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, lets 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

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or permissions@acm.org.
tolerance. With Runge–Kutta methods

\begin{equation}
Y_{n,i} = y_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{n,j}) \quad (i = 1, \ldots, s),
\end{equation}

\begin{equation}
y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(t_n + c_i h, Y_{n,i}),
\end{equation}

the estimator is commonly based on an embedded formula

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

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

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

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

\[ \|Est\| \leq \text{Tol}. \]

Let us suppose now that the estimator is overestimating the actual error, that is, \(\|Est\| >> Error\). This makes the code take step sizes smaller than those really needed to get \(Error \leq \text{Tol}\). 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 \(c \text{Tol}\) 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
Two–step error estimators for implicit RK 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

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

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,

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

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 \( b_0 \). 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 inconvenience 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 \( t_n \rightarrow t_{n+1} = t_n + h \) and next from \( t_{n+1} \rightarrow t_{n+2} = t_{n+1} + h \).

We will consider an embedded formula of the following type

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

methods)
where $\beta_j$ and $\gamma_j$ 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 overcome 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

\[ C(q) : \quad A c^{j-1} = c^j / j, \quad j = 1 \ldots q \]
\[ B(q) : \quad b^T c^{j-1} = 1 / j, \quad j = 1 \ldots q \]

Let $T$ denote the set of unlabeled rooted trees, $\tau \in T$ a rooted tree and $\tau_0$ the tree with only one vertex (the only tree with order 1). Following the notations in [2], a tree $\tau$ with order $\rho(\tau) > 1$ can be written $\tau = [\tau_1', \ldots, \tau_k']$ where $\tau_i'$ are the subtrees of $\tau$ that attached to the root give $\tau$.

It is easy to see that the method reaches order $q + 1$ if and only if the order condition for the tree $\tau_q = [\tau_0, \ldots, \tau_0] = [\tau_q]_0$ is fulfilled. Then, order $q + 2$ is attained if in addition the order conditions associated to the trees $\tau_{q+1}$ and $[\tau_q]$ are satisfied. The order $q + 3$ demands the order conditions on these trees and also on $\tau_{q+2}$, $[\tau_{q+1}]$, $[\tau_q, \tau_0]$ and $[[\tau_q]]$. We are interested in characterizing the subset $ST_q$ of the tree set $T$ such that for any $r \geq 1$

\[ \text{order } q + r \iff b^T \Phi(\tau) = 1 / \gamma(\tau), \forall \tau \in ST_q \text{ with } \rho(\tau) \leq q + r \]
Definition 1. We will say that $\tau'$ is a descendant of $\tau$ if $\tau' \equiv \tau$ or well $\tau'$ is a subtree of $\tau$ or it is a subtree of a subtree of $\tau$ and so on.

Example 1. The tree $\tau_q$ is a descendant of each of the following trees:

\[ \tau_q \]

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

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

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

\[ ST_2 = \{ \tau \} \]
\[ ST_3 = \{ \tau \} \]
\[ ST_4 = \{ \tau \} \]

With this definition it is trivial the following

Lemma 1. $ST_n \subseteq ST_m$ for all $n \geq m \geq 1$.

Denoting now $ST_q^r = \{ \tau \in ST_q, p(\tau) \leq q + r \}$, 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.,

\[ b^T \Phi(\tau) = 1/\gamma(\tau), \quad \forall \tau \in ST_q^r \]

Proof. If $q = 1$ or $r = 1$ the proof is trivial. Let $\tau \notin ST_q$ with $q + 1 \leq p(\tau) \leq q + r$, it has at least a descendant $\tau' = [\tau_0, \ldots, \tau_0]$ with order $p = p(\tau') \leq q$. If $\tau = \tau'$, the proof is trivial by $C(q)$ and $B(q)$. Otherwise, $\tau$ must have a descendant of the form $\tilde{\tau} = [\tau', \tau_1', \ldots, \tau_k']$, $k \geq 0$. Now, by $C(q)$ the order condition for $\tau$ is satisfied if it is satisfied the order condition corresponding to the tree $\tau^*$, which has the same order as $\tau$ obtained by replacing in $\tau$ the descendant $\tilde{\tau}$ by $\tilde{\tau} = [\tau_0^{p+1}, \tau_1', \ldots, \tau_k']$. 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 $(A, b)$ the coefficients of the composed method
resulting after two consecutive steps of size $h$, given by the Butcher tableau

$$
\begin{array}{c|c}
\hat{e} & \hat{A} \\
\hline
\hat{b} & \begin{bmatrix} c & e + c \end{bmatrix}
\end{array}
\begin{array}{c}
A \\
0 \\
\end{array}
\begin{bmatrix}
b^T \\
\end{bmatrix},
$$

$\hat{e}^T = (e^T, e^T)$ 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 $\hat{\beta}^T = (\beta^T, \gamma^T) = (\beta_1, \ldots, \beta_s, \gamma_1, \ldots, \gamma_s)$, with orders $q + 1$ for $l = 1, \ldots, r$. Since the composed method also satisfies $C(q), \hat{A}\hat{\varepsilon}^{i-1} = \hat{\varepsilon}^i/j,$ $j = 1, \ldots, q$, we can apply Theorem 1 to the embedded method, replacing the order conditions (5) by

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

**Theorem 2.** For every $k = 0, \ldots, r$ there exist a family of embedded methods $(\hat{A}, \hat{\beta})$ of order $q + k$ with $l_k = 2s - \text{rank}(M_{q,k})$ 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 $\tau \in ST_q^r$.

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

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

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 $\geq 5$ ($s = 3, q = 3$) such as Radau IIA and Gauss,

- If $k = 1, M_{s,1} = [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3]$ and $\text{rank}(M_{s,1}) = 4$. Then there exist a biparametric family of embedded methods with order 4.
- If $k = 2, M_{s,2} = [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3, \hat{A}\hat{c}^4]$ and $\text{rank}(M_{s,2}) = 6$. Then there is a unique method of order $\geq 5$ (the composed method) and therefore there is not any embedded method with order 5.

For four-stage collocation methods of order $\geq 7$ ($s = 4, q = 4$)

- If $k = 1, M_{s,1} = [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3, \hat{c}^4]$ and $\text{rank}(M_{s,1}) = 5$. Then there exist a three-parameter family of embedded methods with order 5.
- If $k = 2, M_{s,2} = [\hat{e}, \hat{c}, \ldots, \hat{c}^5, \hat{A}\hat{c}^6]$ and $\text{rank}(M_{s,2}) = 7$. Then there is a one-parameter family of embedded methods of order $\geq 6$.
- If $k = 3, M_{s,3} = [\hat{e}, \hat{c}, \ldots, \hat{c}^6, \hat{A}\hat{c}^7, \hat{A}\hat{c}^4, \hat{A}\hat{c}^5, \hat{A}\hat{c}^6, \hat{A}\hat{c}^3]$ and $\text{rank}(M_{s,3}) = 8$. The only method satisfying these equations, and therefore with order $\geq 7$, is the composed method.

For five-stage collocation methods of order $\geq 9$ ($s = 5, q = 5$)
If $k = 1$, $M_{s,1} = [\hat{e}, \hat{c}, \ldots, \hat{c}]^{\circ}$ and 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 rank($M_{s,2}$) = 8. Then there is a two–parameter family of embedded methods of order $\geq 7$.

If $k = 3$, $M_{s,3} = [\hat{e}, \hat{c}, \ldots, \hat{c}^7, \hat{A}\hat{c}^6, \hat{A}^2\hat{c}^5, \hat{A}^2\hat{c}^5, \hat{A}^2\hat{c}^5]$ and rank($M_{s,3}$) = 10. The only method satisfying these equations, and therefore with order $\geq 8$, 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 $\hat{R}(z) = P(z)/Q(z)$ 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

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

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 $|\hat{R}(\infty)|^2$ 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 \( p = 4 \),
\[
\hat{P}(z) = R(z)\hat{Q}(z) = 1 + \frac{4}{5}z + \frac{13}{50}z^2 + \frac{1}{25}z^3 + \frac{1}{400}z^4 + uz^5 + vz^6
\]
with \( u = \hat{\beta}^T\hat{A}^3\hat{c} - \frac{2}{5!} \) 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
\[
\hat{\beta}^T[\hat{e}, \hat{c}, \hat{e}^2, \hat{c}^2, \hat{e}^3, \hat{c}^3, \hat{v}, \hat{w}] = \begin{pmatrix} 2 & \frac{2}{3} & \frac{2}{4} & -\frac{277}{1200} & u + \frac{2}{5!} \end{pmatrix}
\]
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 \((b^T, \hat{b}^T) - \hat{\beta}^T\) can be expressed in terms of the free parameter \( u \) as
\[
(b^T, \hat{b}^T) = u \frac{4}{5} \begin{pmatrix} 19 - 14\sqrt{6}, 19 + 14\sqrt{6}, 52, -29 - 51\sqrt{6}, -29 + 51\sqrt{6}, -32 \end{pmatrix}.
\]

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 \(|\hat{R}(iy)|^2 \leq 1\) or equivalently if \(|\hat{P}(iy)|^2 - |\hat{Q}(iy)|^2 \leq 0\) 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
\[
\frac{1}{12960000}x^3 - (u^2 - \frac{1}{720000})x^2 + \left( \frac{1}{30000} + \frac{2}{25}u \right)x + \frac{1}{1800} - \frac{8}{5}u \geq 0, \quad \text{for all } x \geq 0.
\]

This condition is satisfied for all \( u \in [-0.001904604018365, \ldots, \frac{1}{2800}] \).

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

\[
Est_{ext}(z) = \frac{|R^2(z) - R(2z)|}{31}.
\]

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

\[
Est_{HW}(z) = \left| \frac{R(z)(R(z) - \hat{R}(z))/(1 - z\gamma)}{1 - \frac{4}{5}z + \frac{1}{60}z^3} \right|
\]

with \( \hat{R}(z) \) the amplifying function of the embedded third–order method and \( \gamma \) 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

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

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

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

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

\[
Err(x) \leq Est_{TS}(x), \quad \forall x \leq x_0 = -2.605...
\]

\[
Err(x) - Est_{TS}(x) \leq 8.3 \cdot 10^{-5}, \quad Err(x) \leq 1.96 \cdot Est_{TS}(x), \quad \forall x \in (x_0, 0].
\]
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 $R(z)$ 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 \( \hat{R}(\infty) = \frac{v}{750600} \) while \( R^2(z) = O(1/z^2) \) 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
\[ \hat{\beta}^T [\hat{e}, \hat{c}, \hat{c}^2, \hat{c}^3] = \left[ 2, 2^2/2, 3^2/3, 4^2/4, 5^2/5 \right] \]
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
\[ \hat{P}(z) = P^2(z) + wz^6 + vz^7 \]
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
\[ \sum_{\rho(\tau) = 6} \alpha(\tau)(2^6 - \gamma(\tau)\beta^T \hat{\Phi}(\tau))F(\tau)(y_0) = K \sum_{\rho(\tau) = 6} \alpha(\tau)F(\tau)(y_0) = K6!y^{(6)}(t_0) \]
for some constant \( K \). Taking into account the simplifying assumptions, this is accomplished if and only if \( \beta^T (\hat{c}^5 - 5\hat{A} \hat{c}^4) = 0 \) and since \( \hat{A}^6 \hat{c} = \hat{A}^2 \hat{c}^4/4! \), \( \hat{A}^5 \hat{c} = \hat{A}^2 \hat{c}^4/4! \) and \( \hat{A}^4 \hat{c} = \hat{A} \hat{c}^4/4! \), the family of embedded methods satisfying the required
conditions is given by
\[ \hat{\beta}^T \left[ \hat{c}, \hat{e}, \hat{c}^2, \hat{c}^3, \hat{c}^4, \hat{A} \hat{c}, \left( \hat{A}^3 - \frac{8}{7} \hat{A}^2 \right) \hat{c}^4 \right] = \left( 2, \frac{2^2}{2}, \frac{2^3}{3}, \frac{2^4}{4}, \frac{2^5}{5}, \frac{32}{3}, \frac{32}{6}, K \frac{15}{15}, - \frac{K}{30}, - \frac{16001}{29400} + \frac{K}{49} \right) \]
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 \( K = -0.00101470776549531547265801395193 \) 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
\[ \hat{R}(z) = R^2(z) + \frac{w z^8 + v z^9 + u z^{10}}{Q^2(z)} \]
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 assess 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 assess 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
(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.

<table>
<thead>
<tr>
<th>TOL</th>
<th>EST</th>
<th>GE</th>
<th>NSTEP</th>
<th>NRE</th>
<th>NRC</th>
<th>JAC</th>
<th>LU</th>
<th>NSOL</th>
<th>NFN</th>
<th>NITER</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>Ext</td>
<td>0.33 - 04</td>
<td>174</td>
<td>0</td>
<td>70</td>
<td>87</td>
<td>372</td>
<td>2094</td>
<td>3232</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.75 - 04</td>
<td>164</td>
<td>0</td>
<td>69</td>
<td>82</td>
<td>246</td>
<td>1752</td>
<td>2713</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.51 - 04</td>
<td>135</td>
<td>19</td>
<td>26</td>
<td>135</td>
<td>360</td>
<td>1544</td>
<td>2149</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.90 - 05</td>
<td>202</td>
<td>26</td>
<td>52</td>
<td>101</td>
<td>286</td>
<td>2133</td>
<td>2875</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.94 - 03</td>
<td>103</td>
<td>9</td>
<td>35</td>
<td>103</td>
<td>294</td>
<td>1424</td>
<td>2018</td>
<td>3.9</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>Ext</td>
<td>0.13 - 04</td>
<td>258</td>
<td>10</td>
<td>66</td>
<td>129</td>
<td>566</td>
<td>3534</td>
<td>5422</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.84 - 05</td>
<td>238</td>
<td>12</td>
<td>74</td>
<td>119</td>
<td>334</td>
<td>2838</td>
<td>4378</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.28 - 06</td>
<td>326</td>
<td>24</td>
<td>17</td>
<td>326</td>
<td>734</td>
<td>2820</td>
<td>3754</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.10 - 06</td>
<td>392</td>
<td>42</td>
<td>42</td>
<td>196</td>
<td>482</td>
<td>3606</td>
<td>4801</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.42 - 05</td>
<td>206</td>
<td>18</td>
<td>38</td>
<td>206</td>
<td>524</td>
<td>2460</td>
<td>3463</td>
<td>3.8</td>
</tr>
<tr>
<td>10^{-5}</td>
<td>Ext</td>
<td>0.19 - 05</td>
<td>330</td>
<td>20</td>
<td>62</td>
<td>165</td>
<td>704</td>
<td>4450</td>
<td>6784</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.26 - 05</td>
<td>316</td>
<td>62</td>
<td>62</td>
<td>158</td>
<td>454</td>
<td>3762</td>
<td>5755</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.53 - 07</td>
<td>546</td>
<td>41</td>
<td>0</td>
<td>546</td>
<td>1174</td>
<td>4301</td>
<td>5569</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.97 - 08</td>
<td>622</td>
<td>46</td>
<td>38</td>
<td>311</td>
<td>706</td>
<td>5181</td>
<td>6796</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.30 - 06</td>
<td>316</td>
<td>18</td>
<td>27</td>
<td>316</td>
<td>722</td>
<td>3076</td>
<td>4192</td>
<td>3.5</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>Ext</td>
<td>0.32 - 06</td>
<td>460</td>
<td>36</td>
<td>65</td>
<td>230</td>
<td>990</td>
<td>5970</td>
<td>9076</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.19 - 06</td>
<td>416</td>
<td>58</td>
<td>58</td>
<td>208</td>
<td>538</td>
<td>4448</td>
<td>6766</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.37 - 08</td>
<td>964</td>
<td>8</td>
<td>0</td>
<td>964</td>
<td>1944</td>
<td>4322</td>
<td>7438</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.20 - 08</td>
<td>1012</td>
<td>62</td>
<td>8</td>
<td>506</td>
<td>1082</td>
<td>7284</td>
<td>9319</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.27 - 07</td>
<td>526</td>
<td>18</td>
<td>19</td>
<td>526</td>
<td>1126</td>
<td>4394</td>
<td>5830</td>
<td>3.2</td>
</tr>
<tr>
<td>10^{-7}</td>
<td>Ext</td>
<td>0.23 - 06</td>
<td>594</td>
<td>26</td>
<td>55</td>
<td>297</td>
<td>1236</td>
<td>7240</td>
<td>10957</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.24 - 07</td>
<td>592</td>
<td>24</td>
<td>58</td>
<td>296</td>
<td>674</td>
<td>5390</td>
<td>8170</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.18 - 09</td>
<td>1727</td>
<td>8</td>
<td>0</td>
<td>1727</td>
<td>3470</td>
<td>9929</td>
<td>12289</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.17 - 09</td>
<td>1752</td>
<td>14</td>
<td>0</td>
<td>876</td>
<td>1766</td>
<td>10286</td>
<td>12778</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.59 - 08</td>
<td>888</td>
<td>8</td>
<td>0</td>
<td>888</td>
<td>1792</td>
<td>6390</td>
<td>8239</td>
<td>3.1</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>Ext</td>
<td>0.30 - 07</td>
<td>840</td>
<td>16</td>
<td>57</td>
<td>420</td>
<td>1714</td>
<td>9712</td>
<td>14665</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.27 - 08</td>
<td>836</td>
<td>24</td>
<td>12</td>
<td>418</td>
<td>872</td>
<td>6456</td>
<td>9700</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.20 - 10</td>
<td>3090</td>
<td>8</td>
<td>0</td>
<td>3090</td>
<td>6196</td>
<td>17262</td>
<td>21244</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.20 - 10</td>
<td>3118</td>
<td>14</td>
<td>0</td>
<td>1559</td>
<td>3132</td>
<td>17608</td>
<td>21712</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.29 - 09</td>
<td>1594</td>
<td>8</td>
<td>0</td>
<td>1594</td>
<td>3204</td>
<td>10968</td>
<td>14047</td>
<td>2.9</td>
</tr>
<tr>
<td>10^{-9}</td>
<td>Ext</td>
<td>0.42 - 08</td>
<td>1240</td>
<td>12</td>
<td>46</td>
<td>620</td>
<td>2504</td>
<td>13326</td>
<td>20065</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.35 - 09</td>
<td>1352</td>
<td>26</td>
<td>0</td>
<td>676</td>
<td>1378</td>
<td>9378</td>
<td>14665</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.12 - 10</td>
<td>5525</td>
<td>9</td>
<td>0</td>
<td>5525</td>
<td>11068</td>
<td>30231</td>
<td>37040</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.12 - 10</td>
<td>5558</td>
<td>18</td>
<td>0</td>
<td>2779</td>
<td>5576</td>
<td>30636</td>
<td>37585</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.26 - 10</td>
<td>2851</td>
<td>8</td>
<td>0</td>
<td>2851</td>
<td>5718</td>
<td>17897</td>
<td>22555</td>
<td>2.6</td>
</tr>
</tbody>
</table>
Table 2. CUSP problem.

<table>
<thead>
<tr>
<th>TOL</th>
<th>EST</th>
<th>GE</th>
<th>NSTEP</th>
<th>NRE</th>
<th>NRC</th>
<th>JAC</th>
<th>LU</th>
<th>NSOL</th>
<th>NFN</th>
<th>NITER</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>Ext</td>
<td>0.33D - 04</td>
<td>152</td>
<td>0</td>
<td>81</td>
<td>76</td>
<td>350</td>
<td>1402</td>
<td>2116</td>
<td>1.9, 2.4, 2.4</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.25D - 04</td>
<td>142</td>
<td>0</td>
<td>82</td>
<td>71</td>
<td>250</td>
<td>1214</td>
<td>1831</td>
<td>2.0, 2.6</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.14D - 03</td>
<td>83</td>
<td>0</td>
<td>53</td>
<td>83</td>
<td>272</td>
<td>943</td>
<td>1318</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.82D - 04</td>
<td>144</td>
<td>0</td>
<td>81</td>
<td>72</td>
<td>248</td>
<td>1337</td>
<td>1753</td>
<td>1.9, 2.5</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.39D - 03</td>
<td>83</td>
<td>0</td>
<td>55</td>
<td>83</td>
<td>276</td>
<td>993</td>
<td>1387</td>
<td>3.3</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>Ext</td>
<td>0.19D - 05</td>
<td>198</td>
<td>0</td>
<td>86</td>
<td>99</td>
<td>444</td>
<td>2438</td>
<td>3742</td>
<td>2.6, 3.3, 3.2</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.39D - 05</td>
<td>188</td>
<td>0</td>
<td>90</td>
<td>94</td>
<td>298</td>
<td>2066</td>
<td>3184</td>
<td>2.8, 3.3</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.78D - 06</td>
<td>153</td>
<td>23</td>
<td>35</td>
<td>153</td>
<td>422</td>
<td>1774</td>
<td>2433</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.45D - 06</td>
<td>226</td>
<td>24</td>
<td>77</td>
<td>113</td>
<td>340</td>
<td>2496</td>
<td>3385</td>
<td>2.5, 3.0</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.25D - 04</td>
<td>113</td>
<td>11</td>
<td>47</td>
<td>113</td>
<td>342</td>
<td>1658</td>
<td>2362</td>
<td>4.2</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>Ext</td>
<td>0.24D - 06</td>
<td>236</td>
<td>0</td>
<td>88</td>
<td>118</td>
<td>516</td>
<td>3218</td>
<td>4597</td>
<td>3.0, 3.7, 3.5</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.12D - 05</td>
<td>214</td>
<td>4</td>
<td>86</td>
<td>107</td>
<td>324</td>
<td>2564</td>
<td>3958</td>
<td>3.2, 3.9</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.37D - 07</td>
<td>235</td>
<td>36</td>
<td>36</td>
<td>235</td>
<td>614</td>
<td>2522</td>
<td>3449</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.13D - 07</td>
<td>312</td>
<td>42</td>
<td>64</td>
<td>156</td>
<td>426</td>
<td>3292</td>
<td>4438</td>
<td>2.7, 3.1</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.83D - 06</td>
<td>161</td>
<td>29</td>
<td>43</td>
<td>161</td>
<td>466</td>
<td>2306</td>
<td>3256</td>
<td>4.1</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>Ext</td>
<td>0.28D - 07</td>
<td>284</td>
<td>6</td>
<td>98</td>
<td>142</td>
<td>610</td>
<td>4018</td>
<td>6166</td>
<td>3.3, 4.0, 3.8</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.76D - 07</td>
<td>266</td>
<td>14</td>
<td>92</td>
<td>133</td>
<td>380</td>
<td>3276</td>
<td>5032</td>
<td>3.4, 4.1</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.22D - 08</td>
<td>361</td>
<td>38</td>
<td>32</td>
<td>361</td>
<td>862</td>
<td>3412</td>
<td>4595</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.12D - 08</td>
<td>446</td>
<td>52</td>
<td>56</td>
<td>223</td>
<td>558</td>
<td>4316</td>
<td>5758</td>
<td>2.9, 3.2</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.16D - 07</td>
<td>232</td>
<td>41</td>
<td>47</td>
<td>232</td>
<td>640</td>
<td>2967</td>
<td>4123</td>
<td>3.9</td>
</tr>
<tr>
<td>10^{-10}</td>
<td>Ext</td>
<td>0.30D - 07</td>
<td>362</td>
<td>14</td>
<td>99</td>
<td>181</td>
<td>800</td>
<td>5232</td>
<td>8023</td>
<td>3.5, 4.2, 3.7</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.81D - 07</td>
<td>346</td>
<td>52</td>
<td>81</td>
<td>173</td>
<td>494</td>
<td>4300</td>
<td>6571</td>
<td>3.7, 4.3</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.12D - 09</td>
<td>594</td>
<td>26</td>
<td>33</td>
<td>594</td>
<td>1306</td>
<td>4749</td>
<td>6272</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.14D - 09</td>
<td>688</td>
<td>80</td>
<td>56</td>
<td>344</td>
<td>836</td>
<td>6090</td>
<td>8035</td>
<td>2.7, 3.1</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.52D - 08</td>
<td>355</td>
<td>38</td>
<td>48</td>
<td>355</td>
<td>882</td>
<td>3941</td>
<td>5425</td>
<td>3.7</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>Ext</td>
<td>0.14D - 08</td>
<td>446</td>
<td>26</td>
<td>92</td>
<td>233</td>
<td>1010</td>
<td>6670</td>
<td>10165</td>
<td>3.7, 4.3, 3.8</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.17D - 08</td>
<td>474</td>
<td>84</td>
<td>76</td>
<td>237</td>
<td>648</td>
<td>5514</td>
<td>8389</td>
<td>3.7, 4.2</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.13D - 10</td>
<td>1015</td>
<td>16</td>
<td>26</td>
<td>1015</td>
<td>2114</td>
<td>6859</td>
<td>8799</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.72D - 11</td>
<td>1112</td>
<td>50</td>
<td>58</td>
<td>556</td>
<td>1232</td>
<td>8241</td>
<td>10669</td>
<td>2.5, 2.8</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.21D - 09</td>
<td>572</td>
<td>25</td>
<td>47</td>
<td>572</td>
<td>1288</td>
<td>5447</td>
<td>7399</td>
<td>3.5</td>
</tr>
<tr>
<td>10^{-9}</td>
<td>Ext</td>
<td>0.35D - 09</td>
<td>638</td>
<td>48</td>
<td>87</td>
<td>319</td>
<td>1370</td>
<td>8766</td>
<td>13294</td>
<td>3.7, 4.2, 3.7</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>0.12D - 09</td>
<td>656</td>
<td>74</td>
<td>71</td>
<td>328</td>
<td>806</td>
<td>6750</td>
<td>10219</td>
<td>3.7, 4.2</td>
</tr>
<tr>
<td></td>
<td>HW</td>
<td>0.25D - 11</td>
<td>1777</td>
<td>10</td>
<td>15</td>
<td>1777</td>
<td>3604</td>
<td>10769</td>
<td>13497</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>HW2</td>
<td>0.12D - 11</td>
<td>1860</td>
<td>38</td>
<td>49</td>
<td>930</td>
<td>1952</td>
<td>12258</td>
<td>15565</td>
<td>2.4, 2.6</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.10D - 10</td>
<td>967</td>
<td>16</td>
<td>44</td>
<td>967</td>
<td>2054</td>
<td>8111</td>
<td>10804</td>
<td>3.3</td>
</tr>
</tbody>
</table>
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 $10^{-3}$, $10^{-4}$, . . . 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
Fig. 5. Efficiency plot for Cusp problem using the code RADAU5


