ELSEVIER Journal of Computational and Applied Mathematics 82 (1997) 21–39

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

Numerical investigations on global error estimation for ordinary differential equations[1]

René Aïd[a,b,*], Laurent Levacher[a]

[a] *EDF-DER-ER-FCR 1, Avenue du Général de Gaulle F-92141 Clamart Cedex, France*

[b] *LMC, IMAG, 51, Rue des Maths F-38041 Grenoble Cedex 9, France*

Received 2 September 1996; received in revised form 1 April 1997

Abstract

Four techniques of global error estimation, which are Richardson extrapolation (RS), Zadunaisky's technique (ZD), Solving for the Correction (SC) and Integration of Principal Error Equation (IPEE) have been compared in different integration codes (DOPRI5, DVODE, DSTEP). Theoretical aspects concerning their implementations and their orders are first given. Second, a comparison of them based on a large number of tests is presented. In terms of cost and precision, SC is a method of choice for one-step methods. It is much more precise and less costly than RS, and leads to the same precision as ZD for half its cost. IPEE can provide the order of the error for a cheap cost in codes based on one-step methods. In multistep codes, only RS and IPEE have been implemented since they are the only ones whose theoretical justification has been extended to this case. There, RS still provides a more reliable estimation than IPEE. However, as these techniques are based on variations of the global error, irrespective of the numerical method used, they fail to provide any more usefull information once the numerical method has reached its limit of accuracy due to the finite arithmetic.

Keywords: Global error estimation; Richardson extrapolation; Zadunaisky's technique; Solving for the correction

AMS classification: 65L05

1. Introduction

When using a numerical method to obtain the solution of an ordinary differential equation, one is interested in the reliability of the values obtained. Even if the computation is performed with a control of the local error, the global error, i.e. the true error, can be such that no significant figures of the exact solution are found. This phenomenom is mainly due to the instability of the solution of the differential equation, and can also occur in stiff problems.

[*] Correspondence address: LMC, IMAG, 46 Avenue Felix Vaillet, BP 53x, F-38041, Grenoble Cedex, France. E-mail: rene.aid@imag.fr.

[1] This research is supported by the Research and Development Division of Electricité de France.

PII S0377-0427(97)00079-4

This is typically what happens in the modelisation of some very unstable electrical networks in the industrial code Eurostag [38] that motivates this research. The numerical integration of the differential equations governing the behaviour of the electrical network can exhibit very unstable oscillating solutions. In this case, it is important for us to determine whether the numerical solution still have a physical meaning or if the error growth in this unstable phase spoiled entirely the accuracy of the solution.

It would be fine if one could provide a sharp bound of the global error. But, this requirement is rather difficult to achieve and one is usually satisfied with an estimation.

But even in that case, there is room for two different approaches of the problem. Because of the extra-cost needed to achieve a global error estimation, one can develop either a specific numerical method as cheap as possible with a global error estimation capability or a recipe that can be applied to a large class of numerical methods and can be implemented in numerical codes as an option given to the user.

An example of the first approach is given by Dormand et al.'s work on explicit Runge–Kutta methods [9–11, 13–15].

In the second case, it is no longer possible to modify the numerical method. This is the approach that interests us. A discussion of thirteen of such techniques can be found in [34].

This paper is only concerned with four of them:

1. Richardson extrapolation (RS),
2. Zadunaisky's technique (ZD),
3. Solving for the correction (SC),
4. Integration of the principal error equation (IPEE).

The other estimators listed in [34] are either variants of these or much less popular.

We are interested in the behaviour of these techniques when used in standard numerical integration codes. A global error estimation can be used in two different ways. A user might just want to check if the integration performed well and get confidence in the numerical solution. Another possibility is to use the global error estimation to get a better numerical solution. In the first case, the order of magnitude of the global error and one significant digit are enough. In the latter case, one will look for more significant digits and this case turns out be equivalent to an improvement of the original numerical method. As far as we are concerned, this last point does not fit our goal. We are more concerned by reliable global error estimation techniques.

Numerical tests on these estimation techniques and related variants have already been done in a separate way. One can find tests in [32] for RS applied to a Runge–Kutta–Fehlberg 5(4) method with local error control. In [39, 16, 28], one can find tests for ZD in Runge–Kutta codes as well as in multistep codes. Tests also appear in [9, 13] for ZD applied to some explicit Runge–Kutta methods. In [28], SC has been tested for general RK formulae and in [11, 14, 15, 7] for customized explicit RK formulae. In [37], numerical experimentations of IPEE is carried in the Adams-PC code STEP [3].

There also exists a survey [29] on different global error estimation techniques presenting both theoretical aspects and numerical comparisons.

All these separate tests make it rather difficult to know what can be expected for each of these estimation techniques on a given problem and a given code.

The purpose of this paper is to compare the former four global error estimation techniques on the basis of the same integration code and the same systems of ODE and to furnish by this way a possibility for the user to be able to chose a suitable estimator to achieve his purpose.

First, in Section 2, the theoretical aspects of these techniques are presented. A survey of the work done on each estimator is provided.

Then, in Section 3, the numerical tests are presented. They are divided into two parts: one-step code (DOPRI5) and multistep codes (DVODE, DSTEP). The main results of these tests permits to sort these estimators in terms of cost and accuracy. Solving for the correction is the method of choice for one-step methods. It is much more accurate than RS, less costly and leads to the same accuracy as ZD for half its cost. IPEE can provide in one-step codes usefull information. In multistep codes, RS is still a reliable estimator. A last point to note is that these techniques fail to be able to validate a numerical solution in the case where the numerical method has reached its limit of accuracy. Due to the finite arithmetic, this limit can be already reached for large tolerances in the case of very unstable systems or methods.

2. Theoretical background

An initial value problem is given as

$$\dot{y}(t) = f(t, y(t)), \qquad y(0) = y_0, \tag{1}$$

where $t \in [0, T], T > 0$ and f is a vectorial function that satisfies a Lispschitz condition relatively to y. A numerical method provides approximations y_i at $t_i = t_{i-1} + h_{i-1}, t_0 = 0$ of $y(t_i)$, $i = 0, \ldots, M$. The difference $E_i = y_i - y(t_i)$ denotes the *global error* commited by the method at instant t_i.

$\overline{E}_i$ is said to be a *valid* estimation of E_i of relative order $r > 0$ when $E_i = \overline{E}_i(1 + \mathcal{O}(H^r))$ where $H = \max_i h_i$. It is worth noting here that the values $\overline{y}_n = y_n - \overline{E}_n$ are of order $p + r$ when y_n are of order p. Hence, looking for a valid global error estimation is equivalent to looking for a better value than the y_n. Moreover, an interest of such a valid estimator is to lead to asymptotic bounds for the global error [5, 30].

The theoretical justification of the validity of the estimation techniques listed in the introduction relies on the existence and on the structure of the asymptotic expansion of the global error. Before stating the existence theorem, we recall some definitions.

For a one-step method, the *local error* at t_i is defined by $\varepsilon_i = y_i - u(t_i)$ where u is given by

$$\dot{u}(t) = f(t, u(t)), \qquad u(t_{i-1}) = y_{i-1}.$$

The numerical method is supposed to be of order $p \geqslant 1$ and that, for a one step method,

$$\varepsilon_i = d_{p+1}(t_i)\, h_i^{p+1} + \cdots + d_N(t_i)\, h_i^N + \mathcal{O}(h_i^{N+1}).$$

Theorem 1 (Gragg [20]). *Suppose that the integration of* (1) *is carried using a stable one-step method of order* $p > 1$ *with constant step-size integration* h. *Then, uniformly on* $[0, T]$, *one has*:

$$y_i - y(t_i) = h^p\, e_p(t_i) + \cdots + h^{N-1}\, e_{N-1}(t_i) + \mathcal{O}(h^N), \tag{2}$$

where $e_k, k = p, \ldots, N-1$ *is the solution of*

$$\dot{e}_k = f_y(t, y)\, e_k + \Phi_k, \qquad e_k(0) = 0. \tag{3}$$

For $k = p$, $\Phi_p = d_{p+1}$ *and the differential equation* (3) *is called* the principal error equation.

The important fact about the expansion (2) is the isolation of the influence of h from functions that depend only on time. To improve the values y_n, one just has to kill as many terms as possible in (2). Once this fact noted, two different ways are possible to achieve this purpose. One can use either variations of h or variations of e_k. The first technique leads to extrapolation methods and in particular to RS. The second one is the *defect correction principle*, and gives IPEE, ZD and SC.

The main problem with regard to an effective integration code is to know whether such an expansion still exists. For such codes, the step-size is no longer constant and, in codes based on multistep formulae, even the order varies. Nevertheless, some properties of the global error are known in these cases.

In the case of a variable step-size integration, the above theorem can be readily extended with h replaced by the maximum step-size H if it is supposed that the step-size is given by a relation such as $h_i = \theta(t_i)H$, where θ is a sufficiently smooth function [23]. The use of a local error control leads to functions θ being only piecewise differentiable. The expansion (2) will remain up to the first term in the case of a step-size given by $h_n = \theta(t_n)H + \mathcal{O}(H^2)$. Moreover, it has been showed in [26] that for classical local error control, the global error admits an asymptotic expansion of the form (2) up to the first term with respect to the absolute tolerance fixed by the user.

For k-steps linear multistep methods, the fact that they need k starting values makes both the analysis and the expansion of the global error more complex. Even in the case of starting values provided by a RK with a sufficiently high order, the asymptotic expansion of the global error can exhibit a new behavior. For a detailed analyis of this case, the reader is refered to [20, 22]. For our purpose, we just need to recall that an asymptotic expansion of the form (2) exists for Adams methods of order p with starting values of order p. In the case of BDF formulae started with values of order p, the asymptotic expansion of the global error has no longer the form (2).

In modern numerical codes implementing linear multistep methods, the starting values are not computed with RK formulas. Both the step-size and the order vary, starting with a method of order 1. This is typically the case of the codes used here (DVODE, DSTEP). In this case, the existence and the form of the global error asymptotic expansion is even more complex. For an analysis of this situation, the reader is refered to [19, 8].

For an Adams code such as DSTEP and with reasonnable assumptions Shampine et al. showed in [33] that the global error admits an asymptotic expansion of the form (2) up to the first term w.r.t. H.

As it will be seen for each estimator, these facts have direct consequences on the class of codes they can be implemented into.

2.1. Richardson extrapolation

Using the existence of the asymptotic expansion in Theorem 1, a straightforward estimation formula can be deduced. In parallel with the original integration with maximum step size H, another integration is carried out with the same step size selection function, but with maximum step size $H/2$. Denoting the value obtained in this way at time t_i by y^*_{2i}, we have:

$$y_i = y(t_i) + H^p\, e_p(t_i) + \mathcal{O}(H^{p+1}),$$
$$y^*_{2i} = y(t_i) + (H/2)^p\, e_p(t_i) + \mathcal{O}(H^{p+1}).$$

And, we get directly:

$$E_i = \frac{y_i - y_{2i}^*}{1 - 2^{-p}} + \mathcal{O}(H^{p+1}). \tag{4}$$

This relation provides a valid estimation of relative order 1. It is well-known since Henrici [25] and has been the subject of implementations in numerical codes, such as GERK [32].

In terms of cost, this technique is rather expensive as it multiplies by three the cost of the integration.

According to the preceeding paragraph concerning the asymptotic expansion of the global error, as Richardson extrapolation only needs a global error expansion up to the first term, it is still likely to behave well in codes based on one-step methods with local error control and in multistep codes. The only detail to set in the last case is the choice of the order p for variable order integration. In [33], it was assumed that once the code has reached its higher possible order, it is kept constant until the end of the integration. This is not always the case. Nevertheless, it is always possible to assume that the global error is at least of order two on the whole interval. Hence, we took $p = 2$ to compute this estimation in multistep codes.

2.2. Zadunaisky's technique

Another approach has been initiated by the work of Zadunaisky [39]. His idea can be described as follow. Let $m \geqslant 1$ be an integer. After m steps of integration of (1), it is possible to compute an interpolation function P_j, $j = 1, \ldots, J_T$ using the values y_i, $i = (j-1)m, \ldots, jm$. Let $\hat{y}_0 = y_0$. Consider the perturbed problem:

$$\dot{\hat{y}}(t) = f(t, \hat{y}(t)) + d(t,h), \qquad \hat{y}(0) = y_0, \tag{5}$$

where $d(t,h) := \dot{P}_h(t) - f(t, P_h(t))$ is the *defect of* P_h *in* (1) and P_h is defined by $P_h(t) = P_j(t)$ for $t \in [t_{(j-1)m}, t_{jm}]$ with P_j the interpolation polynomial of degree m of the values y_i, $i = (j-1)m, \ldots, jm$.

The integration of (5) with the same method and on the same grid leads to numerical values $\hat{y}_i$, $i = (j-1)m + 1, \ldots, m$. Zadunaisky's global error estimation of E_i is then:

$$Z_i = \hat{y}_i - y_i. \tag{6}$$

It is not as simple as for RS to prove that this estimation technique do provide a valid estimation, and for which numerical methods it does. The following result holds:

Theorem 2 (Frank and Veberhuber [17]). *For an ERK of order p and for a constant step-size integration, and with P_h, the piecewise polynomial function described above, the relative order of Zadunaisky's technique is*:

$$\begin{array}{lll} 0 & \textit{if} & m < p, \\ m - p & \textit{if} & p \leqslant m \leqslant 2p, \\ p & \textit{if} & 2p \leqslant m. \end{array} \tag{7}$$

The main hypothesis needed for its proof is the existence of an asymptotic expansion of the form (2) for the global error. Then, it is necessary to analyse the variational equations verified by the functions $e_{h,k}$, $k=p,\dots,$ involved in the global error asymptotic expansion of $\hat{y}_i - y_i$.

The cost of the interpolation process can be neglected when compared to the integration of the perturbed problem (5). It leads to the same number of extra f evaluation needed as in RS. Hence, with the same extra-cost, Zadunaisky's technique can achieve a relative order much more better of p.

The tests presented in [39] use both one-step and multistep methods, including predictor–corrector methods. For one-step methods, the results presented are those of a linearized version of equation (5). They show that this technique can lead to a very precise global error estimation. But, in some cases, due to error interpolation, the estimation becomes poor. For multistep methods, the quality of the estimation is related to the stability of the method. The tests in [39] performed with the Adams type methods show a satisfactory estimation, whereas those performed with Milne type methods are very poor.

A variant of Zadunaisky's classical algorithm has been described in [18] and tested in [16] for various differential equations and various methods. It can be summarised as follow: first, integrate the problem (1) with a lower order method, then integrate the perturbed problem (5) with the same low order method, then use the difference between these values as a global error estimation. This variant has the advantage of reducing the cost of the re-integration needed to get the z_i values. With the same polynomial as for Zadunanisky's classical technique, the same order is achieved with a method of order $m-p$. The tests interesting us are IVP for ODEs [16, Algorithm 1, p. 110]. To perform the tests, the authors used an Adams-Bashford method of order p, $(p=2,5)$ together with a starting procedure given by an ERK of order p. It is known that in this case, the multistep method exhibits an asymptotic expansion of the form (2). This is the only case known to the authors for which Zadunaisky's technique, and defect correction in general, is not bound to fail when applied to linear multistep methods.

For a code based on linear multistep methods with variation of order, it is not even obvious to see whether a relative order greater that one can still be achieved, according to what have been said on the global error expansion in this case. Moreover, as the order is no longer constant, the choice of the parameter m can not be fixed using Theorem 2. Hence, it appears not necessary to focus more attention to ZD in modern multistep codes.

From a theoretical point of view, Zadunaisky's technique has given rise to two different research directions.

First, this recipe has been extended to a general principle, the so-called *defect correction principle*, that can be applied to differential equations (partial and ordinary) and general numerical methods, both for a global error estimation purpose and for an iterative improvement of the numerical solution (iterated defect correction) [35]. An analysis of its convergence can be found in [18]. A variant of this principle has been given by Hairer [21]. It consists in consisdering Zadunaisky's technique as a one-step method whose step-size is $m \times h$. Instead of starting the integration from t_{im} with the lowest order value y_{im}, one uses the value obtained after correction. A purely algebraic proof of the order of this variant is provided, together with examples of admissible function $d(t,h)$.

Second, Dormand et al. developed optimized ERK for Zadunaisky's technique [9, 13, 11], to overcome problems occurring from interpolation error. Here, their optimization consisted in using the free parameters of ERK methods to achieve the order given in Theorem 2 with m as low as possible, or to achieve a higher order with the same degree [9, 13]. They also used different interpolation process, as interpolation of the $f(y_i)$ and Hermite interpolation. Finally, they show that Zadunaisky's technique can benefit of continuous ERK. In this frame, one global error estimation per step could be achieved using the continuous polynomial extension instead of P_h [11, 14]. The numerous tests presented [11] clearly show that the accuracy of Zadunaisky's technique can be improved in comparison with the tests obtained in [39] when used together with some special dense output ERK formula.

As explained in the introduction, our purpose is not to look for the best numerical method with a global error capability. If the reader is to chose an explicit RK and need a global error capability, there is no doubt that they are methods of choice. But, since our intention is to compare global error estimation on a general ground, we limited our tests to Zadunaisky's classical technique applied to numerical codes based on one-step methods.

2.3. Solving for the correction

The idea of using the interpolation procedure described in the former section readily leads to another algorithm [34]. With the same definition of the function P_h as above, one can consider the function:

$$\mathscr{E}_h = P_h - y \tag{8}$$

which gives the value of the global error at each t_i, $i = 0, \dots, M$. It satisfies the ordinary differential equation:

$$\dot{\mathscr{E}}_h(t) = \dot{P}_h(t) - f(t, P_h(t) - \mathscr{E}_h(t)), \qquad \mathscr{E}_h(0) = 0. \tag{9}$$

Using the same numerical method as for (1), one gets the numerical values $\mathscr{E}_n$. They give a valid global error estimation with the same relative order as in Zadunaisky's technique [28].

The main advantage of SC over ZD is that the extra-cost needed to get a valid global error estimation is much lower. As f appears only once in (9), the number of f evaluations per step is divided by two. Hence, solving for the correction provides an estimation of relative order p and just doubles the cost of the integration.

A different ERK can be applied to (9). But, Dormand et al. stressed the fact that it should be at least of the same order as the ERK used on (1) to get a valid global error estimation [10].

Following the same analysis as they did for Zadunaisky's technique, they were able to develop special continuous ERK methods so as to optimize the possible global error estimation obtained by integration of (9) [11, 14]. In a more recent paper [10], SC is finally included in continuous ERK so as to provide a pair of solution (y_n, y_n^*) with different global order. These methods are typically the result of the second approach mentioned in the introduction.

For the same reasons as for ZD, SC has not been implemented in multistep codes.

2.4. Integration of the principal error equation

At first sight, it seems rather difficult to use the principal error Eq. (3) because of the need of the Jacobian of f and, worse, of the exact solution of the original problem (1).

However, it is possible to integrate (3) with a low order formula, and to replace both the Jacobian and the exact solution y by approximations and still to have a valid global error estimation.

Applying forward Euler method to (3) leads to

$$e_{p,i+1} = e_{p,i} + h_i\, f_y(t_i, y(t_i))e_{p,i} + h_i\, d_{p+1}(t_i), \qquad e_{p,0} = 0.$$

If we consider now $v_i = H^p e_{p,i}$. It is clear that v_i is a valid estimation of E_i because $v_i = H^p(e_p(t_i) + \mathcal{O}(H))$ and then $v_i = E_i + \mathcal{O}(H^{p+1})$.

Then, using a classical approximation of $f_y(t_i, y(t_i))$ and the first term of the expansion of the local error ε_i, we get

$$v_{i+1} = v_i + h_i(f(t_i, y_i) - f(t_i, y_i - v_i)) + \varepsilon_i, \qquad v_0 = 0. \tag{10}$$

This relation has been derived also by Stetter, using the defect correction principle [36] and implemented in an Adams-PC code [37].

The relation (10) leads to a very cheap estimator. It requires just one more evaluation function $f(t_i, y_i - v_i)$. But, the need of the local error makes it necessary to use a local error estimate. The estimator (10) will remain valid if the ε_i term is replaced by a valid estimation $\hat{\varepsilon}_i = \varepsilon_i(1 + \mathcal{O}(h_n))$.

The use of local extrapolation to select the step-size implies that the local error estimate is no longer valid. Hence, we used two different features. For one-step methods, we used the defect of the function P_h used for ZD and SC. The reason for this is that $h\,d_h = \varepsilon_h + \mathcal{O}(h^{p+2})$. This feature increases the cost of the estimator of one f evaluation per step. In this case, the estimator obtained is renamed DIPEE.

For multistep methods, we used both the local error estimate given by the code and the exact local error to see the best this estimator can do.

3. Numerical computations

To perform the numerical tests, we have chosen several existing integration codes :

1. DOPRI5 [12] a one-step integration method based on a RKDP5(4) formula, with an error per step control and local extrapolation.
2. DVODE [6, 4], a multistep integration method using BDF formulae. Starting order is one. No local extrapolation is done.
3. DSTEP [31], a multistep method based on Adams formulae. Starting order is one. Local extrapolation is done.

DVODE and DSTEP are available at the Netlib web site (`www.netlib.org`). The DOPRI5 we used is available at Hairer's web page (`www.unige.ch/math/folks/hairer`).

In DVODE and DSTEP, only RS and IPEE were implemented.

All the estimation techniques were implemented in DOPRI5.

We used six systems whose exact solutions were known:

I. A linear unstable system of dimension 2 [24]:

$$y' = \begin{bmatrix} -1 + \frac{3}{2}\cos^2 t & 1 - \frac{3}{2}\sin t \cos t \\ -1 - \frac{3}{2}\sin t \cos t & -1 + \frac{3}{2}\sin^2 t \end{bmatrix} y$$

with initial condition $y(0) = (1,0)^t$, integrated on $[0,10]$. Its exact soluion is:

$$Y(t) = \begin{bmatrix} e^{t/2}\cos t & e^{-t}\sin t \\ -e^{t/2}\sin t & e^{-t}\cos t \end{bmatrix} y(0).$$

II. The scalar very unstable equation [32]:

$$y' = 10(y - t^2), \qquad y(0) = 0.02$$

on $[0,2]$. Its solution is:

$$y(t) = 0.02 + 0.2t + t^2.$$

III. A stable non-linear system of dimension 4:

$$y_1' = -y_3 y_1 + y_2, \qquad y_2' = -y_1 - y_3 y_2, \qquad y_3' = y_4, \qquad y_4' = -y_3$$

with initial condition $y(t_0) = (1,1,1,1)^t$, integrated on $[0,7]$.
Its exact solution is:

$$y_1(t) = (\cos t + \sin t)e^{-1 + \cos t - \sin t}, \qquad y_2(t) = (\cos t - \sin t)e^{-1 + \cos t - \sin t},$$
$$y_3(t) = \cos t + \sin t, \qquad y_4(t) = \cos t - \sin t.$$

IV. A linear stiff system of dimension 4 [29]:

$$y' = \begin{bmatrix} -0.1 & -49.9 & 0 \\ 0 & -50 & 0 \\ 0 & 70 & -120 \end{bmatrix} y$$

with $y(0) = (2,1,2)^t$, and $t \in [0,1]$. Its solution is:

$$\phi(t_0, y_0; t) = \begin{bmatrix} e^{-(t-t_0)/10} & e^{-50(t-t_0)} - e^{-(t-t_0)/10} & 0 \\ 0 & e^{-50(t-t_0)} & 0 \\ 0 & e^{-50(t-t_0)} - e^{-120(t-t_0)} & e^{-120(t-t_0)} \end{bmatrix} y_0.$$

V. Problem A3 from DETEST package [27], $\dot{y}(t) = \cos(t)\,y(t)$, $y(0) = 1$, $t \in [0,20]$.

VI. Problem A4 from DETEST package, $\dot{y} = 0.25\,y\,(1 - 0.05\,y)$, $y(0) = 1$, $t \in [0,20]$.

For ZD and SC, we used polynomials of degree 10 and divided differences.

To compare the precision of the estimators, we have used the following feature. We have designed a function that returns 0 if the magnitude of the error is not given by the estimator, and 1 plus the number of significant figures well-estimated if the magnitude of the error is correct. The value one is returned also in the case where the magnitude of the error is correct and the sign of the error is not. Then, we have computed the means on the whole interval of integration. The value obtained is a measure of the *efficiency* of the estimator. An efficiency

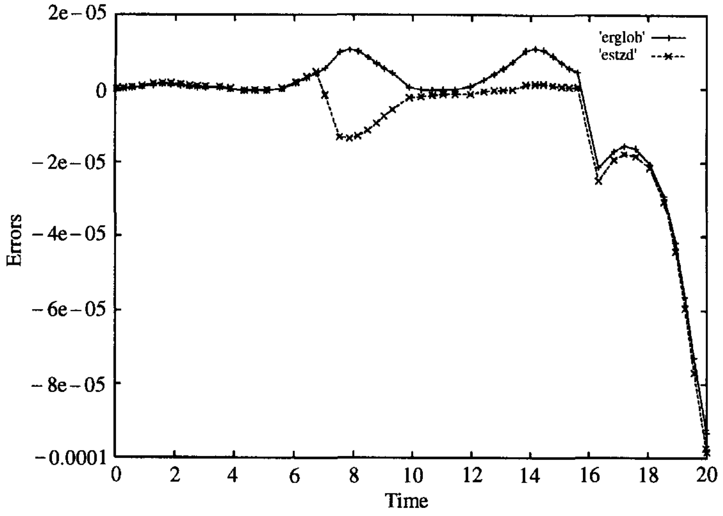

Fig. 1. ZD-system V: atol$=10^{-5}$.

near one means that the estimator gives correctly the order of magnitude of the global error and a significant digit. But, a lower value of 0.5 is enough to insure the magnitude. This measure does not provide a dynamic behaviour of the estimator, but gives a correct view of its precision.

All the computations were carried on a SUN4 using double precision arithmetic. The whole tests are available in [1].

3.1. One-step integration method

Before we comment on the whole results, we begin with an illustration of the problem arising from the use of a basic interpolation procedure for ZD and SC. In Figs. 1–4 are represented both the global error of DOPRI5 on system V and an estimator, ZD or RS. These figures enable to compare the behaviour of ZD and RS for both large tolerance (10^{-5}, Figs. 1 and 3) and small tolerance (10^{-10}, Figs. 2 and 4). In this case, ZD and SC produce roughly the same estimation. In both cases RS behaves well, whereas for large tolerances, ZD clearly exhibits an unsmooth behaviour (Fig. 1). For small tolerances, step-sizes are sufficiently small to make it disappear. Even if it does not prevent ZD and SC from providing useful information on the global error, it just illustrates the difference between them and RS.

RS only needs the existence of the functions e_k of the asymptotic expansion of the global error while ZD and SC need some regularity of the e_k's to ensure a fast enough convergence of the perturbed $e_{h,k}$. On problem V, the functions e_k are only piecewise differentiable due to the variation of the step-size (see [7] for more illustrations, and Fig. 2). Hence, it slows down the convergence of the $e_{h,k}$. This is typically the kind of behaviour that lead Dormand and Prince to design special ERK more suited to ZD and SC.

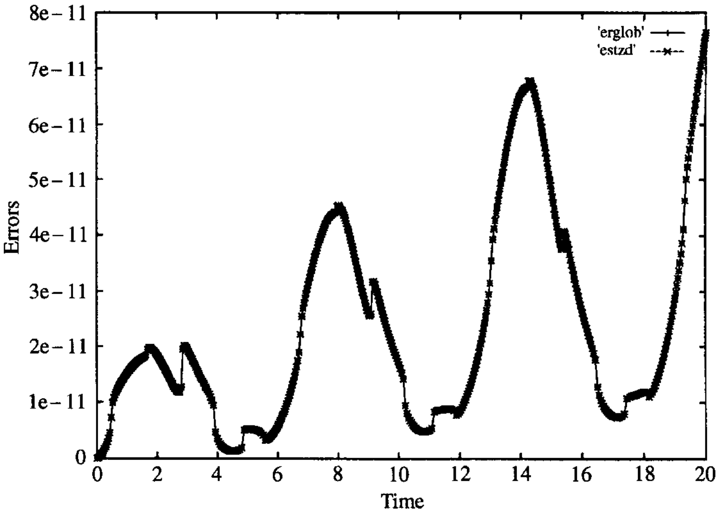

Fig. 2. ZD-system V: atol $= 10^{-10}$.

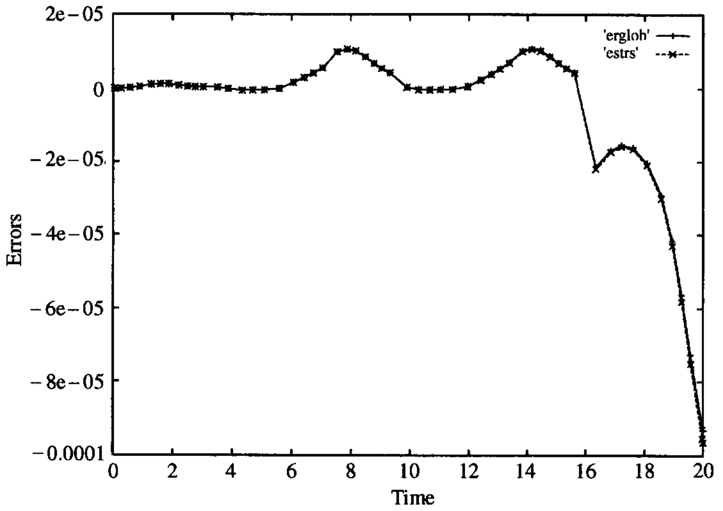

Fig. 3. RS-system V: atol $= 10^{-5}$.

On Tables 1 and 2 are given the efficiency of each estimator as defined previously. In general, ZD and SC give much more significant figures than RS or DIPEE. It should be noticed that the systems I, III and V present no difficulty for the estimators, with the exception of system V for DIPEE. RS gives in a very constant way the order of magnitude and one significant digit of the global error. As soon as the tolerance has been diminished to a certain level (see system V), ZD and SC gives much more significant digits and their results can be used to improve in a drastic way the numerical solution. If only the order of magnitude is under investigation, DIPEE can provide a cheap way to get an idea of it.

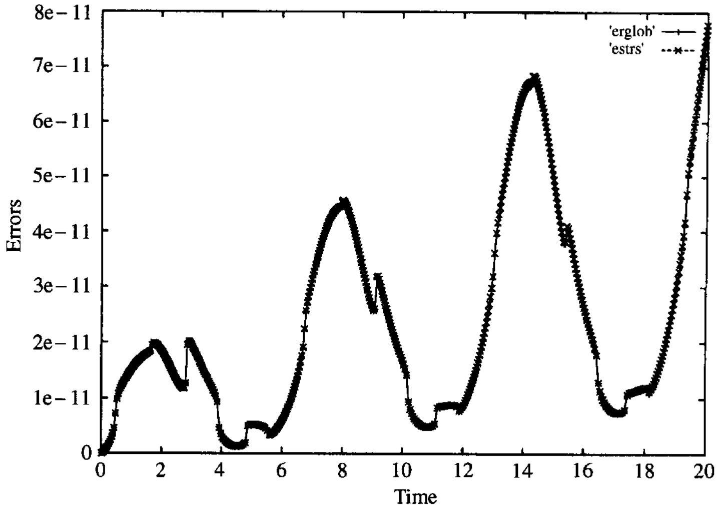

Fig. 4. RS-system V: atol $= 10^{-10}$.

Table 1
RS, ZD, SC, DIPEE - DOPRI5 - I/III

−log atol	3	4	5	6	7	8	9	10	11	12
	2.0	1.7	1.9	2.4	2.7	2.9	3.2	3.1	2.2	1.3
I	4.3	5.5	6.8	6.6	6.4	6.0	4.7	3.9	3.0	1.1
	4.3	5.5	6.8	6.5	6.4	6.5	6.0	4.9	4.1	3.1
	1.1	0.4	0.6	1.2	1.5	1.7	1.8	2.2	1.6	0.7
	1.0	2.1	2.5	2.5	1.4	0.3	0.1	0.0	0.0	0.0
II	0.1	3.5	3.0	2.4	1.3	0.3	0.1	0.0	0.0	0.0
	0.1	3.5	3.0	2.4	1.3	0.3	0.1	0.0	0.0	0.0
	0.1	0.0	0.0	0.2	0.5	0.7	0.0	0.0	0.0	0.0
	2.3	2.5	2.6	2.3	2.5	2.7	2.8	2.8	2.8	2.1
III	2.4	1.3	2.5	3.3	4.2	5.2	6.0	4.1	3.0	2.3
	2.3	1.1	2.5	3.3	4.1	5.0	6.0	5.5	4.4	3.4
	0.2	0.1	0.4	0.3	0.7	1.5	2.0	2.1	1.8	0.8

The systems II, IV and VI are treated with less facility. One would expect that the quality of an estimator increases when the absolute tolerance decreases. None of the estimators provide even the magnitude of the error after some level of tolerances is reached. This behaviour can be explained having a closer look at the global error itself. On Table 3 is given the maximum of the absolute value of the global error with respect to the tolerance for systems II, IV, V and VI. Here, the system V is used as a comparison. Due to finite computer arithmetics together with instability or stiffness of the problem, ERK methods quickly reach their limit of accuracy. Therefore, once the global error

Table 2
RS, ZD, SC, DIPEE - DOPRI5 - III/VI

−log atol	3	4	5	6	7	8	9	10	11	12
	1.0	0.8	0.5	0.3	0.0	0.0	0.0	0.0	0.0	0.0
IV	1.1	0.8	0.6	0.3	0.0	0.0	0.0	0.0	0.0	0.0
	1.1	0.8	0.6	0.3	0.0	0.0	0.0	0.0	0.0	0.0
	0.3	0.4	0.4	0.3	0.0	0.0	0.0	0.0	0.0	0.0
	2.4	2.4	2.4	2.4	2.4	2.0	2.7	2.9	3.4	1.9
V	0.9	0.1	0.8	0.7	1.9	2.6	3.2	4.9	4.6	3.2
	0.9	0.1	0.8	1.7	2.2	2.4	3.5	4.9	3.7	2.5
	0.0	0.0	0.0	0.1	0.2	0.2	0.6	1.0	1.6	0.4
	2.5	2.5	1.9	1.1	0.4	0.2	0.0	0.0	0.0	0.0
VI	0.0	2.0	1.9	1.1	0.4	0.2	0.0	0.0	0.0	0.0
	0.0	2.2	1.9	1.1	0.4	0.2	0.0	0.0	0.0	0.0
	0.0	0.4	0.6	0.4	0.5	0.1	0.0	0.0	0.0	0.0

has reached its limit, no estimation is anymore possible with these techniques since they are based on its variation.

Of course, in the case of stiff systems one should first take care to use a numerical method that is suited for it. Nevertheless, one can follow Stetter's recommendation to replace the relation (10) by its corresponding formula using either the implicit Euler method or a the PECE scheme explicit Euler–implicit Euler. For other estimators, the possibility of adaptation are less obvious.

3.2. Variable step and variable formula integration

As it is shown in Figs. 5 and 6, RS gives a precise estimation of the global error on system III and IV with DSTEP. The curves of the global error and of the estimation given by RS are the same. This is also the case for other systems. If we compare this estimation with what is obtained with DVODE for the same systems and tolerances in Figs. 7 and 8, we note that the code influences the quality of the same estimation technique. In Tables 4 and 5 is given on the first row, for each system, the measure of the efficiency of RS. If it is compared with what was obtained for a one-step code, it is clear that RS is less precise here. The variations of order are the cause for this. Nevertheless, it still leads a reliable estimation as it gives both in DVODE and DSTEP the correct magnitude of the global error and its first digit.

For those particular codes, IPEE gives a better result than in the former cases. Tables 4 and 5 show for each code and system, the quality of the estimation with the local error estimates provided by the code on the second row, and on the third row, with the exact local error (EIPEE). It appears in both codes that this estimation can potentially provide a better estimation than RS. This explains the precise results obtained with a special local error estimates by Stetter [37].

Moreover, the problems that occurred for one-step methods with systems II, IV and VI are less sensible here on system II. Table 6 shows the maximum of the global error in DVODE. It shows

Table 3
Maximum Global Error – DOPRI5

−log atol	3	4	5	6	7	8	9	10	11	12
II	4.0×10^{2}	3.7×10^{2}	81	12	1.3	3.3×10^{-2}	0.2	0.2	0.2	0.2
IV	1.2×10^{-5}	1.5×10^{-6}	1.5×10^{-7}	2.9×10^{-8}	2.9×10^{-8}	2.9×10^{-8}	3.0×10^{-8}	3.0×10^{-8}	3.0×10^{-8}	3.0×10^{-8}
V	4.2×10^{-3}	2.6×10^{-4}	9.3×10^{-5}	2.6×10^{-6}	5.1×10^{-7}	8.8×10^{-9}	1.4×10^{-9}	7.6×10^{-11}	1.6×10^{-11}	6.3×10^{-13}
VI	2.5×10^{-4}	3.3×10^{-5}	2.4×10^{-6}	2.0×10^{-7}	2.3×10^{-7}	2.3×10^{-7}	2.3×10^{-7}	2.3×10^{-7}	2.3×10^{-7}	2.3×10^{-7}

Table 4
RS, IPEE, EIPEE: DVODE

−log atol	3	4	5	6	7	8	9	10	11	12
I	1.2	1.5	1.1	1.1	1.1	1.2	1.1	0.9	1.2	1.2
	0.8	1.2	0.7	0.6	0.7	0.9	0.8	0.8	0.6	0.5
	1.4	1.7	1.4	1.7	1.9	2.2	2.2	2.5	2.6	2.7
II	2.1	1.8	2.0	2.4	2.6	2.1	1.2	0.7	0.1	0.0
	0.3	0.3	0.4	0.4	0.5	0.5	0.5	0.3	0.1	0.1
	0.4	0.5	0.6	0.6	0.7	0.8	0.9	1.1	1.0	1.0
III	1.2	1.1	1.3	1.3	1.2	1.1	1.1	1.1	1.2	1.2
	0.7	0.8	0.7	0.8	0.9	0.9	0.8	0.7	0.5	0.7
	1.6	1.9	2.2	2.3	2.3	2.4	2.5	2.7	3.0	3.0
IV	1.4	1.3	1.3	1.2	0.7	0.4	0.1	0.0	0.0	0.0
	1.0	0.8	0.7	0.7	0.8	0.1	0.1	0.1	0.0	0.0
	1.7	2.2	2.1	2.3	2.6	3.7	4.5	5.3	6.1	6.6
V	1.5	1.0	1.9	1.4	1.3	1.3	1.3	1.2	1.1	1.1
	0.6	0.6	0.8	0.8	0.9	0.6	0.7	0.7	0.7	0.7
	1.5	1.5	2.1	1.8	2.1	2.2	2.3	2.6	2.5	2.7
VI	1.3	1.4	1.9	1.5	1.0	1.1	1.0	0.8	0.2	0.1
	1.3	1.3	1.3	0.6	1.5	0.8	0.5	0.3	0.2	0.1
	1.9	2.3	2.6	2.6	3.3	3.4	3.9	3.8	3.9	3.9

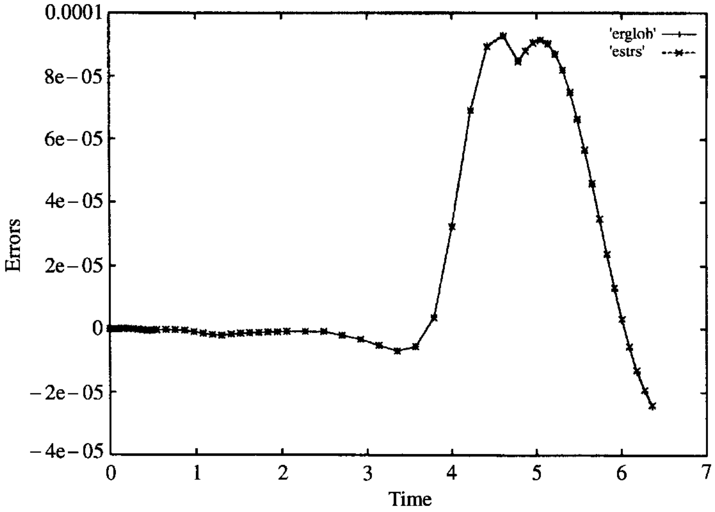

Fig. 5. RS-system III: DSTEP, atol = 10^{-5}.

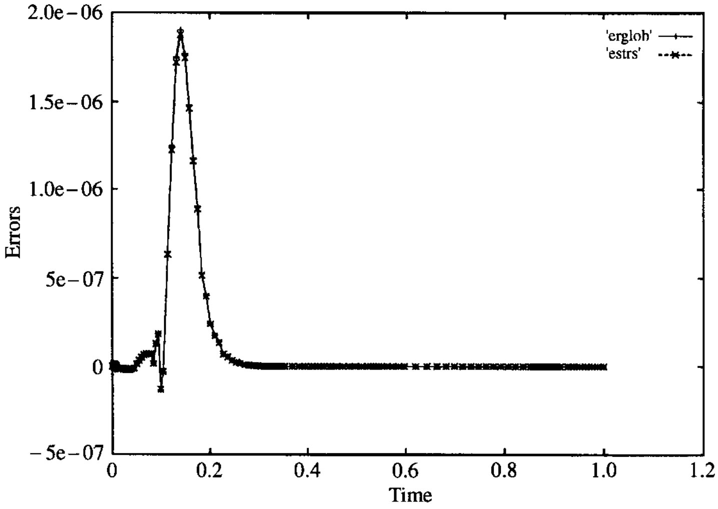

Fig. 6. RS-system IV: DSTEP, atol = 10^{-5}.

that for system II, the stagnation of the global error occurs only for the last acceptable tolerances. For systems IV and VI, the limit of the possible global error is achieved laterer in DVODE as in DOPRI5 and this level of tolerance corresponds to the non-estimation of the global error. The same remark can be done in DSTEP.

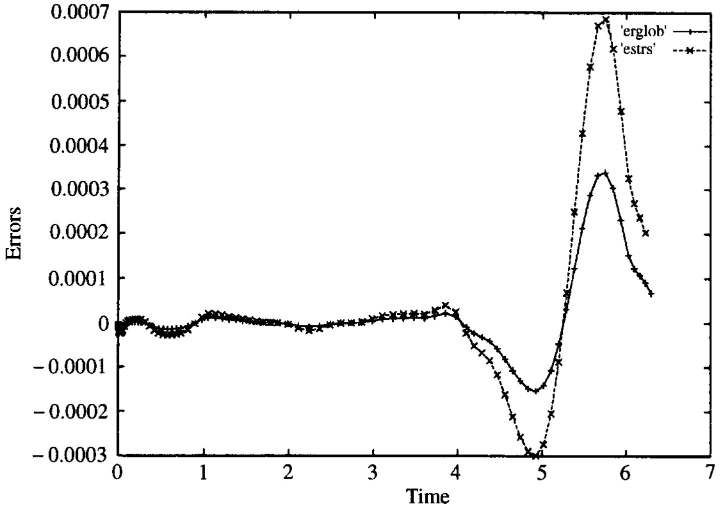

Fig. 7. RS-system III: DVODE, atol = 10^{-5}.

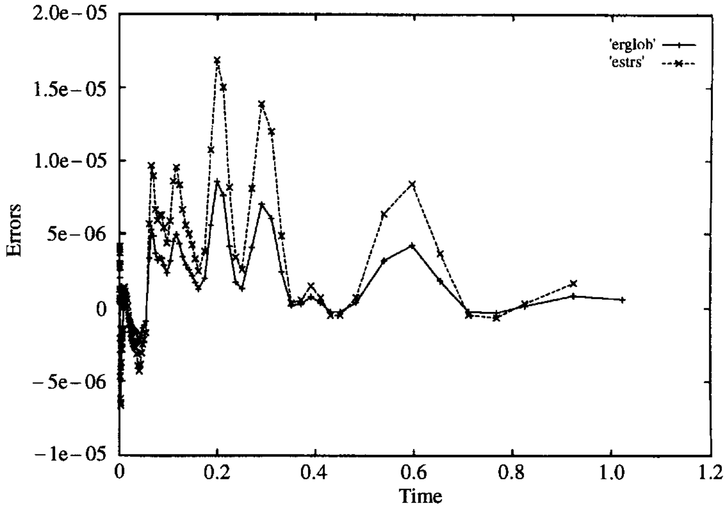

Fig. 8. RS-system IV: DVODE, atol = 10^{-5}.

4. Conclusions

The numerical investigations presented here show that it is possible to get a reliable global error estimation. The tests show that depending on the requirement of the user, the estimators tested here can achieve a wide range of accuracy. To validate a numerical solution it is not necessary to look for a very precise estimator, and IPEE with a local error estimate provided by a defect function is enough. For more precise estimation, ZD and SC can be used as soon as the global error of the

Table 5
RS, IPEE, EIPEE: DSTEP

−log atol	3	4	5	6	7	8	9	10	11	12
I	1.5	1.5	1.2	1.1	1.1	0.7	0.6	0.5	0.7	0.8
	0.7	0.3	0.3	0.3	0.2	0.3	0.4	0.2	0.0	0.0
	1.8	1.7	1.8	2.3	2.5	2.5	2.5	2.9	2.8	3.0
II	1.1	1.3	1.4	0.5	0.7	0.8	0.4	0.2	0.1	0.1
	0.6	0.7	0.4	0.2	0.3	0.5	0.1	0.1	0.0	0.0
	0.5	0.3	0.8	1.0	1.4	1.3	0.8	1.2	1.0	1.2
III	1.3	1.5	1.1	1.1	1.2	0.9	0.7	0.6	0.7	0.8
	0.5	0.7	0.4	0.4	0.5	0.5	0.7	0.1	0.0	0.0
	1.3	1.8	1.8	2.1	2.3	2.5	2.4	2.6	2.9	2.7
IV	0.6	0.6	0.5	0.2	0.2	0.2	0.2	0.2	0.2	0.1
	0.2	0.2	0.2	0.2	0.0	0.1	0.0	0.0	0.0	0.0
	2.8	3.1	3.4	3.7	4.2	4.7	5.2	5.4	5.7	5.6
V	1.4	1.2	1.1	1.2	1.2	1.0	0.7	0.5	0.5	0.6
	0.3	0.2	0.6	0.6	0.0	0.7	0.5	0.2	0.0	0.0
	1.4	1.7	1.8	1.7	1.8	1.9	2.2	2.6	2.4	2.8
VI	1.4	1.5	1.3	1.3	1.1	0.7	0.5	0.5	0.7	0.6
	0.0	0.3	0.7	0.5	0.4	0.2	0.5	0.1	0.0	0.0
	1.8	2.6	2.6	3.2	3.2	3.5	3.9	3.7	4.0	3.7

Table 6
Maximum Global Error – DVODE

−log atol	3	4	5	6	7	8	9	10	11	12
II	4.0×10^{5}	7.3×10^{4}	9.1×10^{3}	9.5×10^{2}	96	9.6	0.8	0.2	0.2	0.2
IV	5×10^{-4}	6.4×10^{-5}	8.5×10^{-6}	2.1×10^{-6}	9.7×10^{-8}	3.7×10^{-8}	3.0×10^{-8}	3.0×10^{-8}	3.0×10^{-8}	3.0×10^{-8}
V	4.4×10^{-2}	3.2×10^{-3}	1.2×10^{-3}	1.6×10^{-4}	1.9×10^{-5}	4.5×10^{-6}	7.7×10^{-7}	6.0×10^{-8}	1.4×10^{-8}	1.9×10^{-9}
VI	3.5×10^{-3}	9.3×10^{-4}	1.5×10^{-4}	1.6×10^{-5}	3.2×10^{-6}	3.0×10^{-7}	2.3×10^{-7}	2.4×10^{-7}	2.4×10^{-7}	2.4×10^{-7}

numerical method implemented admits an expansion of the form (2). In multistep codes, RS still provides a reliable estimation.

For unstable or stiff systems, the useful information provided by the estimators we tested depends less on the tolerances than on the possibility of the numerical method to achieve a better value. This conclusion is coherent with the fact that a valid estimation provides a better value.

Finally, in the case of one-step methods, it is interesting to ask if the kind of optimization developed by Dormand et al. on ERK for SC and ZD to overcome their poor behaviour for large tolerances could have been achieved by considering a different interpolation procedure. For instance, rational interpolation has been successfully used by Alt [2] to local error estimate. Moreover, Hairer proposed in his paper [21] alternatives to classical defect functions that include polynomial functions. This is presently the domain we are working on.

Acknowledgement

The authors would like to thank Philippe Poncet for his kind help, and Gilles Villard for his useful comments.

References

[1] R. Aïd, Estimation de l'erreur globale pour l'intégration numérique d'équations différentielles ordinaires, Technical Report 159, LMC-IMAG, Grenoble, 1996.

[2] R. Alt, Evaluation de l'erreur de discrétisation des méthodes à pas séparés à l'aide d'interpolation rationnelle, in: Les mathématiques de l'informatique, AFCET Colloq. Paris, 1982, pp. 515–524.

[3] L.F. Shampine, M.K. Gordon, Computer solution of ordinary differential equations, Freeman, New York, 1975.

[4] P.N. Brown, G.D. Byrne, A.C. Hindmarsh, VODE: a variable coefficient ODE solver, SIAM J. Sci. Stat. Comput. 10 (5) (1989) 1038–1051.

[5] R. Burlisch, J. Stoer, Numerical treatment of ordinary differential equations by extrapolation methods, Numer. Math. 8 (1966) 1–13.

[6] G.D. Byrne, A.C. Hindmarsh, A polyalgorithm for the numerical solution of ordinary differential equations, ACM Trans. Math. Software 1 (1975) 71–96.

[7] M. Calvo, D.J. Higham, J.I. Montijano, L. Randez, Global error estimation with adaptive explicit Runge–Kutta methods, IMA J. Numer. Anal. 16 (1996) 47–63.

[8] M. Crouzeix, F.J. Lisbona, The convergence of variable-stepsize variable-formula multistep methods, SIAM J. Numer. Anal. 21 (1984) 512–534.

[9] J.R. Dormand, R.R. Duckers, P.J. Prince, Global error estimation with Runge-Kutta methods, IMA J. Numer. Anal. 4 (1984) 169–184.

[10] J.R. Dormand, J.P. Gilmore, P.J. Prince, Globally embedded Runge–Kutta schemes, Ann. Numer. Math. 1 (1994) 97–106. Baltzer A.G. Science Publishers, 1994.

[11] J.R. Dormand, M.A. Lockyer, N.E. McGorrigan, P.J. Prince, Global error estimation with Runge-Kutta triples, Comput. Math. Appl. 18 (9) (1989) 835–846.

[12] J.R. Dormand, P.J. Prince, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. 5 (1980) 977–989.

[13] J.R. Dormand, P.J. Prince, Global error estimation with Runge–Kutta methods II, IMA J. Numer. Anal. 5 (1985) 481–497.

[14] J.R. Dormand, P.J. Prince, Practical Runge–Kutta processes, SIAM J. Sci. Stat. Comput. 10 (5) (1989) 977–989.

[15] J.R. Dormand, P.J. Prince, Global error estimation using RK pairs, in: Cash, Gladwell (Eds.), Computational ODEs, Oxford Univ. Press, Oxford, 1992.

[16] R. Frank, F. Macsek, C.W. Ueberhuber, Iterated defect correction for differential equations, Part II: Numerical experiments, Computing 33 (1984) 107–119.

[17] R. Frank, C.W. Ueberhuber, Iterated defect correction for Runge–Kutta methods, Technical Report 14/75, Institut für Numerische Mathematik, T.U. Wien, 1975.

[18] R. Frank, C.W. Ueberhuber, Iterated defect correction for differential equations, Part I: Theoretical results, Computing 20 (1978) 207–228.

[19] C.W. Gear, D.S. Watanabe, Stability and convergence of variable order multistep methods, SIAM J. Numer. Anal. 11 (1974) 1024–1043.

[20] W.B. Gragg, Repeated extrapolation to the limit in the numerical solution of ordinary differential equation, Ph.D. Thesis, University of California, 1964. See also SIAM J. Numer. Anal. Ser. B 2 (1965) 384–403.

[21] E. Hairer, On the order of iterated defect correction – an algebraic proof, Numer. Math. 29 (1978) 409–424.

[22] E. Hairer, C. Lubich, Asymptotic expansions of the global error of fixed-stepsize methods, Numer. Math. 45 (1984) 345–360.

[23] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations I Nonstiff Problems, Springer, Berlin, 1980.

[24] J.K. Hale, Ordinary Differential Equations, Wiley, New York, 1969.

[25] P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York, 1962.

[26] D.J. Higham, Global error versus tolerance for explicit Runge–Kutta methods, IMA J. Numer. Anal. (1991) 457–480.

[27] T.E. Hull, W.H. Enright, B.M. Fellen, A.E. Sedgwick, Comparing numerical methods for ordinary differential equations, SIAM J. Numer. Anal. 9(4) (1972) 603–637.

[28] P.J. Peterson, Global error estimation using defect correction techniques for explicit Runge–Kutta methods, Tech. Rep. 192/86, Departement of Computer Science, University of Toronto, Canada, 1986.

[29] A. Prothero, Estimating the accuracy of numerical solutions to ordinary differential equations, in: I. Gladwell, D.K. Sayers (Eds.), Computational Techniques for Ordinary Differential Equations, Academic Press, London, 1980, pp. 103–128.

[30] L.F. Shampine, Asymptotic bounds on the errors of one-step methods, Numer. Math. 45 (1984) 201–206.

[31] L.F. Shampine, M.K. Gordon, Solving ordinary differential equations with ode, step and intrp, Technical Report SLA-73-1060, Sandia Laboratories, 1973.

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

[33] L.F. Shampine, W. Zhang, Rate of convergence of multistep codes started by variation of order and stepsize, SIAM J. Numer. Anal. 27 (6) (1990) 1506–1518.

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

[35] H.J. Stetter, The defect correction principle and discretisation methods, Numer. Math. 29 (1978) 425–443.

[36] H.J. Stetter, Global error estimation in ode-solvers, in: G.A. Watson (Ed.), Numerical Analysis, Lecture Notes in Math. vol. 630, Springer, Berlin, 1978.

[37] H.J. Stetter, Global error estimation in Adams PC-codes, ACM Trans. on Math. Software 5 (4) (1979) 415–430.

[38] J.F. Vernotte, P. Panciatici, B. Meyer, J.P. Antoine, M. Stubbe, High fidelity simulation of power system dynamics, Comput. Appl. in Power (1995) 37–41.

[39] P.E. Zadunaisky, On the estimation of error propagated in the numerical solution of a system of ordinary differential equations, Numer. Math. 27 (1976) 21–39.