Two–step error estimators for implicit Runge–Kutta methods applied to stiff systems

S. GONZÁLEZ–PINTO
Universidad de La Laguna, Spain
J.I. MONTIJANO
Universidad de Zaragoza, Spain
and S. PÉREZ–RODRÍGUEZ
Universidad de La Laguna, Spain

This paper is concerned with the local error estimation in the numerical integration of stiff systems of ordinary differential equations by means of Runge–Kutta methods. With implicit Runge–Kutta methods it is often difficult to embed a local error estimate with the appropriate order and stability properties. In this paper a local error estimation based on the information of the last two integration steps (that are supposed to have the same steplength) is proposed. It is shown that this technique, applied to Radau IIA methods, let us get estimators with proper order and stability properties. Numerical examples showing that the proposed estimation improves the efficiency of the integration codes are presented.

Categories and Subject Descriptors: G.1.7 [**Numerical Analysis**]: Ordinary Differential equations

Additional Key Words and Phrases: Error estimation, Stiff Initial Value Problems, Implicit Runge-Kutta methods, Runge–Kutta

1. INTRODUCTION.

Let us consider a stiff system of ordinary differential equations

$$y' = f(t, y), \quad y(0) = y_0 \in \mathbb{R}^m, \quad t \geq 0.$$

Adaptive codes for the numerical solution of differential equations usually control the integration step so that a local error estimate is maintained below a given error

This work was supported by project DGI BFM2001–2562 and by project 1802010205 Universidad de La Laguna.
Name: S. González–Pinto and S. Pérez–Rodríguez
Address: Dpto. de Matemática Aplicada, Universidad de La Laguna, 38271–La Laguna, Spain
Name: J.I. Montijano
Address: Dpto. de Matemática Aplicada, Universidad de Zaragoza, 50009–Zaragoza, Spain

tolerance. With Runge–Kutta methods

$$\begin{aligned} &Y_{n,i} = y_n + h \textstyle\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{n,j}) \quad (i = 1, \ldots, s), \\ &y_{n+1} = y_n + h \textstyle\sum_{i=1}^{s} b_i f(t_n + c_i h, Y_{n,i}), \end{aligned} \tag{1}$$

the estimator is commonly based on an embedded formula

$$\hat{y}_{n+1} = y_n + h \sum_{i=1}^{s} \hat{b}_i f(t_n + c_i h, Y_{n,i}), \tag{2}$$

where the approximations y_{n+1} and $\hat{y}_{n+1}$ have different orders and the estimator is then given by the difference $Est = y_{n+1} - \hat{y}_{n+1}$.

When the RK method is fully implicit with high order of accuracy, the natural embedded formula has its order seriously limited unless some additional information is supplied. Thus, for example, if we consider the fifth–order Radau IIA method, since it has three internal stages, any embedded formula (2) can have at most order 2 and this implies that the stepsize will be controlled in practice by the error of the formula of order 2. In general, for s-stage collocation methods such as Radau IIA, Gauss, Lobatto IIIC or Lobatto IIIA, the embedded methods can attain at most order $s-1$.

With these type of estimators, the stepsize is controlled by the error of a method of order $s-1$, while the error of the advancing solution can have order $2s$, $2s-1$, $2s-2, \ldots$. Then, as the tolerance gets smaller, the lower order error estimate gives an overestimated error which makes the code take steps smaller than those required by the advancing formula. In the case of explicit methods such a situation will result in an over accurate solution, with higher cost, as if we had reduced the error tolerance. However, with implicit methods, the effect of this overestimated error is a bit more complicated.

In implicit methods applied to stiff systems, the stage equations are solved by some Newton type iterative scheme that gives successive approximations $Y_{n,i}^{(k)}$ to the internal stages $Y_{n,i}$. The scheme is typically stopped when the distance between two consecutive approximations satisfies

$$\|Y_{n,i}^{(k)} - Y_{n,i}^{(k-1)}\| \leq c\, Tol, \quad i = 1, \ldots, s.$$

The constant c is a safety coefficient that preserve the final solution of being badly affected by the error in the solution of the implicit equations, and for example in RADAU5 code (see [7], Chap. IV.8) takes a value ranging from 0.1 to 0.01. The approximation y_{n+1} and the estimator are computed from these last values $Y_{n,i}^{(k)}$ and the step is accepted if

$$\| Est \| \leq Tol.$$

Let us suppose now that the estimator is overestimating the actual error, that is, $\| Est \| >> Error$. This makes the code take stepsizes smaller than those really needed to get $Error \leq Tol$. However, these smaller stepsizes might not imply smaller final errors because these errors are also affected by the error in the iterations. Thus, close to $c\, Tol$ final errors are expected independently of the stepsize taken.

In RADAU5, Hairer and Wanner [7], Chap. IV.8, consider an embedded method of the form (a similar error estimate is used in [8] for higher order Radau IIA methods)

$$\hat{y}_{n+1} = y_n + h \sum_{i=1}^{3} \hat{b}_i f(t_n + c_i h, Y_{n,i}) + h b_0 f(t_n, y_n),$$

that can have order three (observe that $f(t_n, y_n)$ does not correspond to any of the three stages of the method). However, this approach is unsatisfactory because there is not enough information to form a result of order more comparable to the basic formula. Moreover, the stability of the embedded formula is not satisfactory, so they have to correct the estimator as proposed by Shampine and Baca [9], using finally

$$Est = (I - h\gamma J)^{-1}(y_{n+1} - \hat{y}_{n+1}).$$

The matrix $(I - h\gamma J)^{-1}$ is available and factored from the Newton iteration used to solve the internal stages $Y_{n,i}$ and the estimator only requires the solution of the corresponding triangular linear systems.

De Swart and Söderlind in [10] propose an improved estimator, theoretically justified,

$$Est = (I - h\gamma J)^{-1} h \left(\sum_{i=1}^{s} (b_i - \hat{b}_i) f(t_n + c_i h_n, Y_i) - \gamma f(t_{n+1}, y_{n+1}) - \hat{b}_0 f(t_n, y_n) \right)$$

where the coefficient $\hat{b}_0$ is chosen so that the error estimate does not underestimate the actual error in some particular conditions. For the three-stage Radau IIA formula this error estimate is similar to the one by Hairer and Wanner [7] except by the choice of the free parameter $\hat{b}_0$. The authors show that this new estimator provides a better agreement between actual and estimated errors.

In both cases the estimators have some limitations. On one hand, they require the existence of a previously factored matrix $(I - h\gamma J)$. In the case of Radau IIA with 4 stages, the matrix A does not have real eigenvalues and the solution of the implicit stage equations can be reduced to two block of complex systems. Then, there are not any pre–factored matrix to be used in the estimator. On the other hand, the estimators are also based on the additional evaluation $f(t_n, y_n)$. However there are methods such as Lobatto IIIA for which $f(t_n, y_n)$ is in fact one of the stages and therefore this kind of error estimation gives no advantage.

A classic technique for estimating the local error is the extrapolation technique. It consists in giving, after two consecutive steps with the same stepsize, a double step. Then, the error is estimated by a linear combination of the two approximations to the solution at t_{n+2}. This technique gives very good results, but has the inconvenient of the computational cost implied by the double step (the matrix factorization plus the solution of some triangular linear systems per iteration).

In this paper we consider the construction of local error estimators based, like extrapolation, on two consecutive steps, using the information along these two steps to get an embedded formula with order as high as possible. Thus, let us suppose that we have advanced from $t_n \to t_{n+1} = t_n + h$ and next from $t_{n+1} \to t_{n+2} = t_{n+1} + h$. We will consider an embedded formula of the following type

$$\hat{y}_{n+2} = y_n + h \sum_{j=1}^{s} \left[\beta_j f(t_n + c_j h, Y_{n,j}) + \gamma_j f(t_{n+1} + c_j h, Y_{n+1,j}) \right], \tag{3}$$

where β_j and γ_j are constants that will be determined taking into account the accuracy and stability of $\hat{y}_{n+2}$. This estimator has practically null computational cost. It is important to notice that in practice the evaluation of derivative functions in (3) can be avoided by replacing the derivative functions through the Runge–Kutta stage equations, obtaining a more stable formula for computational purposes.

We must mention that the proposed approach has some disadvantages. Thus, a failed step implies that we must reject two steps. On the other hand, since the code is forced to take two consecutive steps with the same size, the stepsize is adjusted less often, and this can imply a losing of efficiency in some circumstances. These drawbacks can be overcame by extending our technique so that an error estimate can be obtained from two consecutive, non equal, steps. In this case the estimator will depend on the stepsize ratio and some modification to the stepsize change formula must be done. Some research in this direction is being carried out, but in this paper, for simplicity, we have limited our study to the case of two equal consecutive steps.

Error estimators based on two or more consecutive steps have been also considered in [4] and [3] for explicit methods and in [1] for Singly–Implicit Runge–Kutta methods.

Next, we briefly outline the rest of the paper. In section 2 we give some order results for two–step Runge–Kutta methods. In section 3 we obtain two–step error estimators for Radau IIA methods with 3, 4 and 5 stages and finally in section 4 we present some numerical experiments showing the performance of the proposed estimates.

2. ORDER RESULTS FOR TWO–STEP EMBEDDED METHODS.

In this section we study the maximum attainable order for the embedded methods introduced in the previous section, based on the information of two consecutive steps.

2.1 General order results

Let us suppose that the Runge–Kutta method satisfies the simplifying conditions

$$\begin{array}{lll} C(q): & Ac^{j-1} = c^j/j, & j = 1\dots,q \\ B(q): & b^T c^{j-1} = 1/j, & j = 1\dots,q \end{array} \tag{4}$$

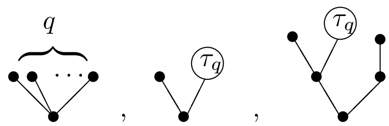
Let T denote the set of unlabeled rooted trees, $\tau \in T$ a rooted tree and τ_0 the tree with only one vertex (the only tree with order 1). Following the notations in [2], a tree τ with order $\rho(\tau) > 1$ can be written $\tau = [\tau'_1, \dots, \tau'_k]$ where τ'_i are the subtrees of τ that attached to the root give τ.

It is easy to see that the method reaches order $q+1$ if and only if the order condition for the tree $\tau_q = [\tau_0, \dots, \tau_0] = [\tau_0^q]$ is fulfilled. Then, order $q+2$ is attained if in addition the order conditions associated to the trees τ_{q+1} and $[\tau_q]$ are satisfied. The order $q+3$ demands the order conditions on these trees and also on τ_{q+2}, $[\tau_{q+1}]$, $[\tau_q, \tau_0]$ and $[[\tau_q]]$. We are interested in characterizing the subset ST_q of the tree set T such that for any $r \geq 1$

$$\text{order } q+r \Longleftrightarrow b^T\Phi(\tau) = 1/\gamma(\tau), \forall \tau \in ST_q \text{ with } \rho(\tau) \leq q+r$$

Definition 1. We will say that τ' is a descendant of τ if $\tau' \equiv \tau$ or well τ' is a subtree of τ or it is a subtree of a subtree of τ and so on.

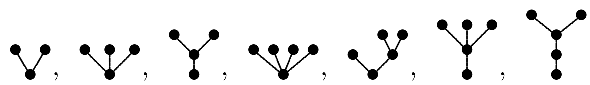
Example 1. The tree τ_q is a descendant of each of the following trees:

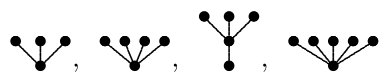


Let us denote $ST_1 = T$ and for $q \geq 2$:

$ST_q = \{\tau \in T \setminus \{\tau_0\},\ \tau$ has no descendants of type $\tau^* = [\tau_0^k],\ 1 \leq k \leq q-1\}$

Example 2. For the first values of q the sets ST_q are:

$ST_2 = \{$  $, \dots\}$

$ST_3 = \{$  $, \dots\}$

$ST_4 = \{$  $, \dots\}$

With this definition it is trivial the following

LEMMA 1. *$ST_n \subsetneqq ST_m$ for all $n \geq m \geq 1$.*

Denoting now $ST_q^r = \{\tau \in ST_q, \rho(\tau) \leq q + r\}$, we have the following

THEOREM 1. *A Runge–Kutta method satisfying (4) reaches order $q + r$ if and only if it satisfies the order condition for every tree in ST_q^r, i.e.,*

$$b^T \Phi(\tau) = 1/\gamma(\tau), \quad \forall\, \tau \in ST_q^r \tag{5}$$

PROOF. If $q = 1$ or $r = 1$ the proof is trivial. Let $\tau \notin ST_q$ with $q+1 \leq \rho(\tau) \leq q+r$. It has at least a descendant $\tau' = [\tau_0, \dots, \tau_0]$ with order $p = \rho(\tau') \leq q$. If $\tau = \tau'$, the proof is trivial by $C(q)$ and $B(q)$. Otherwise, τ must have a descendant of the form $\hat{\tau} = [\tau', \tau_1', \dots, \tau_k'],\ k \geq 0$. Now, by $C(q)$ the order condition for τ is satisfied if it is satisfied the order condition corresponding to the tree τ^*, which has the same order as τ obtained by replacing in τ the descendant $\hat{\tau}$ by $\bar{\tau} = [\tau_0^{p+1}, \tau_1', \dots, \tau_k']$. If $\tau^* \in ST_q^r$, the proof is concluded. Otherwise we can repeat the process until we arrive at a tree in ST_q^r. $\square$

2.2 Order results for two–step Runge–Kutta methods

Let us consider an s–stages Runge–Kutta method (A, b) fulfilling (4) and with order $p = q + r$. Let us also denote by $(\hat{A}, \hat{b})$ the coefficients of the composed method resulting after two consecutive steps of size h, given by the Butcher tableau

$$\begin{array}{c|c} \hat{c} & \hat{A} \\ \hline & \hat{b}^\top \end{array} \quad = \quad \begin{array}{c|cc} c & A & 0 \\ e+c & eb^\top & A \\ \hline & b^\top & b^\top \end{array},$$

$\hat{e}^\top = (e^\top, e^\top)$ and by $\hat{\Phi}(\tau)$ the $2s-$dimensional tree functions associated to matrix $\hat{A}$.

We are interested in obtaining the families of embedded methods (3), given by the coefficients $(\hat{A}, \hat{\beta})$, with $\hat{\beta}^\top = (\beta^\top, \gamma^\top) = (\beta_1, \ldots, \beta_s, \gamma_1, \ldots, \gamma_s)$, with orders $q+l$ for $l = 1, \ldots, r$. Since the composed method also satisfies $C(q)$, $\hat{A}\hat{c}^{j-1} = \hat{c}^j/j$, $j = 1, \ldots, q$, we can apply Theorem 1 to the embedded method, replacing the order conditions (5) by

$$\hat{\beta}^\top \hat{\Phi}(\tau) = 2^{\rho(\tau)}/\gamma(\tau), \quad \forall\, \tau \in ST_q^r.$$

THEOREM 2. *For every $k = 0, \ldots, r$ there exist a family of embedded methods $(\hat{A}, \hat{\beta})$ of order $q+k$ with $l_k = 2s-$ rank$(M_{q,k})$ free parameters, being $M_{q,k}$ the matrix whose columns are the vectors $\hat{e}, \hat{c}, \ldots, \hat{c}^{q-1}$ and the vectors $\hat{\Phi}(\tau)$ with $\tau \in ST_q^k$.*

PROOF. Order $q+k$ is equivalent to $\hat{\beta}^\top \hat{c}^{j-1} = 2^{j-1}/j$, for $j = 1, \ldots, q$ and $\hat{\beta}^T \hat{\Phi}(\tau) = 2^{\rho(\tau)}/\gamma(\tau)$ for all $\tau \in ST_q^k$ and they are equivalent to the linear system

$$\hat{\beta}^T M_{q,k} = (2, 2^2/2, \ldots, 2^q/q, 2^{\rho(\tau)}/\gamma(\tau))\ \forall \tau \in ST_q^k.$$

Since (b^T, b^T) is always a solution of the above linear system, then it is compatible and the number l_k of free parameters in the solution $\hat{\beta}$ are determined by the Rouché–Fröbenius theorem which give us $l_k = 2s - \text{rank}(M_{q,k})$. □

2.3 Particular cases

For three–stage collocation methods of order ≥ 5 ($s = 3$, $q = 3$) such as Radau IIA and Gauss,

—If $k = 1$, $M_{s,1} = [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3]$ and rank$(M_{s,1}) = 4$. Then there exist a biparametric family of embedded methods with order 4.

—If $k = 2$, $M_{s,2} = [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3, \hat{c}^4, \hat{A}\hat{c}^3]$ and rank$(M_{s,2}) = 6$. Then there is a unique method of order ≥ 5 (the composed method) and therefore there is not any embedded method with order 5.

For four–stage collocation methods of order ≥ 7 ($s = 4, q = 4$)

—If $k = 1$, $M_{s,1} = [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3, \hat{c}^4]$ and rank$(M_{s,1}) = 5$. Then there exist a three–parameter family of embedded methods with order 5.

—If $k = 2$, $M_{s,2} = [\hat{e}, \hat{c}, \ldots, \hat{c}^5, \hat{A}\hat{c}^4]$ and rank$(M_{s,2}) = 7$. Then there is a one–parameter family of embedded methods of order ≥ 6.

—If $k = 3$, $M_{s,3} = [\hat{e}, \hat{c}, \ldots, \hat{c}^6, \hat{A}\hat{c}^4, \hat{A}\hat{c}^4 \cdot \hat{c}, \hat{A}\hat{c}^5, \hat{A}^2\hat{c}^4]$ and rank$(M_{s,3}) = 8$. The only method satisfying these equations, and therefore with order ≥ 7, is the composed method.

For five–stage collocation methods of order ≥ 9 ($s = 5, q = 5$)

—If $k = 1$, $M_{s,1} = [\hat{e}, \hat{c}, \ldots, \hat{c}^5]$ and $\text{rank}(M_{s,1}) = 6$. Then there exist a four–parameter family of embedded methods with order 6.
—If $k = 2$, $M_{s,2} = [\hat{e}, \hat{c}, \ldots, \hat{c}^6, \hat{A}\hat{c}^5]$ and $\text{rank}(M_{s,2}) = 8$. Then there is a two–parameter family of embedded methods of order ≥ 7.
—If $k = 3$, $M_{s,3} = [\hat{e}, \hat{c}, \ldots, \hat{c}^7, \hat{A}\hat{c}^5, \hat{A}\hat{c}^5 \cdot \hat{c}, \hat{A}\hat{c}^6, \hat{A}^2\hat{c}^5]$ and $\text{rank}(M_{s,3}) = 10$. The only method satisfying these equations, and therefore with order ≥ 8, is the composed method.

3. EMBEDDED FORMULAS FOR RADAU IIA METHODS.

In this section we present the construction of embedded methods for Radau IIA methods with 3, 4 and 5 stages, taking into account not only the accuracy of the formulas but also the stability properties so that the error estimate can fit the actual error as closely as possible.

When a two–step method $(\hat{A}, \hat{\beta})$ is applied to the linear scalar test equation $y' = \lambda y$, we have

$$\hat{y}_{n+2} = \hat{R}(h\lambda) y_n$$

where $\hat{R}(z)$ is the amplifying function of the method given by

$$\hat{R}(z) = 1 + z\hat{\beta}^T (I - z\hat{A})^{-1} \hat{e}.$$

In particular, for the composed method $(\hat{A}, \hat{b})$, it is clear that $\hat{R}(z) = R^2(z)$, being $R(z) = P(z)/Q(z)$ the stability function of the base method.

For any embedded method $(\hat{A}, \hat{\beta})$ of order p we have that $\hat{R}(z) = \hat{P}(z)/\hat{Q}(z)$ is a rational function of degree $2s$. Moreover its denominator is $\hat{Q}(z) = \det(I - z\hat{A}) = \det(I - zA)^2 = Q^2(z)$. Here we have denoted by I the identity matrix of appropriate dimension.

Since $\hat{y}_{n+2}$ approximates the local solution $y(t_{n+2}; t_n, y_n) = e^{2h\lambda} y_n$ up to order p, then

$$\hat{R}(z) = 1 + 2z + \ldots + 2^p \frac{z^p}{p!} + z^{p+1} \hat{\beta}^T \hat{A}^{p-1} \hat{c} + \ldots + z^{2s} \hat{\beta}^T \hat{A}^{2s-2} \hat{c} + \mathcal{O}(z^{2s+1}) \quad (6)$$

and since we know the denominator of $\hat{R}(z)$, the free parameters in the numerator $\hat{P}(z)$ can be determined so that the embedded formula has adequate stability properties.

To estimate properly the error in the stiff components it is desirable that $|\hat{R}(z)|$ is not large in the complex region $\text{Re}\lambda \leq 0$. In particular it is important that the value $|\hat{R}(\infty)| = |1 - \hat{\beta}^T \hat{A}^{-1} \hat{e}|$ is similar to the corresponding value $|R(\infty)|^2$ for the composed method. Therefore, in the case of Radau IIA methods, we will search for embedded methods for which $\hat{R}(\infty) = 0$ so that they are stiffly stable, like the underlying formula, and so there is no need of "filtering" in the estimator.

3.1 Radau IIA with 3 stages

In this case it is possible to embed a method with order 4 and we have two free parameters to choose the method having the most adequate stability properties.

Since

$$\hat{Q}(z) = \left(1 - \frac{3}{5}z + \frac{3}{20}z^2 - \frac{1}{60}z^3\right)^2$$

then from (6) with $p=4$,

$$\hat{P}(z)=\hat{R}(z)\hat{Q}(z)=1+\frac{4}{5}z+\frac{13}{50}z^2+\frac{1}{25}z^3+\frac{1}{400}z^4+uz^5+vz^6$$

with $u=\hat{\beta}^T\hat{A}^3\hat{c}-\frac{2^5}{5!}$ and $v=\hat{\beta}^T\hat{A}^4\hat{c}-\frac{6}{5}u-\frac{107}{1200}$.

Note that $\hat{R}(z)$ and $R^2(z)$ coincide up to terms of order z^4 and since $R(\infty)=0$, then

$$\hat{R}(z)=R^2(z)+\frac{uz^5+vz^6}{\hat{Q}(z)}.$$

We can choose the free parameters in such a way that $\hat{R}(\infty)=0$. It is enough to impose that $v=0$, and this is achieved when

$$\hat{\beta}^T(\hat{A}^4\hat{c}-\frac{6}{5}\hat{A}^3\hat{c})=-\frac{277}{1200}.$$

Then, the vector of coefficients of such a method of order 4 must satisfy the linear system

$$\hat{\beta}^T[\hat{e},\hat{c},\hat{c}^2,\hat{c}^3,\hat{v},\hat{w}]=\left(2,\frac{2^2}{2},\frac{2^3}{3},\frac{2^4}{4},-\frac{277}{1200},u+\frac{2^5}{5!}\right)$$

being $\hat{v}=\hat{A}^4\hat{c}-\frac{6}{5}\hat{A}^3\hat{c}$. and $\hat{w}=\hat{A}^3\hat{c}$. This system has a unique solution and hence the coefficients of the estimator $(b^T,b^T)-\hat{\beta}^T$ can be expressed in terms of the free parameter u as

$$(b^T,b^T)-\hat{\beta}^T=u\frac{4}{5}\left(19-14\sqrt{6},19+14\sqrt{6},52,-29-51\sqrt{6},-29+51\sqrt{6},-32\right).$$

Let us observe that we cannot also impose $u=0$, because the embedded method would coincide with the composed method.

We can now investigate for which values of the available free parameter u the embedded method will be A–stable. This is accomplished if $|\hat{R}(iy)|^2\le 1$ or equivalently if $|\hat{P}(iy)|^2-|\hat{Q}(iy)|^2\le 0$ for all $y\in\mathbb{R}$. After some calculations, it can be seen that

$$|\hat{P}(iy)|^2-|\hat{Q}(iy)|^2=\left(\frac{8}{5}u-\frac{1}{1800}\right)y^6-\left(\frac{2}{25}u+\frac{1}{30000}\right)y^8+\left(u^2-\frac{1}{720000}\right)y^{10}-\frac{1}{12960000}y^{12}$$

and the method is A–stable if and only if

$$\frac{1}{12960000}x^3-(u^2-\frac{1}{720000})x^2+(\frac{1}{30000}+\frac{2}{25}u)x+\frac{1}{1800}-\frac{8}{5}u\ge 0,\quad \text{for all } x\ge 0.$$

This condition is satisfied for all $u\in[-0.001904604018365...,\frac{1}{2880}]$.

Note that, since the integration is advanced with the Radau IIA formula, instabilities on the numerical solution provided by the embedded method will not be propagated, hence A–stability for the embedded formula is not necessary. Thus, the parameter u can be chosen outside the above stability interval. However, we will get a nice value for u (below) which belongs to this interval.

In order to choose a suitable value of the free parameter u we will consider the linear test equation $y' = \lambda y$ and following the ideas in [10] we will consider the relative local error. However in this case, since our estimators are based on the computations of the two last integration steps we will consider the relative local error along two consecutive steps with the same stepsize

$$Err(z) = \left| \frac{y(t_{n+2}; t_n, y_n) - y_{n+2}}{y_n} \right| = |e^{2z} - R^2(z)|.$$

being $z = h\lambda$. This error is estimated with the proposed two–step estimator by the quantity

$$Est_{TS}(z) = |R^2(z) - \hat{R}(z)|,$$

while the estimate provided by extrapolation is

$$Est_{ext}(z) = |R^2(z) - R(2z)|/31.$$

Also, the estimator used by Hairer and Wanner in RADAU5 [7] gives

$$Est_{HW}(z) = |R(z)(R(z) - \tilde{R}(z))/(1 - z\gamma)|$$

with $\tilde{R}(z)$ the amplifying function of the embedded third–order method and γ the only real eigenvalue of the matrix A (the estimator proposed by de Swart and Söderlind in [10] gives the same error function of z scaled by a factor of $1/(50\gamma)$).

For Radau IIA method of three stages these functions of z can be easily computed giving

$$Est_{TS}(z) = \left| \frac{u z^5}{\left(1 - \frac{3}{5}z + \frac{3}{20}z^2 - \frac{1}{60}z^3\right)^2} \right|$$

$$Est_{ext}(z) = \left| \frac{z^6 \left(1 - \frac{8}{155}z + \frac{1}{155}z^2\right)}{3600 \left(1 - \frac{3}{5}z + \frac{3}{20}z^2 - \frac{1}{60}z^3\right)^2 \left(1 - \frac{6}{5}z + \frac{3}{5}z^2 - \frac{2}{15}z^3\right)} \right|$$

$$Est_{HW}(z) = \left| \frac{\gamma \left(1 + \frac{2}{5}z + \frac{1}{20}z^2\right) z^4}{60(1 - z\gamma) \left(1 - \frac{3}{5}z + \frac{3}{20}z^2 - \frac{1}{60}z^3\right)^2} \right|.$$

Concerning the selection of the parameter u, we must note that the estimator is a linear homogeneous function of u and therefore changing u will imply just a scaling of the estimate. It is important however that the estimator does not underestimate the real error as pointed out in [10]. We have selected the value $u = 0.0000529585077373525889677785167637$, for which

$$Err(x) \le Est_{TS}(x), \quad \forall x \le x_0 = -2.605...$$
$$Err(x) - Est_{TS}(x) \le 8.3 \cdot 10^{-5}, \quad Err(x) \le 1.96 \cdot Est_{TS}(x), \quad \forall x \in (x_0, 0].$$

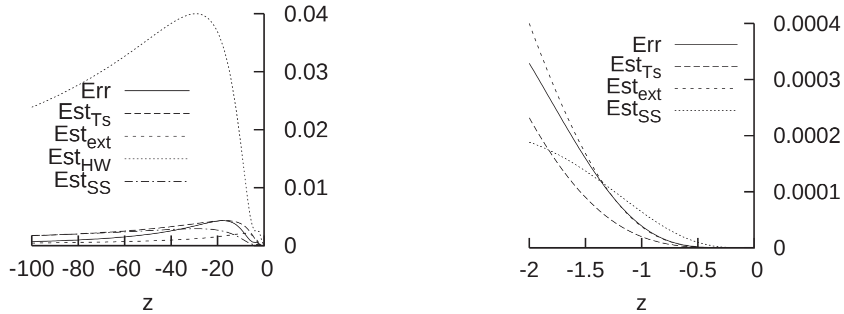


Fig. 1. Estimators on the negative real axis

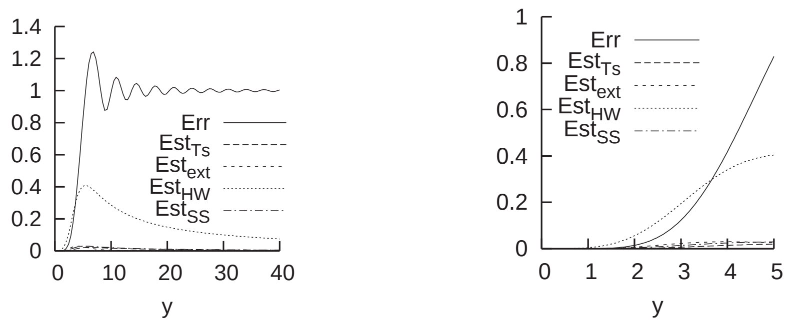


Fig. 2. Estimators on the imaginary axis

That is, the estimated error is greater than the actual error along the negative real axis except in the interval $(x_0, 0)$ where the actual error is very close to the estimated one. Moreover, without the absolute value, neither *Err(x)* nor Est_{TS} change their signs on the negative real axis, which has some importance in linear problems of high dimension.

In figure 1 we have plotted the functions $Est_{TS}(z)$, $Est_{ext}(z)$, $Est_{HW}(z)$ and the corresponding one for the estimator proposed by de Swart and Söderlind [10], $Est_{SS}(z) = Est_{HW}(z)/(50\gamma)$, together with the local error $Err(z)$ for real values of z ranging from -100 to 0 (left side of the figure), and from -2 to 0 (right side of the figure). As it can be seen, the new estimator gives values that are closer to the true local error than those provided by the estimator in RADAU5, and similar to the ones given by the estimator by de Swart and Söderlind. In figure 2 we have plotted the estimate functions taking values of z along the imaginary axis, $z = iy$, with y ranging from 0 to 40 (left hand side) and from 0 to 5 (right hand side).

3.2 Radau IIA with 4 stages

In this case it is possible to embed a method with order 6 and we have one free parameter to choose the method having the most adequate stability properties.

Since $\hat{R}(z)$ and $R^2(z)$ coincide up to terms of order z^6 and $R(\infty) = 0$ we have that,

$$\hat{R}(z) = R^2(z) + \frac{uz^7 + vz^8}{Q^2(z)}$$

with $u = \beta^T \hat{A}^5 \hat{c} - 2^7/7!$ and $v = \beta^T \hat{A}^6 \hat{c} - \frac{8}{7}u - \frac{1493}{235200}$. Moreover,

$$\hat{Q}(z) = Q^2(z) = \left(1 - \frac{4}{7}z + \frac{1}{7}z^2 - \frac{2}{105}z^3 + \frac{1}{840}z^4\right)^2,$$

and

$$\hat{P}(z) = P^2(z) + uz^7 + vz^8 = \left(1 + \frac{3}{7}z + \frac{1}{14}z^2 + \frac{1}{210}z^3\right)^2 + uz^7 + vz^8.$$

Here v must be non zero to have an embedded method (different from the composed method) of order 6 exactly. Then $\hat{R}(\infty) = v/750600$ while $R^2(z) = \mathcal{O}(1/z^2)$ and the error in the stiff components can be overestimated.

We can also search for an embedded method of order 5 with $\hat{R}(\infty) = 0$. By imposing the order five conditions

$$\hat{\beta}^T[\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3, \hat{c}^4] = \left(2, \frac{2^2}{2}, \frac{2^3}{3}, \frac{2^4}{4}, \frac{2^5}{5}\right)$$

we get a three-parameter family of methods of order 5. For these methods the amplifying function $\hat{R}(z) = \hat{P}(z)/\hat{Q}(z)$ is given by $\hat{Q}(z) = Q^2(z)$ and

$$\hat{P}(z) = P^2(z) + wz^6 + uz^7 + vz^8$$

with

$$w = \hat{\beta}^T \hat{A}^4 \hat{c} - 2^6/6!$$

$$u = \hat{\beta}^T \hat{A}^5 \hat{c} - \frac{8}{7}w - \frac{8}{315}$$

$$v = \hat{\beta}^T \hat{A}^6 \hat{c} - \frac{34}{49}w - \frac{8}{7}u - \frac{1493}{235200}$$

Clearly, $\hat{R}(\infty) = 0$ if and only if $v = 0$, that is,

$$\hat{\beta}^T \hat{A}^6 \hat{c} = \frac{34}{49}w + \frac{8}{7}u + \frac{1493}{235200}.$$

One possibility for choosing the free parameters w, u is to make the main term of the local error of the embedded method proportional to the sixth derivative of the solution, that is

$$\sum_{\substack{\tau \in T \\ \rho(\tau)=6}} \alpha(\tau)(2^6 - \gamma(\tau)\hat{\beta}^T \hat{\Phi}(\tau))F(\tau)(y_0) = K \sum_{\substack{\tau \in T \\ \rho(\tau)=6}} \alpha(\tau)F(\tau)(y_0) = K6!y^{(6)}(t_0)$$

for some constant K. Taking into account the simplifying assumptions, this is accomplished if and only if $\hat{\beta}^T(\hat{c}^5 - 5\hat{A}\hat{c}^4) = 0$ and since $\hat{A}^6\hat{c} = \hat{A}^3\hat{c}^4/4!$, $\hat{A}^5\hat{c} = \hat{A}^2\hat{c}^4/4!$ and $\hat{A}^4\hat{c} = \hat{A}\hat{c}^4/4!$, the family of embedded methods satisfying the required conditions is given by

$$\hat{\beta}^T \left[\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3, \hat{c}^4, \hat{c}^5, \hat{A}\hat{c}^4, \quad \left(\hat{A}^3 - \frac{8}{7}\hat{A}^2\right)\hat{c}^4\right] = \left(2, \frac{2^2}{2}, \frac{2^3}{3}, \frac{2^4}{4}, \frac{2^5}{5}, \frac{32}{3} - \frac{K}{6}, \frac{32}{15} - \frac{K}{30}, -\frac{16001}{29400} + \frac{K}{49}\right)$$

being K a free parameter. For $K = 0$ we get the composed seventh–order method and the estimate is not valid.

We have followed for the selection of the parameter K a similar approach to that used for the three–stage method. In this case we have determined numerically the values of K for which the embedded method is A–stable, obtaining the interval $[-0.00733464..., 0]$. For $K = -0.0010147077654953154726580139519\,3$ the estimated error is greater than the actual error along the negative real axis.

3.3 Radau IIA with 5 stages

In this case it is possible to embed a method with order 7 and we have two free parameters. In these conditions

$$Q(z) = 1 - \frac{5}{9}z + \frac{5}{36}z^2 - \frac{5}{252}z^3 + \frac{5}{3024}z^4 - \frac{1}{15120}z^5,$$

$$P(z) = 1 + \frac{4}{9}z + \frac{1}{12}z^2 + \frac{1}{126}z^3 + \frac{1}{3024}z^4$$

and

$$\hat{R}(z) = R^2(z) + \frac{wz^8 + vz^9 + uz^{10}}{Q^2(z)}$$

with

$$w = \hat{\beta}^T \hat{A}^6 \hat{c} - 2^8/8!$$

$$v = \hat{\beta}^T \hat{A}^7 \hat{c} - \frac{10}{9}\hat{\beta}^T \hat{A}^6 \hat{c} - \frac{16}{2835}$$

$$u = \hat{\beta}^T \hat{A}^8 \hat{c} - \frac{10}{9}\hat{\beta}^T \hat{A}^7 \hat{c} + \frac{95}{162}\hat{\beta}^T \hat{A}^6 \hat{c} - \frac{185771}{76204800}$$

We can select the free parameters so that $u = 0$ and then, the one–parameter family of embedded methods is determined by the eight conditions of order 7 and the condition $\hat{\beta}^T \left(\hat{A}^8\hat{c} - \frac{10}{9}\hat{A}^7\hat{c} + \frac{95}{162}\hat{A}^6\hat{c}\right) = \frac{185771}{76204800}$. The free parameter can again be selected taking into account the stability of the embedded method and experimental experience.

4. NUMERICAL EXPERIMENTS.

In this section we present some numerical experiments showing the performance of the proposed error estimate for the three-stage Radau IIA method.

In order to compare the different error estimates we have developed a code, not intended to compete with standard codes, but rather to asses the performance of several estimators when they work in similar conditions. The code can control the stepsize by means of the following local error estimates:

Ext – Local extrapolation.

HW – The Hairer and Wanner one–step estimator used in RADAU5.

HW2 – The HW estimator used in a two–step mode.
Here, the code is forced to take two consecutive steps with the same steplength. If the error test is not satisfied in one of these two consecutive steps, both are rejected. This estimator will help us to compare HW with the proposed two–step one.

SS – The one–step estimator by de Swart and Söderlind.

TS – The proposed two–step error estimator.

The stage equations are solved by a modified Newton scheme, and the iterations are stopped when two consecutive approximations differ less than $Tol/100$. The Jacobian matrix is computed, and the corresponding matrix factored, at each step when one-step estimators are used, whereas for two-step estimators the LU factorization carried out on the first step (of each pair) is used on the second step and there is no need of recomputing the Jacobian matrix (that is one of the advantages of using two step-estimators in the way proposed in this paper). In a production code this strategy could be improved monitoring the rate of convergence on the first step and if it is not adequate, forming a new Jacobian for the second step. This could improve the efficiency by reducing the number of convergence failures or well reducing the number of iterations on the second step. Nevertheless we have considered such an strategy unnecessary for our experiments.

We have integrated a number of stiff problems, including those of the well known DETEST package [5] and we present here the results obtained with the following two problems:

Problem 1.- The Van der Pol oscillator (see e. g. [7] pp. 144) with integration interval $[0, 2]$.

Problem 2.- The CUSP problem (see e. g. [7] pp. 144) on the interval $[0, 1]$.

First, in order to show how the estimations approximate the actual local error we have integrated the problems with the code using local extrapolation to control the step-size and along this integration we have computed at each step the estimated errors provided by the different estimators. The values obtained for problem 1 and $Tol = 10^{-6}$ are displayed in figure 3. We have pictured just the most representative areas of the integration interval and as it can be appreciated extrapolation (Ext) gives the most accurate estimations. It has however the minor inconvenience that in some areas the actual error is slightly underestimated. The two-step estimator (TS) gives accurate estimations, which are in general better than SS. On the other hand, SS estimates better than HW and HW2 (which work similarly).

Secondly, to asses the performance of the estimators when they are used (separately) to control the step-size we have computed for each problem, error tolerance and error estimate the global error at the end point of the integration interval (GE), the number of successful steps (NSTEP), the number of rejected steps in the estimation test (NRE), the number of rejected steps due to convergence failure in the iterative scheme (NRC), the number of evaluations of the Jacobian matrix (JAC), the number of LU factorizations (NLU), the number of triangular systems solved

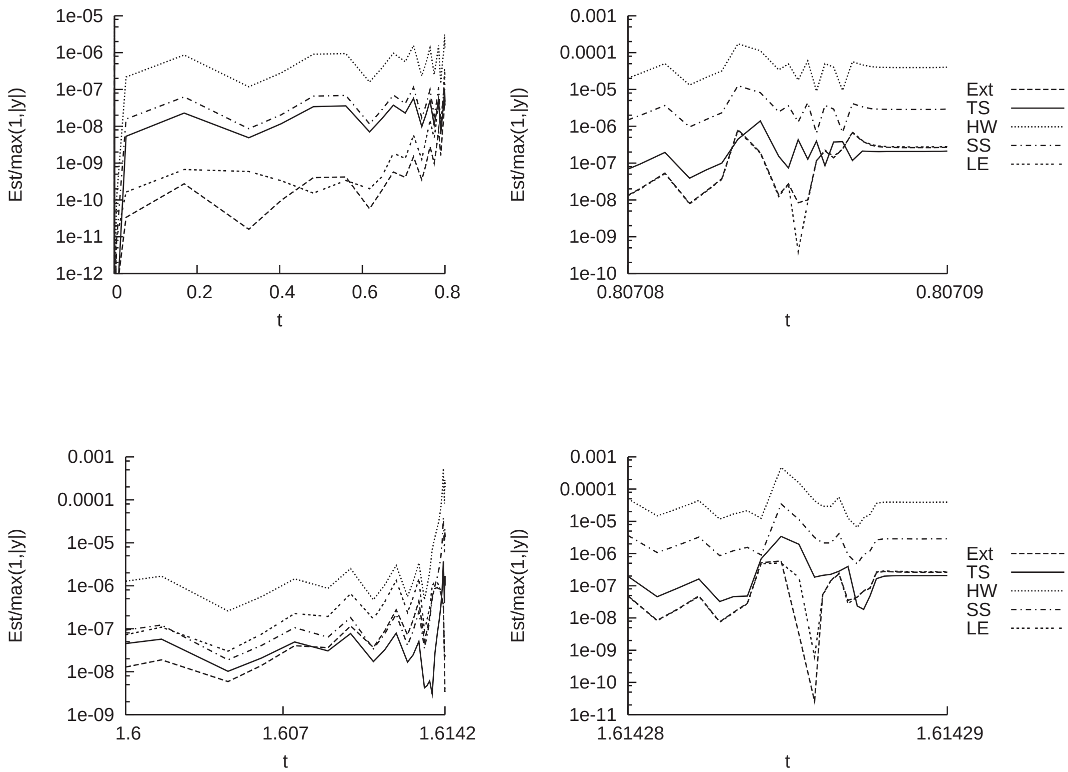


Fig. 3. Van der Pol oscillator: actual and estimated local error with $Tol = 10^{-6}$

(NSOL), the number of evaluations of the derivative function (NFN), and the average number of iterations required to reach convergence at each step (NITER). In tables 1 and 2 we present the results for the Van der Pol and CUSP problems respectively.

When a two–step mode is used (TS and HW2 rows) two values for the average number of iterations are given. They correspond to the first and the second step respectively. When extrapolation is used (Ext row) the three data for NITER correspond to the first, second and double step respectively.

It can be seen in the tables that in general the behavior of the new two–step error estimator (TS) is similar to local extrapolation and with both estimators the final errors are in a better agreement with the tolerance specified than with the other estimators. In this sense, SS performs better than HW and HW2.

For small error tolerances the code with the new two–step error estimator takes less (or a similar number of) steps to accomplish the integration than with the one–step estimators (HW and SS), for a similar global error, and this implies that the new estimator performs in general more efficiently. This difference between the behavior of the estimators is more clear when the error tolerance is decreased, as it can be expected from the fact that in one case we are using a pair 3(5) to estimate the local error while in the other case we are using a pair 4(5). It is also interesting

Table 1. Van der Pol problem.

TOL	EST	GE	$NSTEP$	NRE	NRC	JAC	LU	$NSOL$	NFN	$NITER$
10^{-2}	Ext	$0.33D-04$	174	0	70	87	372	2094	3232	$2.6, 3.2, 3.2$
	TS	$0.75D-04$	164	0	69	82	246	1752	2713	$2.7, 3.4$
	HW	$0.51D-04$	135	19	26	135	360	1544	2149	3.3
	$HW2$	$0.90D-05$	202	26	52	101	286	2133	2875	$2.5, 3.0$
	SS	$0.94D-03$	103	9	35	103	294	1424	2038	3.9
10^{-4}	Ext	$0.13D-04$	258	10	66	129	566	3534	5422	$3.2, 3.8, 3.6$
	TS	$0.84D-05$	238	12	74	119	334	2838	4378	$3.4, 4.0$
	HW	$0.28D-06$	326	24	17	326	734	2820	3754	3.1
	$HW2$	$0.10D-06$	392	42	42	196	482	3606	4801	$2.8, 3.1$
	SS	$0.42D-05$	206	18	38	206	524	2460	3463	3.8
10^{-5}	Ext	$0.19D-05$	330	20	62	165	704	4450	6784	$3.4, 4.0, 3.7$
	TS	$0.26D-05$	316	62	62	158	454	3762	5755	$3.5, 4.0$
	HW	$0.53D-07$	546	41	0	546	1174	4301	5569	3.0
	$HW2$	$0.97D-08$	622	46	38	311	706	5181	6796	$2.8, 3.0$
	SS	$0.30D-06$	316	18	27	316	722	3076	4192	3.5
10^{-6}	Ext	$0.32D-06$	460	36	65	230	990	5970	9076	$3.4, 3.9, 3.6$
	TS	$0.19D-06$	416	58	58	208	538	4448	6766	$3.5, 4.0$
	HW	$0.37D-08$	964	8	0	964	1944	5932	7438	2.5
	$HW2$	$0.20D-08$	1012	62	8	506	1082	7284	9319	$2.5, 2.9$
	SS	$0.27D-07$	526	18	19	526	1126	4394	5830	3.2
10^{-7}	Ext	$0.23D-06$	594	26	55	297	1236	7240	10957	$3.5, 3.9, 3.6$
	TS	$0.24D-07$	592	24	58	296	674	5390	8170	$3.5, 4.0$
	HW	$0.18D-09$	1727	8	0	1727	3470	9929	12289	2.4
	$HW2$	$0.17D-09$	1752	14	0	876	1766	10286	12778	$2.3, 2.5$
	SS	$0.59D-08$	888	8	0	888	1792	6390	8239	3.1
10^{-8}	Ext	$0.30D-07$	840	16	57	420	1714	9712	14665	$3.5, 3.8, 3.5$
	TS	$0.27D-08$	836	24	12	418	872	6456	9700	$3.4, 3.9$
	HW	$0.20D-10$	3090	8	0	3090	6196	17262	21244	2.3
	$HW2$	$0.20D-10$	3118	14	0	1559	3132	17608	21712	$2.3, 2.4$
	SS	$0.29D-09$	1594	8	0	1594	3204	10968	14047	2.9
10^{-9}	Ext	$0.42D-08$	1240	12	46	620	2504	13326	20065	$3.4, 3.7, 3.3$
	TS	$0.35D-09$	1352	26	0	676	1378	9378	14065	$3.2, 3.5$
	HW	$0.12D-10$	5525	9	0	5525	11068	30231	37040	2.2
	$HW2$	$0.12D-10$	5558	18	0	2779	5576	30636	37585	$2.2, 2.3$
	SS	$0.26D-10$	2851	8	0	2851	5718	17897	22555	2.6

Table 2. CUSP problem.

TOL	EST	GE	$NSTEP$	NRE	NRC	JAC	LU	$NSOL$	NFN	$NITER$
10^{-2}	Ext	$0.33D-04$	152	0	81	76	350	1402	2116	1.9, 2.4, 2.4
	TS	$0.25D-04$	142	0	82	71	250	1214	1831	2.0, 2.6
	HW	$0.14D-03$	83	0	53	83	272	943	1318	3.0
	$HW2$	$0.82D-04$	144	0	81	72	248	1337	1753	1.9, 2.5
	SS	$0.39D-03$	83	0	55	83	276	993	1387	3.3
10^{-4}	Ext	$0.19D-05$	198	0	86	99	444	2438	3742	2.6, 3.3, 3.2
	TS	$0.39D-05$	188	0	90	94	298	2066	3184	2.8, 3.3
	HW	$0.78D-06$	153	23	35	153	422	1774	2433	3.4
	$HW2$	$0.45D-06$	226	24	77	113	340	2496	3385	2.5, 3.0
	SS	$0.25D-04$	113	11	47	113	342	1658	2362	4.2
10^{-5}	Ext	$0.24D-06$	236	0	88	118	516	3218	4957	3.0, 3.7, 3.5
	TS	$0.12D-05$	214	4	86	107	324	2564	3958	3.2, 3.9
	HW	$0.37D-07$	235	36	36	235	614	2522	3449	3.3
	$HW2$	$0.13D-07$	312	42	64	156	426	3292	4438	2.7, 3.1
	SS	$0.83D-06$	161	29	43	161	466	2306	3256	4.1
10^{-6}	Ext	$0.28D-07$	284	6	98	142	610	4018	6166	3.3, 4.0, 3.8
	TS	$0.76D-07$	266	14	92	133	380	3276	5032	3.4, 4.1
	HW	$0.22D-08$	361	38	32	361	862	3412	4595	3.2
	$HW2$	$0.12D-08$	446	52	56	223	558	4316	5758	2.9, 3.2
	SS	$0.16D-07$	232	41	47	232	640	2967	4123	3.9
10^{-7}	Ext	$0.30D-07$	362	14	99	181	800	5232	8023	3.5, 4.2, 3.7
	TS	$0.81D-07$	346	52	81	173	494	4300	6571	3.7, 4.3
	HW	$0.12D-09$	594	26	33	594	1306	4749	6272	2.9
	$HW2$	$0.14D-09$	688	80	56	344	836	6090	8035	2.7, 3.1
	SS	$0.52D-08$	355	38	48	355	882	3941	5425	3.7
10^{-8}	Ext	$0.14D-08$	466	26	92	233	1010	6670	10165	3.7, 4.3, 3.8
	TS	$0.17D-08$	474	84	76	237	648	5514	8389	3.7, 4.2
	HW	$0.13D-10$	1015	16	26	1015	2114	6859	8799	2.6
	$HW2$	$0.72D-11$	1112	50	58	556	1232	8241	10669	2.5, 2.8
	SS	$0.21D-09$	572	25	47	572	1288	5447	7399	3.5
10^{-9}	Ext	$0.35D-09$	638	48	87	319	1370	8766	13294	3.7, 4.2, 3.7
	TS	$0.12D-09$	656	74	71	328	806	6750	10219	3.7, 4.2
	HW	$0.25D-11$	1777	10	15	1777	3604	10769	13497	2.4
	$HW2$	$0.12D-11$	1860	38	49	930	1952	12258	15565	2.4, 2.6
	SS	$0.10D-10$	967	16	44	967	2054	8111	10804	3.3

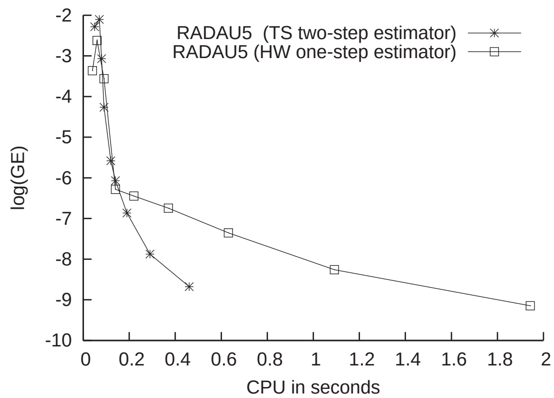


Fig. 4. Efficiency plot for Van der Pol problem using the code RADAU5

to note that for small tolerances HW and HW2 give similar number of steps. The two–step strategy is however advantageous because it lets the code integrate with fewer LU factorizations.

For large tolerances in general the one–step estimators (HW and SS) requires fewer steps to integrate the problem than the two–step estimators and consequently they need fewer evaluations of the derivative function and fewer solutions of triangular systems. The reason for it seems to be that the two–step strategy is less flexible, and the stepsize is adjusted less frequently. This can be observed from the results obtained with HW and HW2. However, the two–step strategy lets the code integrate doing fewer LU factorizations in any case.

It is also remarkable that with the two–step strategy and large error tolerances, there are many steps rejected because the iterative scheme cannot solve the implicit equations. Also in the second step the iterative scheme requires more iterations than in the first step, and this can be explained because the second step uses the Jacobian matrix computed in the first step. A better strategy of re–evaluating the Jacobian could improve the efficiency in a production code.

Finally, to test the efficiency of the new estimator in a production code, instead of using the code we have developed, we have compared the performance of the code RADAU5 with a modification of it using the proposed two–step estimate (TS). In figures 4 and 5 we give the efficiency plots (CPU time versus Log(GE)) obtained integrating the problems 1 and 2 with absolute and relative error tolerances 10^{-3}, $10^{-4}, \ldots$. For the Van der Pol problem we have taken for this experiment a longer integration interval $[0, 20]$ to get more significant CPU time values. As expected, for small error tolerances it is clearly seen in the figures that the code improves its efficiency when the two–step estimator (TS) is used, while the performance with both estimators is similar for large error tolerances.

REFERENCES

[1] A. Bellen, Z. Jackiewitz and M. Zennaro, *Local error estimation for singly–implicit formulas by two–step Runge-Kutta methods*, BIT 32 (1992), pp. 104–117.

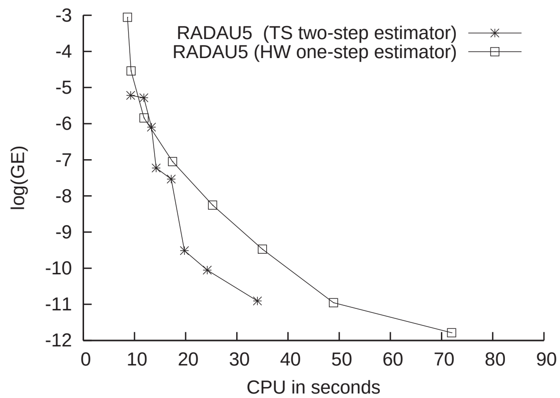


Fig. 5. Efficiency plot for Cusp problem using the code RADAU5

[2] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta an General Linear Methods*, John Wiley & Sons, Chichester, 1987.

[3] J.C. Butcher and T.M.H. Chan, *Multi–step zero approximations for stepsize control*, Appl. Numer. Math. 34 (2000), pp. 167–177.

[4] J.C. Butcher and P.B. Johnston, *Estimating local truncation errors for Runge–Kutta metods*, J. Comput. Appl. Math. 45 (1993) 1–2, pp. 203–212.

[5] W. H. Enright, T. E. Hull and B. Lindberg, *Comparing numerical methods for stiff systems of ODEs*, BIT, 15 (1975), pp. 10–48.

[6] S. Gonzalez-Pinto, J. I. Montijano and L. Randez, *Iterative schemes for implicit Runge-Kutta methods*, Appl. Numer. Math., 17 (1995), pp. 363-382.

[7] E. Hairer and G. Wanner, *Solving ordinary differential equations II*, Springer Verlag, Berlin, 1996.

[8] E. Hairer and G. Wanner, *Stiff differential equations solved by Radau methods*, J. Comput. Appl. Math., 111 (1999), pp. 93-111.

[9] L.F. Shampine and L.S. Baca, *Error estimators for stiff differential equations*, J. Comp. Appl. Math., 11 (1984), pp. 197–207.

[10] J.J.B. de Swart and G. Söderlind, *On the construction of error estimators for implicit Runge–Kutta methods*, J. Comp. Appl. Math., 86 (1997), pp. 347–358.