

Derivation and Implementation of Two-Step Runge–Kutta Pairs

Z. Jackiewicz* and J.H. Verner †

July 28, 2004

Abstract. Explicit Runge–Kutta pairs are known to provide efficient solutions to initial value differential equations with inexpensive derivative evaluations. Two-step Runge–Kutta methods strive to improve the efficiency by utilizing approximations to the solution and its derivatives from the previous step. This article suggests a strategy for computing embedded *pairs* of such two-step methods using a smaller number of function evaluations than that required for traditional Runge–Kutta methods of the same order. This leads to the efficient and reliable estimation of local discretization error and a robust step control strategy. The change of stepsize is achieved by a suitable interpolation of stage values from the previous step and does not require any additional function evaluations. Two examples illustrate the features of these pairs.

Key words. two-step Runge–Kutta methods, order conditions, local error estimation, implementation aspects

AMS subject classifications. 65L05, 65L06, 65L20.

*Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804 (jackiewi@math.la.asu.edu). The work of this author was supported by the National Science Foundation grant NSF DMS-9971164.

†Department of Mathematics, PIMS, Simon Fraser University, Burnaby, Canada, V5A 1S6 (jverner@pims.math.ca). The work of this author was supported by the National Science and Engineering Council of Canada grant A8147.

1 Introduction

Explicit Runge–Kutta (RK) pairs are known to be efficient algorithms for obtaining approximate solutions to initial value problems for nonstiff and mildly stiff ordinary differential equations (ODEs) with relatively inexpensive derivative function evaluations. Here, we shall assume that an initial value problem is written as an autonomous system

$$\begin{cases} y'(x) = f(y(x)), & x \in [x_0, x_{end}], \\ y(x_0) = y_0, \end{cases} \quad (1.1)$$

where the derivative function $f : R^n \rightarrow R^n$ is assumed to be sufficiently smooth.

RK methods for the numerical solution of (1.1) require many evaluations of the function f per step and hence are not as efficient as, for example, linear multistep methods, when the derivative evaluations are relatively expensive. To seek compromises between the strengths and weaknesses of the standard methods, a number of authors [4], [5], [7], [8], [9], [10], [12], [14], [15], [16], [17], [18], [19], [20], [23], [24] have studied the possibility of using approximations to the solution and its derivatives at two consecutive steps. This approach leads to the general class of two-step Runge–Kutta (TSRK) methods of the form

$$\begin{cases} Y_i^j = u_j y_{i-1} + (1 - u_j) y_i + h \sum_{k=1}^s (a_{jk} f(Y_{i-1}^k) + b_{jk} f(Y_i^k)), \\ y_{i+1} = \theta y_{i-1} + (1 - \theta) y_i + h \sum_{j=1}^s (v_j f(Y_{i-1}^j) + w_j f(Y_i^j)), \end{cases} \quad (1.2)$$

$j = 1, 2, \dots, s$, $i = 1, 2, \dots, N - 1$. Here, N is a positive integer; $h = (X - x_0)/N$ is a fixed stepsize; $x_i = x_0 + ih$, $i = 0, 1, \dots, N$, are the grid points. Also, $\{Y_i^j, j = 1, \dots, s\}$ are approximations of stage order q to $y(x_i + c_j h)$ and y_i is an approximation of order p to $y(x_i)$: this means that $\{Y_i^j = y(x_i + c_j h) + O(h^{q+1}), j = 1, 2, \dots, s\}$, and $y_i = y(x_i) + O(h^{p+1})$ respectively as $h \rightarrow 0$. Variable step implementations exploit explicit RK pairs for efficiency, and so the focus here is on the derivation of TSRK pairs suitable for implementation with variable stepsizes.

As TSRK methods have been introduced only recently, production software based on them has not yet appeared. However, the numerical evidence

presented in [5] with an explicit TSRK method of order five demonstrates that these methods are quite reliable and more efficient than the corresponding RK methods of the same order. Hence, we might expect that the results presented in [5] and in this paper will stimulate the construction of and contribute to reliable and efficient software based on TSRK pairs that might compete with that now existing for RK and linear multistep methods.

Each TSRK method to be derived can be conveniently displayed as an array of their coefficients in the following form:

$$\frac{\begin{array}{c|c|c} u & A & B \\ \theta & v^T & w^T \end{array}}{=} = \frac{\begin{array}{c|cccc|cc} u_1 & a_{11} & a_{12} & \dots & a_{1s} & & \\ u_2 & a_{21} & a_{22} & \dots & a_{2s} & b_{21} & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ u_s & a_{s1} & a_{s2} & \dots & a_{ss} & b_{s1} & \dots & b_{s,s-1} \end{array}}{\begin{array}{c|cccc|cccc} \theta & v_1 & v_2 & \dots & v_s & w_1 & w_2 & \dots & w_s \end{array}} \quad (1.3)$$

This method is explicit: one in the form shown with B as a lower triangular matrix is designated as type 1, while one with $B = 0$ is of type 3 [18].

Jackiewicz and Tracogna [18] derived the general form of order conditions for (1.2) using the approach developed recently by Albrecht [1], [2], [3] for RK and Rosenbrock methods. This approach not only leads to order conditions, but also gives the expressions for the errors of stage values. The general form of these order conditions is obtained by requiring that individual terms in the expansions of

$$v^T t_i + w^T t_{i+1} = O(h^p), \quad h \rightarrow 0, \quad (1.4)$$

$i = 0, 1, \dots$ be annihilated, where the errors $q_{i+1} = y(x_i + ch) - Y_i$ of stage values Y_i are given by

$$q_{i+1} = hAt_i + hBt_{i+1} + hd_{i+1} + O(h^p),$$

t_i is the difference in derivative evaluations,

$$t_i = f(y(x_i + ch)) - f(Y_i),$$

and d_{i+1} , the local discretization error in the propagated solution, is

$$\begin{aligned} d_{i+1} &= y(x_{i+1}) - [\theta y(x_{i-1}) + (1 - \theta)y(x_i) \\ &\quad + h \sum_{j=1}^s (v_j f(y(x_{i-1} + c_j h)) + w_j f(y(x_i + c_j h)))]. \end{aligned}$$

Here, $Y_i = [Y_i^1, \dots, Y_i^s]^T$ and $y(t_i + ch) = [y(t_i + c_1h), \dots, y(t_i + c_sh)]^T$. It is demonstrated in [18] that the quantities q_{i+1} and t_i can be generated recursively and that they depend on the so-called recursive differentials (which differ from Butcher's elementary differentials). Substituting the expansions obtained for t_i into (1.4) and equating the coefficients of each term to zero, it is possible to express order conditions in terms of the coefficients of the method only, and order conditions up to order six are listed in [18]. A Mathematica program [7] was also written to generate these conditions automatically for any arbitrary order.

Order conditions for TSRK methods (1.2) were also obtained by Butcher and Tracogna [7] using the algebraic approach by Butcher [6], by Hairer and Wanner [12] and Tracogna and Welfert [24] using the theory of B -series [11], and by Jackiewicz and Vermiglio using the theory of general linear methods with external stages of different orders described in [16]. In [17] and [4] a representation formula for a coefficient matrix A was derived for TSRK methods with $p = q = s + 1$ in terms of the vectors c and u and the coefficient matrix B . Various implementation issues related to these methods have been discussed in [5], [23], and [24].

In this paper, we adopt a number of strategies to facilitate the implementation. To *initiate* the integration, we need y_0 , y_1 , and the stage values Y_0^j , $j = 1, 2, \dots, s$, on the first step $[x_0, x_1]$. These values will be computed by a continuous RK method of order 5 constructed by Owren and Zennaro [21]. The formulas developed provide an estimate of the local error as the difference $est(x_{i+1}) = y_{i+1} - \hat{y}_{i+1}$ of two approximations of orders p and $p - 1$ respectively, where the latter is

$$\hat{y}_{i+1} = \hat{\theta}y_{i-1} + (1 - \hat{\theta})y_i + h \sum_{j=1}^s (\hat{v}_j f(Y_{i-1}^j) + \hat{w}_j f(Y_i^j)) . \quad (1.5)$$

To enhance efficiency, each stepsize is maximized to keep this estimate below a user-specific tolerance, and accordingly, we will need a technique for changing the stepsize from step to step. We focus on methods of type 1, and restrict parameters as follows. We assume throughout that $\theta = 0$ (this will also guarantee zero-stability), and also that $u = 0$ so that none of y_{i+1} , \hat{y}_{i+1} , Y_i^j will depend directly on y_{i-1} . Then, at each step of the propagation, we will use $s = p/2 + 1$ stages, where p is even, so that we can construct an interpolant of degree $q = p/2$ using the past derivative evaluations. When changing stepsize, we will use this interpolant to compute new values $f(\tilde{Y}_i^j)$ at appropriate

points of the past interval scaled to the adjustment in stepsize: that is, we expect approximations of order q to the solution values $f(y(x))$ of (1.1) at the points $x = x_i + (c_j - 1)h_{i+1}$, $j = 1, 2, \dots, s$, where $h_{i+1} = x_{i+1} - x_i$ is the new stepsize. Since we are interpolating only function values $f(\tilde{Y}_i^j)$ at past stages (but not solution values at the step points), the lower order $q = p/2$ of this interpolation does not prevent the overall method from attaining the order of convergence equal to p . This is confirmed, especially for pairs of orders 3 and 4, and 5 and 6, by the numerical experiments presented in Section 6. While there is some loss of order for pairs of orders 7 and 8 as discussed in Section 6, this is may be due to the limited accuracy available in MATLAB, or perhaps due to the structure of the family of pairs of orders 7 and 8 obtained.

A different approach to interpolation which utilizes TSRK methods of order p and stage order $q = p$ has been discussed in [5], [19], [23] and [24].

2 Solving the Order Conditions

In this section, we derive some families of pairs of orders $p - 1$ and p for $p = 4, 6, 8$, which may be used both to propagate the solution and to estimate the local error. Thus, we obtain vectors c , v , w , and matrices A , and B to yield a method of order p , together with two additional weight vectors \hat{v} and \hat{w} which will be used with the same stages to obtain a method of order $p - 1$. The parameters for these methods will be obtained by direct solutions of those order conditions tabulated in [18] and those which may be obtained from [7] when $p > 6$.

We begin by describing a general strategy for solving the order conditions. Assume that p is even. Let e be an s -vector with each element having a unit value. The consistency condition is $c = (A + B)e - u$. Let U and C be the diagonal matrices for which $u = Ue$ and $c = Ce$, respectively. For these, we define scalars

$$Q^{[k]} = \frac{1}{(k-1)!} \left(v^T (C - I)^{k-1} + w^T C^{k-1} + \frac{(-1)^k}{k} \theta - \frac{1}{k} \right) e, \quad (2.1)$$

$k = 1, 2, \dots, p$, which we call quadrature expressions since they must be annihilated to attain order p for a quadrature problem (1.1) in which one or more components depend only on the independent variable x . For linear and other problems, the required conditions rely significantly on values of

the subquadrature (vector) expressions

$$q^{[k]} = \frac{1}{(k-1)!} \left(A(C-I)^{k-1} + BC^{k-1} + \frac{(-1)^k}{k} U - \frac{C^k}{k} \right) e. \quad (2.2)$$

The proposed technique for solving the order conditions yields methods for which the stage order of all stages will be (at least) $p/2$. This constraint (which is the analog of Butcher's row simplifying assumptions [6]) makes many of the order condition equivalent. Furthermore, each step will use $p/2+1$ internal stages. This tactic was adopted with a view that a $(p/2+1)$ -node interpolant of propagated derivative evaluations will allow easy change of stepsize using the derivative evaluations from the previous step to construct an interpolant of degree $p/2$.

To attain the required quadrature order, it is necessary that

$$Q^{[k]} = 0, \quad (2.3)$$

$k = 1, 2, \dots, p$. To achieve the required stage order, it is necessary that

$$q^{[k]} = 0, \quad (2.4)$$

$k = 1, 2, \dots, q$, for $q \geq p/2$. The additional conditions that must be imposed to achieve order p for the method correspond uniquely to the set of all rooted trees (see [6]) each of which has no more than p nodes and as well a "bushy" branch of at least $q+1$ nodes. These additional order conditions will be stated separately for each family of pairs considered.

Solutions of these order conditions lead to formulas which yield approximate solutions of problems of the form (1.1), and strategies for deriving these solutions are adapted from techniques developed by Verner for explicit Runge–Kutta pairs [25], [26]. To obtain the coefficients of each family parametrically, we partition the order conditions, and solve the complete set by solving several subsystems separately. The strategy is to partition each set of order conditions so that most subsystems are linear in the variables evaluated. That is, we strive to obtain a sequence of linear problems to solve. This strategy is not always successful, but it does allow for a uniform derivation of the methods considered here.

In particular, we begin by assuming that the set of nodes c is arbitrary. If we choose these to be distinct and all lying within the interval $[0, 1]$, then (2.3) form linear conditions in the entries of v and w . These weights form 2s

parameters, and if we select $2s - p$ of them arbitrarily, the remaining p can be chosen uniquely to satisfy

$$\sum_{i=1}^s v_i (c_i - 1)^{k-1} + \sum_{i=1}^s w_i c_i^{k-1} - \frac{1}{k} = 0, \quad (2.5)$$

$k = 1, 2, \dots, p$, which is equivalent to (2.3).

Once v and w have been selected along with c , a similar strategy can be invoked to make the stage order equal to q . For each of the $p/2 + 1$ stages, we need to satisfy $q = p/2$ annihilating conditions from (2.4). Since there are p^2 elements in A , and $p > q$, this is a somewhat larger linear system in more unknowns than equations:

$$q_i^{[k]} \equiv \sum_{j=1}^s a_{ij} (c_j - 1)^{k-1} + \sum_{j=1}^s b_{ij} c_j^{k-1} - \frac{c_i^k}{k} = 0, \quad (2.6)$$

$k = 1, 2, \dots, p/2$, for each of the s rows of A and B . This is a system with the entries of A and B unknown, and provided that the matrix is nonsingular, the system could be solved to obtain expressions of a corresponding subset of $sp/2$ entries in A in terms of the remaining unknown values in A and B .

Next, the remaining order conditions will depend on v , w and values of A and possibly B already found, and are not always linear. For s large enough, there will be a sufficient number of unrestricted parameters to satisfy these remaining conditions, and it may be possible to solve them using an algebraic processing language such as Maple or Mathematica, if not directly. Finally, values of \hat{v} and \hat{w} must be selected so that an embedded method of order $p - 1$ is obtained.

The process described above leads to a family of embedded pairs of TSRK methods of required order and stage order which depend on some number of free parameters. The free parameters of the method of order p which will be used to advance the numerical solution, are then chosen to maximize the region of absolute stability of this method. The free parameters of the method of order $p - 1$ which will be used to estimate the local discretization error, are then chosen to achieve some balance between the quality of this error estimate and the efficiency of the overall numerical algorithm based on embedded pair of TSRK formulas. This will be discussed in more detail in the subsections below.

2.1 Derivation of methods with $s=3$, $p=4$, and $q=2$

For the first family, in addition to annihilating (2.1) and (1.4), we require one additional condition:

$$(v + w)^T q^{[3]} = 0. \quad (2.7)$$

For the 3 stages available, v and w provide six weights. With the three nodes, and v_3, w_3 chosen arbitrarily, the remaining four weights can be chosen to annihilate $Q^{[k]}$ in (2.1) for $k = 1, 2, 3, 4$. Similarly, with the last column of A and each entry of B arbitrary, we could choose the first two columns of A to annihilate $q^{[k]}$ in (2.1) for $k = 1, 2$. However, (2.7) involves all of v, w, A, B and c , and so we must consider this condition simultaneously. Since c, v and w have already been selected, we observe that (2.7) is now linear in A and B , and we may add one more equation to those imposing the stage order. This gives a system of seven linear conditions, which have a unique solution for most choices of the arbitrary parameters (provided that the system attempts to solve for one entry of matrix B). For example, one family of methods with three stages, order $p = 4$, stage order $q = 2$, has 10 arbitrary parameters: $c_1, c_2, c_3, v_3, w_3, a_{1,3}, a_{2,3}, a_{3,3}, b_{2,1}$, and $b_{3,1}$.

It remains to construct an embedded method of order $p = 3$ so that the local error may be estimated. For this, we have only the six coefficients in the weights \hat{v} and \hat{w} available, and one of these must be forced to be different than its corresponding value for the propagated method. Because the stage order is $q = 2$, it is sufficient that the weights satisfy the quadrature conditions (2.3) for $k = 1, 2, 3$. That is, these parameters need to be constrained only to satisfy three order conditions, and so we can choose two additional weights arbitrarily, and use the remaining three weights to satisfy the quadrature conditions. For example, by choosing $\hat{v}_3 \neq v_3$, and \hat{v}_2 and \hat{w}_3 arbitrarily, we could choose the remaining three coefficients to satisfy

$$\sum_{i=1}^3 \hat{v}_i (c_i - 1)^{k-1} + \sum_{i=1}^3 \hat{w}_i c_i^{k-1} = \frac{1}{k},$$

$k = 1, 2, 3$. This will yield a family of pairs with stage order $q = 2$, and orders 3 and 4 each of which may be implemented with a change of stepsize from step to step so that the local error may be controlled. This family has a total of 13 essentially arbitrary parameters which may be selected to enhance the accuracy of the error estimator, and/or the stability and the accuracy of the propagating method.

Other related methods may be constructed by simple modifications to this procedure. For example, three-stage pairs with stage order $q = 3$, and orders 3 and 4 may be obtained by imposing three more conditions on the three stages so that A and B annihilate $q^{[k]}$ for $k = 1, 2, 3$. As a consequence, (2.7) is identically zero, and so this constrains only two of the arbitrary parameters. In contrast to pairs with stage order $q = 2$, these pairs may yield more flexibility in changing stepsize. A family of these pairs can be represented in terms of 11 arbitrary parameters: $c_1, c_2, c_3, v_3, w_3, b_{2,1}, b_{3,1}, b_{3,2}, \hat{v}_3, \hat{w}_2$ and \hat{w}_3 .

2.2 Derivation of methods with $s=4$, $p=6$, and $q=3$

We begin again by choosing $p = 6$ of the $2s = 8$ weights from v and w to satisfy the quadrature conditions (2.3). Thus, for example, c, v_4 and w_4 may be chosen arbitrarily and the remaining entries of v and w may be selected to satisfy the quadrature order conditions.

Then we may consider the $sq = 12$ stage order conditions as constraints on the 16 entries of the matrix A . Thus, after satisfying (2.3) and (2.4), there remain 10 entries in A and B which might be used to satisfy the remaining order conditions. By applying the stage order $q = 3$ on expressions (2.2), we find that there are only four more conditions to satisfy. These may be written as

$$\begin{aligned}
 (v + w)^T q^{[4]} &= 0, \\
 (v + w)^T C q^{[4]} - v^T q^{[4]} &= 0, \\
 (v + w)^T q^{[5]} - v^T q^{[4]} &= 0, \\
 (v + w)^T (A + B) q^{[4]} &= 0.
 \end{aligned} \tag{2.8}$$

As v and w have been computed previously, we observe that the first three of these equations are linear in the entries of A and B . Together with the equations imposing the stage order, this would give a linear system in 15 equations, and these could be solved using only 15 of the entries in A if the matrix were nonsingular. Before proceeding, we consider the final equation (2.8) which is not linear in the elements of A and/or B . It might be possible to solve this as a quadratic equation within the system, but this would be inconvenient unless an algebraic processor could be used (an inefficient alternative if the free parameter set were to be searched for an optimal procedure). To simplify the role of this equation, we observe that it is necessary

that each of $(v+w)^T$, $(v+w)^T C - v^T$ and $(v+w)^T(A+B)$ must be orthogonal to $q^{[4]}$. Since solution of the system of 15 equations ensures that the first two of these orthogonality conditions holds, it would be sufficient to satisfy (2.8) if we required that

$$(v+w)^T(A+B) = \kappa(v+w)^T + \lambda\left((v+w)^T C - v^T\right)$$

for some constants κ and λ . This condition is linear in A , B , κ and λ , and so by constraining two more parameters of A and B , we can solve all of the remaining order conditions while leaving five parameters of A and B arbitrary in addition to c , v_4 and w_4 . It turns out that at least one of the entries of A must be chosen arbitrarily in order that the matrix of this system be nonsingular, but this is only a marginal restriction on the free parameters.

There remains only the construction of the embedded method of order $p = 5$. Because the stage order is $q = 3$, the weights of the embedded method must be selected to satisfy the quadrature conditions (2.3) for $k = 1, 2, 3, 4, 5$, and as well, the additional condition

$$(\hat{v} + \hat{w})^T q^{[4]} = 0.$$

Since the subquadrature expressions $q^{[k]}$ (2.2) are already known, this forms a system of six linear conditions on the eight weights of \hat{v} and \hat{w} . As well, at least one weight must differ from its analog in the weights for the propagated method, so there is no difficulty in solving this subsystem. The two essentially arbitrary weights of this embedded method may be utilized to enhance the accuracy of the error estimator. In summary, this formulation yields a parametric family in the 13 arbitrary parameters $c_1, c_2, c_3, c_4, v_4, w_4, a_{4,4}, b_{2,1}, b_{3,1}, b_{4,1}, b_{4,2}, \hat{v}_4$, and \hat{w}_4 .

2.3 Derivation of methods with s=5, p=8, and q=4

We have applied this strategy in an attempt to obtain a family of 5-stage pairs of stage order $q = 4$ and orders 7 and 8. This leads to a family of pairs of these orders when the analog

$$(v+w)^T(A+B)q^{[6]} = 0$$

of (2.7) is solved by making $(v+w)^T(A+B)$ a linear combination of $(v+w)^T$ and $(v+w)^T C - v^T$. However, the dependency of the order conditions satisfied

forces the stage order to be equal to 5. Perhaps with more study, these defects in the families may be overcome by developing different approaches to solving the order conditions. In the approach used, we obtained a parametric family in 15 parameters $c_1, c_2, c_3, c_4, c_5, v_5, w_5, a_{5,3}, b_{2,1}, b_{3,1}, b_{4,1}, b_{4,2}, b_{5,1}, \widehat{v}_5$ and \widehat{w}_5 .

2.4 Flexibility from other choices of the parameters

Values of the parameters θ and u different from 0 would allow more of the order conditions to be satisfied with a given number of stages. However, this would complicate the implementation of methods, and so we have assumed in this paper that each of these parameters is zero. The derivation of methods has been based on assuming values of the nodes $c_j, j = 1, 2, \dots, s$, can be assigned arbitrarily. (This is not necessarily the situation if one wishes to minimize the number of stages for a desired order, but for many families of methods, this is close to what is possible.) In this case, the quadrature conditions (2.3) are linear in v and w . Hence, the ‘‘quadrature’’ order can be at least $2s$ provided that the nodes $c_1 - 1, \dots, c_s - 1, c_1, \dots, c_s$ are distinct. Increases in the quadrature order are possible by selecting an appropriate value of θ probably different from zero, for example by making the nodes satisfy an orthogonality condition such as

$$\theta = \frac{\int_0^1 (c - c_1 + 1) \cdots (c - c_s + 1)(c - c_1) \cdots (c - c_s) dc}{\int_{-1}^0 (c - c_1 + 1) \cdots (c - c_s + 1)(c - c_1) \cdots (c - c_s) dc}.$$

3 Linear stability analysis of TSRK Methods

To facilitate an analysis of stability, define the vectors

$$Y = \begin{bmatrix} Y_1^{[i-1]} \\ \vdots \\ Y_s^{[i-1]} \\ Y_1^{[i]} \\ \vdots \\ Y_s^{[i]} \end{bmatrix}, \quad f(Y) = \begin{bmatrix} f(Y_1^{[i-1]}) \\ \vdots \\ f(Y_s^{[i-1]}) \\ f(Y_1^{[i]}) \\ \vdots \\ f(Y_s^{[i]}) \end{bmatrix}, \quad y^{[i]} = \begin{bmatrix} Y_1^{[i-1]} \\ \vdots \\ Y_s^{[i-1]} \\ y_{i-1} \\ y_i \end{bmatrix}.$$

Then the TSRK method (1.2) can be written as a general linear method [6] of the form

$$\begin{bmatrix} Y \\ y^{[i+1]} \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix} \begin{bmatrix} hf(Y) \\ y^{[i]} \end{bmatrix}, \quad (3.1)$$

$i = 1, 2, \dots, N - 1$, where the matrices A_1 , A_2 , B_1 , and B_2 are defined by

$$A_1 = \begin{bmatrix} 0 & 0 \\ A & B \end{bmatrix}, \quad B_1 = \begin{bmatrix} I & 0 & 0 \\ 0 & u & e - u \end{bmatrix},$$

$$A_2 = \begin{bmatrix} A & B \\ 0 & 0 \\ v^T & w^T \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & u & e - u \\ 0 & 0 & 1 \\ 0 & \theta & 1 - \theta \end{bmatrix}.$$

To study stability properties of (3.1) we employ the standard test equation

$$y' = \lambda y, \quad t \geq 0, \quad (3.2)$$

where λ is a complex parameter. It is easy to verify that the application of (3.1) to (3.2) leads to the recurrence relation

$$y^{[i+1]} = M(z) y^{[i]},$$

$i = 1, 2, \dots$, where $z = h\lambda$ and the matrix $M(z)$ is given by

$$M(z) = B_2 + zA_2(I_{2s} - zA_1)^{-1}B_1.$$

This matrix can be written in terms of u , A , B , θ , v and w as

$$M(z) = \begin{bmatrix} zS(z)A & S(z)u & S(z)(e - u) \\ 0 & 0 & 1 \\ z(v^T + zw^T S(z)A) & \theta + zw^T S(z)u & 1 - \theta + zw^T S(z)(e - u) \end{bmatrix},$$

where $S(z) = (I_s - zB)^{-1}$. As $\theta = u = 0$, we can omit y_{i-1} in the definition of the vector $y^{[i]}$, and the matrix $M(z)$ assumes the simpler form

$$\widehat{M}(z) = \begin{bmatrix} z(I_s - zB)^{-1}A & (I_s - zB)^{-1}e \\ z(v^T + zw^T(I_s - zB)^{-1}A) & 1 + zw^T(I_s - zB)^{-1}e \end{bmatrix}.$$

The stability properties of (3.1) for $\theta = 0$ and $u = 0$ with respect to (3.2) are determined by the location of the roots of the stability polynomial $p(w, z)$ defined by

$$p(\omega, z) = \det(\omega I_{s+1} - \widehat{M}(z)).$$

Since B is strictly lower triangular, it can be verified that $p(w, z)$ is a polynomial with the form

$$p(\omega, z) = \omega^{s+1} - p_1(z)\omega^s + \cdots + (-1)^{s+1}p_{s+1}(z),$$

where $p_k(z)$ are polynomials in z . The region of absolute stability of the method (3.1) is the set of all $z \in C$ such that all the roots $\omega_k(z)$ of $p(\omega, z)$ are inside of the unit circle, i.e.,

$$\mathcal{A} = \{z \in C : |\omega_k(z)| < 1, \quad k = 1, 2, \dots, s+1\}. \quad (3.3)$$

In Section 4 we will present some examples of TSRK methods obtained by optimizing the region of absolute stability \mathcal{A} defined by (3.3).

4 Selected Pairs of TSRK Methods

To demonstrate that the claimed orders can be realized, we selected some pairs with optimally selected arbitrary parameters for implementation.

Using the techniques from Section 2, we selected the arbitrary parameters from A and B to (nearly) maximize the size of the region of absolute stability, and we used the remainder of the parameters to minimize the error coefficient of order h^{p+1} in the error estimating formula. The motivation for the latter was to minimize the contamination in the error estimating formula.

Each pair was implemented with “error per step” and the solution was propagated by extrapolation using the higher order formula of the pair. This implementation makes the global error proportional to the tolerance, even though the local error is that of the lower order formula (compare [22]).

The selected pairs are displayed using a notation that is slightly modified from that of (1.3): since both θ and u have been selected equal to zero we omit them from the tables, but for completeness in characterizing each pair, we include the entries of the vector c .

A three-stage pair of orders 3 and 4 with stage order 2 denoted by TSRK4(3)2 is

c	A	B	0	$-\frac{1}{15}$	$\frac{2}{15}$	$-\frac{1}{15}$			
	v^T	w^T	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{2}$		
	\hat{v}^T	\hat{w}^T	1	$\frac{283}{720}$	$-\frac{73}{80}$	$\frac{11}{24}$	$\frac{3}{16}$	$\frac{629}{720}$	
				$\frac{1}{6}$	$-\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	0	$\frac{1}{3}$
				$\frac{19}{21}$	$-\frac{71}{42}$	$\frac{5}{8}$	$-\frac{137}{168}$	$\frac{17}{6}$	$-\frac{6}{7}$

Table 1: A three-stage pair of orders 3 and 4

The method of order four was obtained by using the remaining free parameters to maximize the region of absolute stability \mathcal{A} given by (3.3). This region is displayed in Fig. 1 (thick line) together with the region of absolute stability of *the* traditional four-stage RK method of order four (thin line). We observe that even with fewer stages, the regions are approximately the same size, although the new pair is more favourable towards problems with real eigenvalues.

A three-stage pair of orders 3 and 4 with stage order 3 denoted by TSRK4(3)3 is

c	A	B	$\frac{1}{10}$	$\frac{17}{2160}$	$-\frac{29}{1200}$	$\frac{157}{1350}$			
	v^T	w^T	$\frac{1}{2}$	$\frac{463}{2160}$	$-\frac{131}{240}$	$\frac{103}{270}$	$\frac{9}{20}$		
	\hat{v}^T	\hat{w}^T	1	$\frac{17}{36}$	$-\frac{181}{180}$	$\frac{23}{45}$	$\frac{2}{9}$	$\frac{4}{5}$	
				$\frac{295}{1344}$	$-\frac{43}{64}$	$\frac{7}{12}$	$\frac{115}{192}$	$-\frac{85}{1344}$	$\frac{1}{3}$
				$\frac{127}{1056}$	$-\frac{599}{3168}$	$-\frac{1}{4}$	$\frac{757}{1584}$	$\frac{3}{4}$	$\frac{1}{11}$

Table 2: A three-stage pair of orders 3 and 4

A four-stage pair of orders 5 and 6 with stage order 3 denoted by TSRK6(5)3

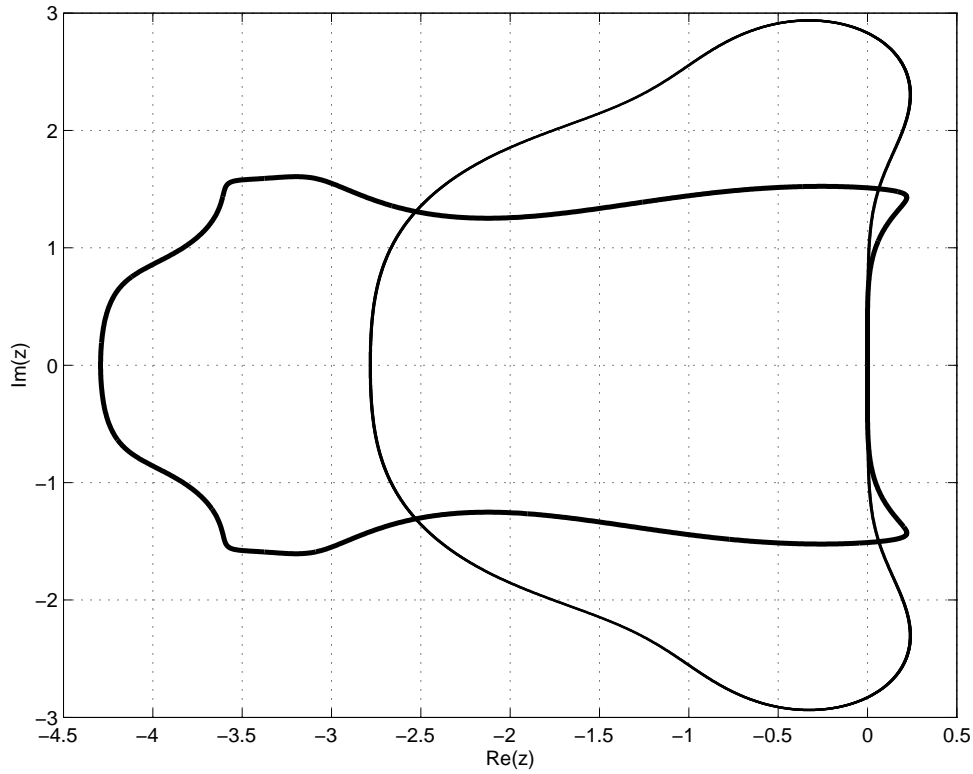


Figure 1: Regions of absolute stability of TSRK method of order four (thick line) and RK method of order four (thin line)

is

0	-0.0040	0.0159	-0.0165	0.0046				
$\frac{23}{60}$	-0.0591	0.5042	-0.9294	0.2176	$\frac{13}{20}$			
$\frac{13}{20}$	-0.2384	1.3269	-1.8617	0.4344	$\frac{5}{8}$	0.3638		
1	-0.5532	2.7754	-3.3026	$\frac{4}{5}$	$\frac{1}{7}$	$\frac{8}{9}$	0.2486	
	-0.0857	0.7684	-1.6864	$\frac{13}{50}$	1.8395	-1.3394	1.2353	$\frac{1}{120}$
	-0.1665	1.2154	-2.2515	$\frac{1}{10}$	2.0506	-0.8482	0.8002	$\frac{1}{10}$

Table 3: Approximate coefficients for a four-stage pair of orders 5 and 6

A five-stage pair of orders 7 and 8 with stage order 5 denoted by TSRK8(7)5

is

0	0	0	0	0	0					
$\frac{1}{4}$	0.0721	-0.3547	0.6675	-0.8551	0.2202	$\frac{1}{2}$				
$\frac{1}{2}$	0.0234	-0.0455	-0.1063	0.6124	-1.0519	$\frac{1}{3}$	0.7345			
$\frac{4}{5}$	0.0046	-0.1366	0.6797	-2.3787	4.4585	-1	-2	1.1725		
1	-0.5110	2.2684	-3	-1.6650	3.8330	4	-7.2701	3.3367	0.0080	
	0.0455	-0.3517	1.1203	-3.0814	$\frac{1}{6}$	4.0279	-2.5316	1.4992	-0.0200	$\frac{1}{8}$
	-0.1733	1.0430	-2.3831	3.4367	1	-2.7201	-0.8302	1.7979	-0.4208	$\frac{1}{4}$

Table 4: Approximate coefficients for a five-stage pair of orders 7 and 8

The coefficients of the methods displayed were selected first by optimizing properties as described in the following paragraph, and then selecting rational coefficients nearby so that Tables 1 and 2 tabulate exact coefficients. For pairs of orders 5 and 6, all choices of the parameters would require large numbers of digits if exact representation were used. For the pair selected, the arbitrary parameters are displayed as rationals, and the determined coefficients are indicated by four decimal-digit approximations. For pairs of orders 7 and 8, b_{54} is the solution of a quadratic, and so representation as rationals is not always possible. For the optimal pair found, arbitrary coefficients were selected to be rationals, and are shown as such in Table 4. Again, all other coefficients (constrained to satisfy the order conditions) are approximated as four-digit decimals. Accurate versions of these coefficients may be computed using the algorithms described above with the arbitrary parameters shown, or else may be obtained by e-mail from the authors. The numerical experiments which follow later use the exact coefficients for each of TSRK4(3)2 and TSRK4(3)3; for the other two pairs, accurate (but inexact) coefficients were computed from exact rational choices for the arbitrary parameters.

As mentioned before, in the first pair the arbitrary parameters were selected to optimize the region of absolute stability of the propagating method (of order 4). We will see in Section 6 that this pair provides exceptionally accurate estimates of the local error for most of the problems tested. However, this accuracy is realized at the expense of a larger number of function evaluations than is usually incurred by conventional numerical schemes of

the same orders of accuracy. In contrast, each stage of the second pair has stage order 3, and this makes it more efficient when the same stepsize selection strategy is used, yet the global error and the local error estimates are larger than for the first pair. The parameters of the third pair were again selected to maximize the region of absolute stability, and the weights of the error estimating method were chosen so that the number of steps was not too large. For the pair of orders 7 and 8, the arbitrary parameters were selected to minimize the largest parameter in the expectation that this would make the error smaller. The higher order pairs yielded somewhat less reliable error estimation than the pairs of order 3 and 4.

5 Implementation Issues

We choose the initial stepsize h_1 following the approach described in [11] (compare also [5]). After the initial stepsize is selected, we compute y_1 and stage values Y_0^j , $j = 1, 2, \dots, s$, on the first step $[x_0, x_1]$ by a continuous RK method of order 5 constructed by Owren and Zennaro [21]. (The coefficients of this method are not reproduced here, but they can be found in [21] or [5].) After the first step computed by the continuous RK method is accepted (possibly after some initial stepsize adjustment), we define $h_2 = h_1$ and continue the integration using a TSRK pair of order p . Each step of the propagation begins with a trial step of length h_{i+1} from x_i to x_{i+1} . The estimate $est(x_{i+1})$ of local discretization error at x_{i+1} is obtained by comparing the approximations of order $p - 1$ and p . Then, again following the approach described in [11] we compute the quantity

$$err = \| est(x_{i+1}) \|_{sc},$$

where

$$\| y \|_{sc} = \sqrt{(1/s) \sum_{j=1}^s (y_j^2 / sc_j^2)}$$

with $sc_j = Atol_j + \max\{|y_{i,j}|, |y_{i+1,j}|\} Rtol_j$, $j = 1, 2, \dots, s$, and compare it to 1 in order to determine the optimal stepsize

$$h_{opt} = h_{i+1} r_{opt}, \quad r_{opt} = (1/err)^{1/(p+1)}.$$

(i) If $err \leq 1$, the step is accepted and a new stepsize h_{i+2} is computed from the formula

$$h_{i+2} = h_{i+1} \min\{\delta_{max}, \max\{\delta_{min}, \delta \cdot r_{opt}\}\},$$

with $\delta = 0.9$, $\delta_{min} = 0.1$, and $\delta_{max} = 2$. We then compute the rescaled values $f(\tilde{Y}_i^j)$, corresponding to the points $\{x_{i+1} + (c_j - 1)h_{i+2}, j = 1, 2, \dots, s\}$, using the polynomial that interpolates to $f(Y_i^j)$ at $x_i + c_j h_{i+1}$. Since we interpolate directly function values $f(Y_i^j)$, this process does not require any extra evaluations of the function f . We continue the integration using the formula of order p obtained using local extrapolation.

(ii) If $err > 1$, the step is rejected and computations are repeated with a new stepsize \tilde{h}_{i+1} equal to

$$\tilde{h}_{i+1} = h_{i+1} \min\{\delta_{max}, \max\{\delta_{min}, \delta \cdot r_{opt}\}\},$$

and back values $f(\tilde{Y}_{i-1}^j)$ corresponding to the points $x_i + (c_j - 1)\tilde{h}_{i+1}$, $j = 1, 2, \dots, s$. These values are computed using the polynomial that interpolates $f(Y_{i-1}^j)$ at $\{x_{i-1} + c_j h_i, j = 1, \dots, s\}$ if $i \geq 2$ or by the continuous RK method if $i = 1$.

6 Numerical Experiments

To test the order of the derived methods and the quality of local error estimation, we have applied the embedded pairs listed in Section 4 to many DETEST problems listed in [13]. We present below the results at $x_{end} = 20$ on the problem B5:

$$\begin{cases} y_1' = y_2 y_3, & y_1(0) = 0, \\ y_2' = -y_1 y_3, & y_2(0) = 1, \\ y_3' = -0.51 y_1 y_2, & y_3(0) = 1, \end{cases}$$

which are Euler equations of motion of a rigid body without external forces, and at $x_{end} = 20$ on the problem E3:

$$\begin{cases} y_1' = y_2, & y_1(0) = 0, \\ y_2' = y_1^3/6 - y_1 + 2 \sin(2.78535x), & y_2(0) = 0, \end{cases}$$

which is derived from Duffing's equation $y'' + y - y^3/6 = 2 \sin(2.78535x)$.

The following global errors were obtained using MATLAB 5 on a Laptop Windows installation. They were different than that obtained from a UNIX MATLAB 6. (For publication I want to use the latter.)

<i>tol</i>	Problem B5					Problem E3				
	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>
10^{-6}	690	1	2102	1.3e-7		1404	0	4238	3.5e-7	
10^{-7}	1227	1	3713	1.1e-8	4.00	2490	0	7496	3.7e-8	4.02
10^{-8}	2182	1	6578	1.1e-9	4.00	4421	0	13289	3.8e-9	4.01
10^{-9}	3880	1	11672	1.2e-10	4.00	7857	0	23597	3.8e-10	4.00
10^{-10}	6903	2	20747	4.7e-12	4.00	13969	0	41933	3.9e-11	4.00

Table 5: Numerical results for the pair TSRK4(3)2.

<i>tol</i>	Problem B5					Problem E3				
	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>
10^{-6}	378	0	1160	9.5e-7		772	0	2342	2.4e-6	
10^{-7}	672	1	2048	4.0e-8	4.00	1366	0	4124	2.5e-7	4.03
10^{-8}	1195	1	3617	3.6e-9	4.00	2423	0	7295	2.5e-8	4.02
10^{-9}	2126	1	6410	4.2e-10	4.00	4303	0	12935	2.5e-9	4.01
10^{-10}	3781	1	11375	5.2e-11	4.00	7647	0	22967	2.5e-10	4.00

Table 6: Numerical results for the pair TSRK4(3)3.

The results of numerical simulations with the embedded pairs TSRK4(3)2, TSRK4(3)3, TSRK6(5)3 and TSRK8(7)5 are presented in Tables 5–8. In these tables, *ns* stands for the number of steps, *nr* for the number of rejected steps, *nfe* for the number of function evaluations, *ge* for the global error at the end point of integration, and *p* for the average order of convergence given by

$$p = \log(tol_1/tol_2)/\log(ns_2/ns_1). \quad (6.1)$$

On the assumption that the global error is proportional to the tolerance, this

<i>tol</i>	Problem B5					Problem E3				
	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>
10 ⁻⁶	151	0	630	8.3e-6		250	0	1026	9.6e-7	
10 ⁻⁷	221	0	910	1.0e-6	6.04	370	15	1566	1.4e-6	5.87
10 ⁻⁸	324	0	1322	1.5e-7	6.02	557	9	2290	1.2e-7	5.62
10 ⁻⁹	480	0	1946	2.0e-8	5.85	886	50	3770	5.7e-9	4.96
10 ⁻¹⁰	733	0	2958	3.0e-9	5.43	1408	203	6470	9.7e-10	4.97

Table 7: Numerical results for the pair TSRK6(5)3.

<i>tol</i>	Problem B5					Problem E3				
	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>	<i>ns</i>	<i>nr</i>	<i>nfe</i>	<i>ge</i>	<i>p</i>
10 ⁻⁶	125	8	691	2.7e-6		182	20	1036	2.0e-6	
10 ⁻⁷	156	7	841	1.8e-7	10.39	231	12	1241	5.5e-7	9.65
10 ⁻⁸	203	7	1076	2.6e-8	8.74	325	11	1706	6.8e-8	6.74
10 ⁻⁹	285	7	1486	2.1e-9	6.78	462	22	2446	1.8e-9	6.54
10 ⁻¹⁰	409	138	2136	2.7e-10	6.37	667	29	3506	1.7e-10	6.27

Table 8: Numerical results for the pair TSRK8(7)5.

formula is based on the assumptions that

$$global\ error = K h^p = K \left(\frac{x_{end} - x_0}{ns} \right)^p$$

and that

$$global\ error = C tol.$$

In the experiments described, the global error is almost proportional to the tolerance except for results from TSRK6(5)3. We could modify the formula for estimating the order to accommodate the differences, but this tends to exaggerate differences in the estimates from one tolerance to the next, from

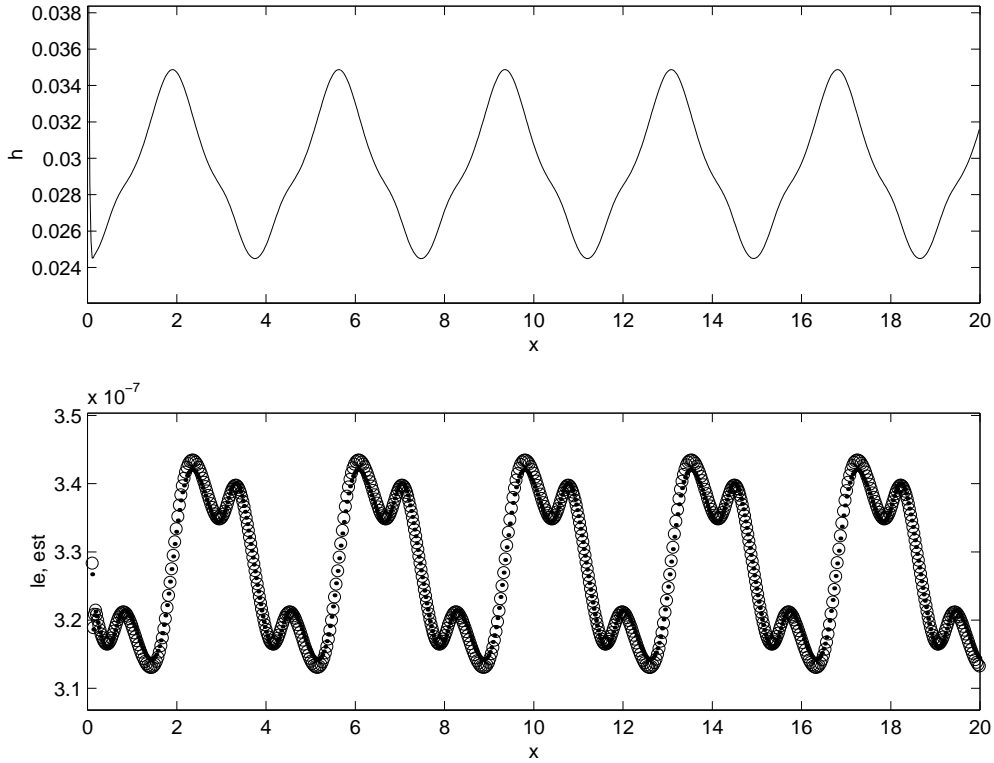


Figure 2: Embedded pair TSRK4(3)2 on problem B5, $tol = 10^{-6}$. Step-size distribution with rejected steps, contrast of local error with local error estimate

the estimates given by (6.1), but the average is much the same. We can observe that the first three pairs achieve the expected order of convergence in spite of the fact that the order of interpolation is only equal to the stage order q . For pairs TSRK4(3)2 and TSRK4(3)3, the rejected steps were all due to an inadequate stepsize choice for the second step in the starting procedure. For pairs of orders 7 and 8, there were a large number of step rejections in many cases. The choice of a pair with a minimized maximum coefficient reduced the number of rejections considerably. Additional reduction of rejected steps was achieved by linearly reducing the stepsize whenever the stepsize is reduced naturally by the algorithm for step control. Perhaps the high order of these 7,8 pairs cannot be exploited by using only the limited accuracy available in MATLAB for computing the approximate solutions. This might also explain

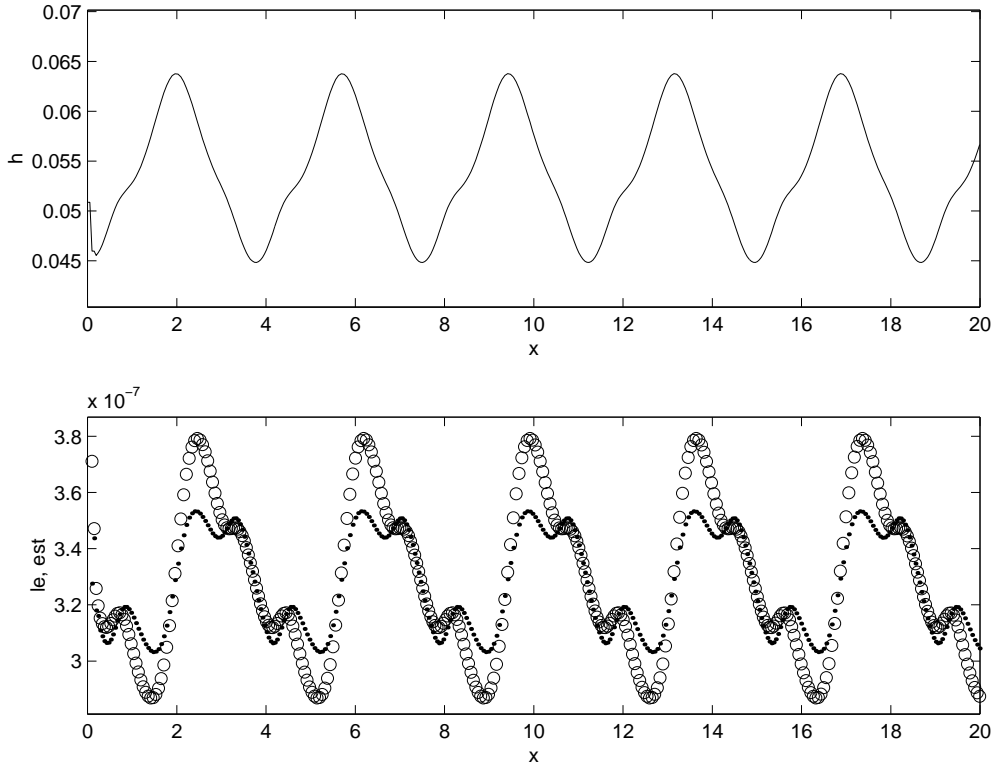


Figure 3: Embedded pair TSRK4(3)3, problem B5, $tol = 10^{-6}$.

the poorer asymptotic performance for severe tolerances.

To illustrate the quality of local error estimation, we have plotted in Fig. 2–5 the observed stepsize pattern (upper graphs), and local errors ‘o’ and local error estimates ‘.’ (lower graphs). There is an exceptionally good agreement between local error and local error estimate for the pair TSRK4(3)2 which was achieved at the expense of increased number of steps ns and function evaluations nfe . By choosing different weights \hat{v} and \hat{w} of the error estimating formula (1.5), it is possible to construct more efficient pairs such as TSRK4(3)3, but then the agreement between the local error and the local error estimate deteriorates somewhat (compare Fig. 3). In constructing the pair TSRK6(5)3 we were just aiming at efficiency, and this resulted in somewhat less reliable error estimation than for the pairs TSRK4(3)2 and TSRK4(3)3. In obtaining the pair TSRK8(7)5, we found many steps were rejected when the derivatives became large: this may be considerably reduced

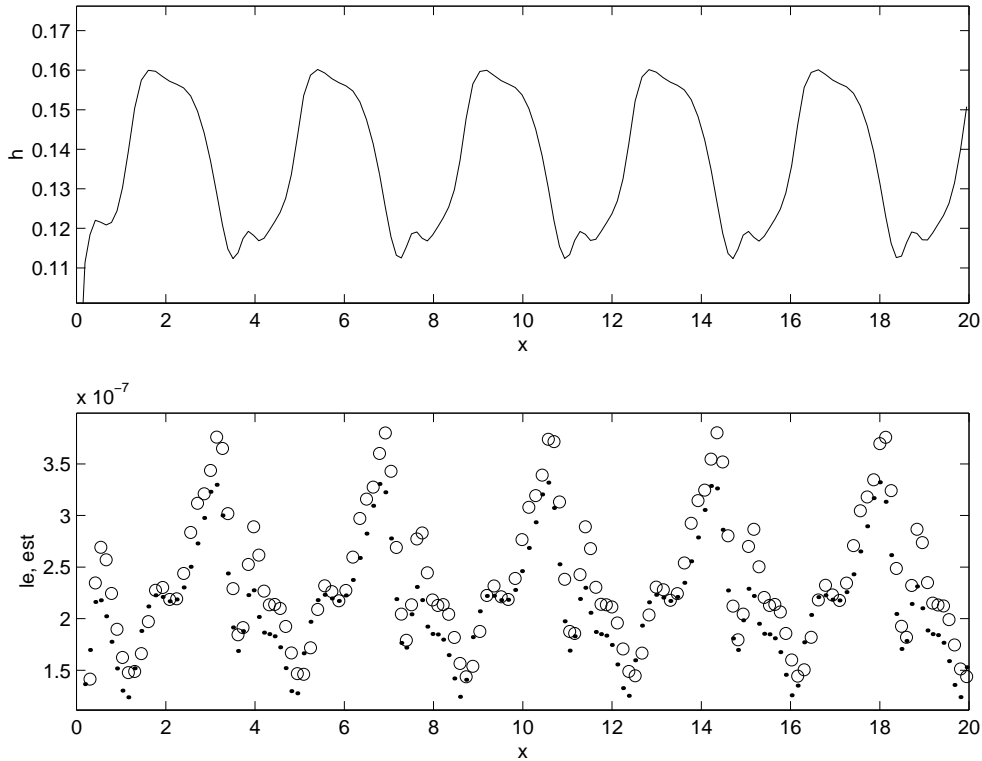


Figure 4: Embedded pair TSRK6(5)3, problem B5, $tol = 10^{-6}$.

by extrapolating the reduction in stepsize linearly from the previous step whenever the stepsize decreases. In the upper graph of Fig. 5 the rejected steps are indicated by the symbol ‘ \times ’.

7 Concluding Remarks

In this paper we have developed an alternative strategy for implementing TSRK methods to control error by estimating the local error and changing the stepsize accordingly. This is in contrast to the approach by Bartoszewski and Jackiewicz [5] and Tracogna and Welfert [24] in which the coefficients are functions of the stepsize ratio from one step to the next. To achieve the present approach, the parameters θ and u have to be restricted to be equal to zero at the outset in order that an interpolant of the appropriate structure is available with a marginal additional cost.

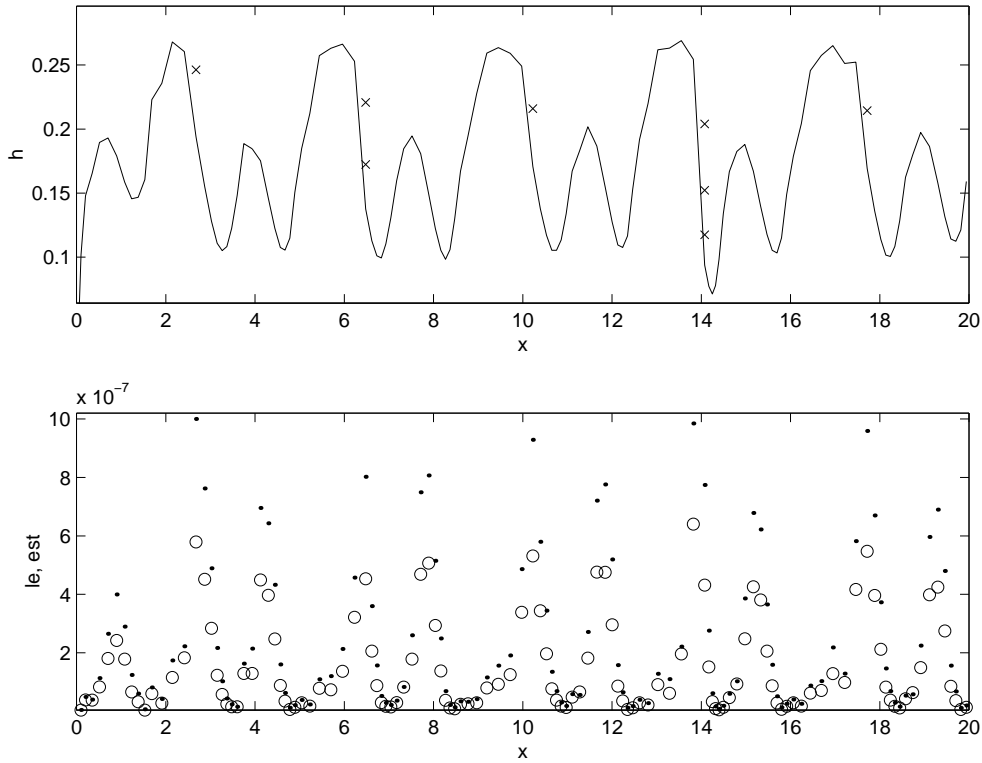


Figure 5: Embedded pair TSRK8(7)5, problem B5, $tol = 10^{-6}$.

With this beginning, we have been able to satisfy the order conditions with a satisfactory number of stages, and obtained parametric families of pairs which are easily implemented with stepsize change. (Incidentally, it may be possible to implement these methods slightly more efficiently by absorbing the interpolation mechanism into the matrix A and the vector v which provide the computations required for a stepsize change.) Perhaps the more general methods than those considered in this paper will be implementable with additional work.

The most satisfactory feature of this study is the exceptional accuracy of the local error estimator over a broad range of problems. This feature may be particularly attractive to applications in celestial mechanics and robotics where attainable high accuracy may be more significant than efficiency.

Acknowledgments. The authors wish to express their gratitude to anonymous referees for their constructive criticism.

References

- [1] P. Albrecht, *A new theoretical approach to Runge–Kutta methods*, SIAM J. Numer. Anal. 24 (1987), pp. 391–406.
- [2] P. Albrecht, *Consequences of a linear approach to Runge–Kutta methods*, manuscript.
- [3] P. Albrecht, *The Runge–Kutta theory in a nutshell*, SIAM J. Numer. Anal. 33 (1996), pp. 1712–1735.
- [4] Z. Bartoszewski and Z. Jackiewicz, *Construction of two-step Runge–Kutta methods of high order for ordinary differential equations*, Numerical Algorithms 18 (1998), pp. 51–70.
- [5] Z. Bartoszewski and Z. Jackiewicz, *Toward a two-step Runge–Kutta code for nonstiff differential systems*, to appear in Appl. Math.
- [6] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, John Wiley, New York 1987.
- [7] J.C. Butcher and S. Tracogna, *Order conditions for two-step Runge–Kutta methods*, Appl. Numer. Math. 24 (1997), pp. 351–364.
- [8] G.D. Byrne and R.J. Lambert, *Pseudo-Runge–Kutta methods involving two points*, J. Assoc. Comput. Mach. 13 (1966), pp. 114–123.
- [9] R. Cairra, C. Costabile and F. Costabile, *A class of pseudo Runge–Kutta methods*, BIT 30 (1990), pp. 642–649.
- [10] N. huu Cong and T. Mitsui, *Collocation-based two-step Runge–Kutta methods*, Japan J. Industr. Appl. Math. 13 (1996), pp. 171–183.
- [11] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I, Nonstiff Problems*, Springer-Verlag, Berlin, Heidelberg, New York 1991.

- [12] E. Hairer and G. Wanner, *Order conditions for general two-step Runge–Kutta methods*, SIAM J. Numer. Anal. 34 (1997), pp. 2087–2089.
- [13] T.E. Hull, W.H. Enright, B.M. Fellen and A.E. Sedgwick, *Comparing numerical methods for ordinary differential equations*, SIAM J. Numer. Anal. 9 (1972), pp. 603–637.
- [14] Z. Jackiewicz, R. Renaut, and A. Feldstein, *Two-step Runge–Kutta methods*, SIAM J. Num. Anal. 28 (1991), pp. 1165–1182
- [15] Z. Jackiewicz, R. Renaut, and M. Zennaro, *Explicit two-step Runge–Kutta methods*, Appl. Math. 40 (1995), pp. 433–456.
- [16] Z. Jackiewicz and R. Vermiglio, *General linear methods with external stages of different orders*, BIT 36 (1996), pp. 688–712.
- [17] Z. Jackiewicz and S. Tracogna, *A representation formula for two-step Runge–Kutta methods*. In: Proc. of the Second Hellenic European Conference on Mathematics and Informatics (E. Lipitakis, ed.), September 22-24, 1994, pp. 111–120, Hellenic Mathematical Society, Athens, Greece, 1994.
- [18] Z. Jackiewicz and S. Tracogna, *A general class of two-step Runge–Kutta methods for ordinary differential equations*, SIAM J. Num. Anal. 32 (1995), pp. 1390–1427.
- [19] Z. Jackiewicz and S. Tracogna, *Variable stepsize continuous two-step Runge–Kutta methods for ordinary differential equations*, Numerical Algorithms 12 (1996), pp. 347–368.
- [20] Z. Jackiewicz and M. Zennaro, *Variable-stepsize explicit two-step Runge–Kutta methods*, Math. Comp. 59 (1992), pp. 421–438.
- [21] B. Owren and M. Zennaro, *Derivation of efficient, continuous, explicit Runge–Kutta methods*, SIAM J. Sci. Stat. Comput. 13 (1992), pp. 1488–1501.
- [22] L.F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, London 1994.

- [23] S. Tracogna, *Implementation of two-step Runge–Kutta methods for ordinary differential equations*, J. Comp. Appl. Math. 76 (1996), pp. 113–136.
- [24] S. Tracogna and B. Welfert, *Two-step Runge–Kutta methods. Theory and practice*, BIT 40, (2000), pp. 775-799.
- [25] J.H. Verner, *Strategies for deriving new explicit Runge–Kutta pairs*, Annals of Numerical Mathematics 1, (1994), pp. 225–244.
- [26] J.H. Verner, *High order explicit Runge–Kutta pairs with low stage order*, Applied Numerical Mathematics, 20 (1996), 1–13.