tau) r_1(t_n+\tau) d\tau \le \frac{GTOL}{T} h_{n+1}. \tag{29}$$

Taking $\lambda(t_n+\tau) \approx \lambda(t_n)$, we have

$$\lambda(t_n) C h_{n+1}^{k+1} \le \frac{GTOL}{T} h_{n+1}.$$

Thus the stepsize h_{n+1} is determined by

$$h_{n+1} \le \left(\frac{GTOL}{T\lambda(t_n)C} \right)^{\frac{1}{k}}. \tag{30}$$

As we have mentioned, this condition is similar to the local error control with $ltol = \frac{GTOL}{T\lambda(t_n)} h_{n+1}$. If $\lambda(t)$ is a constant, this is the same as the strategy (28).

Here is the global error stepsize control algorithm with two orthogonal random vectors.

Initialization. The user chooses an initial error tolerance $GTOL$ for the integration time interval $[0,T]$. Generate two orthogonal random vectors z_1 and z_2 on S_{n-1}. Define $g_1 = z_1^T x$ and $g_2 = Z_2^T x$.

Step I. Solve the ODE (by DASPK) with a loose error tolerance. Store the solution. (The details of this part of the implementation can be found in the description of the adjoint sensitivity analysis method and software DASPKADJOINT [7].)

Step II. Solve the adjoint equation (5) with initial value $\lambda_1(T) = z_1$ and $\lambda_2(T) = z_2$. Store $\lambda(t) = \frac{E_2}{E_n}\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}$ (at discrete points).

Step III. Solve the ODE with stepsize control (30).

Remark. If the requirement is to estimate and control the error of $g(X(T))$, where g is of dimension $m < n$, when m is very small such as 1 or 2, the small sample statistical method is not necessary. This situation has been discussed in [26, 27]. In this case, one can just let $\lambda(T) = g_x^T$ and use the norm of $\lambda(t)$. When m is also large, in the initialization z is taken from S_{m-1} and the functions g_1 and g_2 are defined as $g_1 = z_1^T g(x)$ and $g_2 = z_2^T g(x)$. In Step II, the initial values for the adjoint equations are taken as $\lambda_1(T) = g_x^T z_1$ and $\lambda_2(T) = g_x^T z_2$. E_n is also replaced by E_m. With these modifications, this method can provide error estimation and control of the global error of a multidimensional derived function g.

[3] This is the "constant stepsize assumption" that is used in variable-stepsize multistep codes (see [4, 5, 29]) when choosing the next stepsize. It is not strictly valid, but it leads to a simple formula for choosing the next stepsize that has worked well for local error control.

Table 2

Comparison of errors and steps for Example 1 *with and without the global error control* (30). $a = 1$, $y(0) = 10^{-4}$, $T = 10$, *condition number* $K = 26217$, $y(T) = 2.2$.

Tol		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Original DASPK	error	2.2	6.7	10	1.35	0.2	0.045
	steps	10	33	44	54	43	73
Global error control	error	0.01	4×10^{-4}	1.6×10^{-4}	2.3×10^{-6}	2×10^{-7}	2×10^{-7}
	steps	49	81	98	146	196	355

Table 3

Comparison of errors and steps for Example 1 *with and without the global error control* (30). $a = -1$, $y(0) = 1$, $T = 1$, *condition number* $K = 0.6$, $y(T) = 0.42$.

Tol		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Original DASPK	error	0.05	0.01	2×10^{-4}	9×10^{-6}	6.3×10^{-6}	8×10^{-7}
	steps	11	12	17	27	34	40
Global error control	error	0.03	4.8×10^{-4}	2×10^{-4}	1.8×10^{-6}	2.5×10^{-7}	3.4×10^{-8}
	steps	12	17	24	32	42	52

Table 4

Comparison of errors and steps for Example 1 *with and without the global error control* (30). $a = -20$, $y(0) = 1$, $T = 1$, *condition number* $K = 0.05$, $y(T) = 2.06 \times 10^{-9}$.

Tol		10^{-9}	10^{-10}	10^{-11}	10^{-12}
Original DASPK	error	6.3×10^{-11}	3.2×10^{-11}	3.7×10^{-12}	5.8×10^{-13}
	steps	288	588	557	776
Global error control	error	1.1×10^{-10}	1.8×10^{-12}	8.3×10^{-13}	7.7×10^{-14}
	steps	134	225	263	416

5. Numerical results. The examples in this section were mostly chosen from [10]. The software used for the experiments was DASPK3.0 [4] and its adjoint sensitivity analysis tool DASPKADJOINT [7]. For each example we computed the condition number, and compared the final error for the original DASPK code with our modified DASPK code after applying global error control (30), for several different error tolerances. Here the tolerance for the original DASPK is the local tolerance, and the tolerance for the modified DASPK code is the global tolerance. In both codes the tolerances are the absolute tolerance.

Example 1 (scalar, linear ODE). Consider the ODE $\dot{y} = ay$. We give the results for $a = 1$ (unstable), $a = -1$ (stable), and $a = -20$ (stiff). For the example when $a = 1$, we took the initial condition to be $y(0) = 10^{-4}$ and the final time $T = 10$. The numerical results are listed in Table 2. For the example when $a = -1$ and $a = -20$, we took the initial condition to be $y(0) = 1$ and the final time $T = 1$. The numerical results are listed in Tables 3 and 4.

Example 2 (nonlinear with changing stability). Consider $\dot{y} + (0.25 + \sin \pi t)y^2 = 0$, $y(0) = 1$, with solution $y(t) = \pi/(\pi + 1 + 0.25\pi t - \cos \pi t)$, and final time $T = 1$. The numerical results are listed in Table 5.

Example 3 (unstable system). Consider

$$\begin{cases} \dot{y}_1 - \dfrac{1}{2(1+t)} y_1 + 2ty_2 = 0, & t > 0, \\ \dot{y}_2 - \dfrac{1}{2(1+t)} y_2 - 2ty_1 = 0, & t > 0, \\ y_1(0) = 1, \quad y_2(0) = 0. \end{cases}$$

Table 5

Comparison of errors and steps for Example 2 *with and without the global error control* (30). $y(0) = 1$, $T = 1$, *condition number* $K = 0.6$, $y(T) = 0.53$.

Tol		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Original	error	0.1	0.012	0.001	1×10^{-4}	6×10^{-5}	3×10^{-7}
DASPK	steps	11	15	22	29	43	56
Global error	error	0.04	4×10^{-4}	2×10^{-4}	3×10^{-6}	3×10^{-6}	2×10^{-7}
control	steps	12	20	29	42	63	81

Table 6

Comparison of errors and steps for Example 4 *with and without the global error control* (30). $T = 10$, *condition number* $K = 19.36$, $y(T) = (2.86, -1.68)$.

Tol		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Original	error(y_1)	2.8	0.02	0.25	8×10^{-3}	1×10^{-3}	5.7×10^{-5}
DASPK	error(y_2)	1.6	0.8	0.09	0.03	0.005	1.3×10^{-3}
	steps	333	294	422	568	683	940
Global	error(y_1)	0.04	3×10^{-3}	3×10^{-4}	9×10^{-6}	3.7×10^{-6}	6×10^{-7}
error	error(y_2)	0.03	5×10^{-3}	1×10^{-3}	1.5×10^{-4}	1.5×10^{-5}	1.5×10^{-6}
control	steps	529	767	1002	1472	2275	3613

Table 7

Comparison of errors and steps for Example 5 *with and without the global error control* (30). $T = 10$, *condition number* $K = 12553$, $y(T) = (2.2, -2.2)$.

Tol		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Original	error(y_1)	2.2	6.8	10	1.3	0.2	0.05
DASPK	error(y_2)	2.2	6.8	10	1.3	0.2	0.05
	steps	10	33	44	54	48	73
Global	error(y_1)	0.08	3×10^{-3}	1×10^{-3}	1×10^{-5}	3×10^{-6}	3.5×10^{-7}
error	error(y_2)	0.08	3×10^{-3}	1×10^{-3}	1×10^{-5}	3×10^{-6}	3.5×10^{-7}
control	steps	36	53	85	155	192	291

The solution is

$$\begin{cases} y_1(t) = (1+t)^{\frac{1}{2}} \cos(t^2), \\ y_2(t) = (1+t)^{\frac{1}{2}} \sin(t^2). \end{cases}$$

We took the final time to be $T = 10$. The numerical results are listed in Table 6.

Example 4 (mixed system with unstable mode). Consider $\dot{y} + \left(\begin{smallmatrix} 0 & -1 \\ -1 & 0 \end{smallmatrix}\right) y = 0$, $y(0) = \left(\begin{smallmatrix} 2 \times 10^{-4} \\ 0 \end{smallmatrix}\right)$, with solution $y_1(t) = 10^{-4}(e^t + e^{-t})$, $y_2(t) = 10^{-4}(e^{-t} - e^t)$, and final time $T = 10$. The numerical results are listed in Table 7.

From the above examples, we can see that the global error control nearly always generates errors below the tolerances. When the system has an unstable mode (for example, in Example 1 with $a = 1$ and Examples 3 and 4), global error control yields a much more accurate solution than the original DASPK. When the system is mildly stable (for example, in Example 1 with $a = -1$), the original DASPK has already done a good job; thus the global error control does not give much improvement. For stiff systems (for example, in Example 1 with $a = -20$), the global error control produces an acceptable solution with fewer time steps than the original DASPK by recognizing that the condition number is small. One might argue that even in the unstable cases, we can simply tighten the error tolerances to improve the final result. We can always do that, but we do not know how much we should change the tolerance. The global error control does that automatically.

REFERENCES

[1] R. Becker, V. Heuveline, and R. Rannacher, *An optimal control approach to adaptivity in computational fluid mechanics*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 105–120.

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

[3] R. Becker, H. Kapp, and R. Rannacher, *Adaptive finite element methods for optimal control of partial differential equations: Basic concept*, SIAM J. Control Optim., 39 (2000), pp. 113–132.

[4] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics Appl. Math. 14, SIAM, Philadelphia, 1995.

[5] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, *VODE, a variable-coefficient ODE solver*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1038–1051.

[6] Y. Cao, S. Li, L. Petzold, and R. Serban, *Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution*, SIAM J. Sci. Comput., 24 (2003), pp. 1076–1089.

[7] Y. Cao, S. Li, and L. Petzold, *Adjoint sensitivity analysis for differential-algebraic equations: Algorithms and software*, J. Comput. Appl. Math., 149 (2002), pp. 171–191.

[8] Y. Cao and L. Petzold, *A subspace error estimate for linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 787–801.

[9] G. Dahlquist, *On the control of the global error in stiff initial value problems*, in Numerical Analysis, Dundee, 1981, Lecture Notes in Math. 912, Springer-Verlag, Berlin, 1981, pp. 38–49.

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

[11] D. Estep, D. H. Hodges, and M. Warner, *Computational error estimation and adaptive error control for a finite element solution of launch vehicle trajectory problems*, SIAM J. Sci. Comput., 21 (2000), pp. 1609–1631.

[12] D. Estep, M. Larson, and R. Williams, *Estimating the error of numerical solutions of systems of reaction-diffusion equations*, Mem. Amer. Math. Soc., 146 (2000).

[13] D. Estep and R. William, *Accurate parallel integration of large sparse systems of differential equations*, Math. Models Methods Appl. Sci., 6 (1996), pp. 535–568.

[14] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[15] M. B. Giles, *On adjoint equations for error analysis and optimal grid adaptation in CFD*, in Frontiers of Computational Fluid Dynamics (1998), D. A. Caughey and M. M. Hafez, eds., World Scientific, River Edge, NJ, 1998, pp. 155–169.

[16] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations* I*: Nonstiff Problems*, Spring-Verlag, New York, 1987.

[17] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II*: Stiff and Differential-Algebraic Problems*, Springer-Verlag, New York, 1991.

[18] E. Hairer, *Backward analysis of numerical integrators and symplectic methods*, Ann. Numer. Math., 1 (1994), pp. 107–132.

[19] R. Hartmann and P. Houston, *Goal-oriented a posteriori error estimation for compressible fluid flows*, in Numerical Mathematics and Advanced Applications, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer-Verlag, Berlin, 2003, pp. 775–784.

[20] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley & Sons, New York, 1962.

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

[22] C. S. Kenney, A. J. Laub, and M. S. Reese, *Statistical condition estimation for linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 566–583.

[23] C. S. Kenney and A. J. Laub, *Small-sample statistical condition estimates for general matrix functions*, SIAM J. Sci. Comput., 15 (1994), pp. 36–61.

[24] S. Li and L. Petzold, *Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement*, J. Comput. Phys., to appear.

[25] A. Logg, *Multi-Adaptive Error Control for ODEs*, Preprint 2000–03, Chalmers Finite Element Center, Göteborg, Sweden, 2000.

[26] K. Moon, A. Szepessy, R. Tempone, and G. Zouraris, *A variational principle for adaptive approximation of ordinary differential equations*, Numer. Math., 96 (2003), pp. 131–152.

[27] K. Moon, A. Szepessy, R. Tempone, and G. Zouraris, *Convergence rates for adaptive approximation of ordinary differential equations*, Numer. Math., 96 (2003), pp. 99–129.

[28] J. R. Rice, *A theory of condition*, SIAM J. Numer Anal., 3 (1966), pp. 287–310.

[29] L. F. Shampine and H. A. Watts, *Global error estimation for ordinary differential equations*, ACM Trans. Math. Software, 2 (1976), pp. 172–186.

[30] L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations, the Initial Value Problem*, W. H. Freeman, San Francisco, 1975.

[31] R. D. Skeel, *Thirteen ways to estimate global error*, Numer. Math., 48 (1986), pp. 1–20.

[32] H. J. Stetter, *Global error estimation in ODE-solvers*, in Numerical Analysis, Lecture Notes in Math. 630, Springer-Verlag, Berlin, Heidelberg, New York, 1978, pp. 179–189.

[33] M. Utumi, R. Takaki, and T. Kawai, *Optimal time step control for the numerical solution of ordinary differential equations*, SIAM J. Numer. Anal., 33 (1996), pp. 1644–1653.