RUNGE-KUTTA PAIRS FOR SCALAR AUTONOMOUS INITIAL VALUE PROBLEMS

G. PAPAGEORGIOU and Ch. TSITOURAS*

Department of Applied Mathematics and Physical Sciences, National Technical University of Athens, Zografou Campus 15780, Athens, Greece

We present the equations of condition up to sixth order for Runge-Kutta (RK) methods, when integrating scalar autonomous problems. Two RK pairs of orders 5(4) are derived. The first at a cost of only five stages per step, while the other having an extremely small principal truncation error. Numerical tests show the superiority of the new pairs over traditional ones.

Keywords: Order conditions; RK5(4) pairs

AMS Classification: 65L05; 65L06

C.R. Categories: G.1.7

1 INTRODUCTION

The scalar autonomous initial value problem has the form

$$y' = f(y), \quad y(t_0) = y_0$$

with $f : \Re \to \Re$.

This problem is a special case of the general class of initial value problems of first order:

$$y' = f(x, y), \quad y(x_0) = y_0 \in \Re^m, \ x \in [x_0, x_e], \tag{1}$$

where $f : \Re \times \Re^m \to \Re^m$.

Runge-Kutta (RK) pairs are widely used for the numerical solution of the initial value problem (1). These pairs are characterized by the extended Butcher tableau [1, 2]:

$$\begin{array}{c|c} c & A \\ \hline & b \\ & \hat{b} \end{array}$$

with $b^T, \hat{b}^T, c \in \Re^s$ and $A \in \Re^{s\times s}$ is strictly lower triangular. The procedure that advances the solution from (x_n, y_n) to $x_{n+1} = x_n + h_n$ computes at each step two approximations $y_{n+1}, \hat{y}_{n+1}$ to $y(x_{n+1})$ of orders p and $p-1$, respectively, given by

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i f_{ni} \tag{2}$$

and

$$\hat{y}_{n+1} = y_n + h_n \sum_{i=1}^{s} \hat{b}_i f_{ni}$$

with

$$f_{ni} = f\left(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{ij} f_{nj}\right) \tag{3}$$

for $i = 1, 2, \ldots, s \geq p$. In the following we use letters with cups to denote quantities pertaining to the lower-order method of a pair. The methods studied in this article obey the simplifying assumption $A \cdot e = c$, $e = \underbrace{(1, 1, \ldots, 1)^T}_{s}$.

From this embedded form (called RK$p(p-1)$) we can obtain an estimate

$$E_{n+1} = \|y_{n+1} - \hat{y}_{n+1}\|$$

of the local truncation error of the $p-1$ order formula. So the step-size control algorithm

$$h_{n+1} = 0.9 \cdot h_n \cdot \left(\frac{\mathrm{TOL}}{E_{n+1}}\right)^{1/p},$$

is in common use, with TOL being the requested tolerance. The above formula is used even if TOL is exceeded by E_{n+1}, but then h_{n+1} is simply the recomputed current step. See [16] for more details on the implementation of these type of step size policies.

2 TREES THEORY FOR RK METHODS

2.1 Taylor series expansions

Setting $x' = 1$, then problem (1) reduces, without loss of generality, to the more convenient autonomous problem $y' = f(y)$. When advancing the p-order RK method (2) and (3), applied to the latter problem, we actually try to approximate the corresponding Taylor method of the form

$$y(x_{n+1}) \approx y(x_n) + hy'(x_n) + \frac{1}{2!}h^2y''(x_n) + \cdots + \frac{1}{p!}h^py^{(p)}(x_n). \tag{4}$$

On the other hand we may expand f_{ni} around the point (x_n, y_n) and derive from (2) the expression

$$y_{n+1} = y_n + hq_{11}y'_n + h^2q_{21}y''_n + h^3(q_{31}f'f + q_{32}f''f^2) + \cdots \tag{5}$$

with q_{ij} depending exclusively on the coefficients A, b, c.

Verify now,

$$\begin{aligned} y'' &= \frac{\partial f(y(x))}{\partial x} = \frac{\partial f}{\partial y}f = f'f, \\ y''' &= \frac{\partial^2 f}{\partial y^2}\cdot(f, f) + \frac{\partial f}{\partial y}\cdot\frac{\partial f}{\partial y}\cdot f = f''(f, f) + f'f'f, \\ y^{(4)} &= \frac{\partial^3 f}{\partial y^3}\cdot(f, f, f) + \frac{\partial f}{\partial y}\cdot\frac{\partial f}{\partial y}\cdot\frac{\partial f}{\partial y}\cdot f + \frac{\partial f}{\partial y}\cdot\frac{\partial^2 f}{\partial y^2}\cdot(f, f) + 3\cdot\frac{\partial^2 f}{\partial y^2}\cdot\left(\frac{\partial f}{\partial y}\cdot f, f\right) \\ &= f'''(f, f, f) + f'f'f'f + f'f''(f, f) + 3f''(f'f, f), \\ &\ldots, \end{aligned}$$

where the elementary differentials $f''(f, f)$, $f'''(f, f, f)$, $f'f''(f, f)$, $f''(f'f, f)$ are Frechet derivatives [9, pp 158].

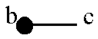
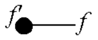
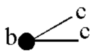
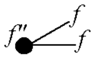
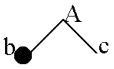
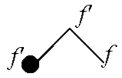
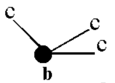
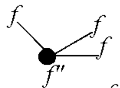
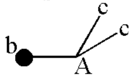
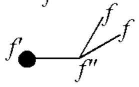
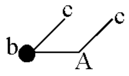
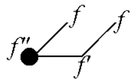
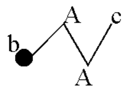
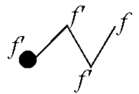
After matching (4) and (5) we arrive at

$$\begin{aligned} y(x_{n+1}) - y_{n+1} = {}& h(q_{11} - 1)f + h^2\left(q_{21} - \frac{1}{2}\right)\frac{\partial f}{\partial y}f \\ &+ h^3\left(\left(q_{31} - \frac{1}{6}\right)f^T\cdot\frac{\partial^2 f}{\partial y^2}\cdot f + \left(q_{32} - \frac{1}{6}\right)\frac{\partial f}{\partial y}\cdot\frac{\partial f}{\partial y}\cdot f\right) + \cdots. \end{aligned} \tag{6}$$

So requiring $t_{11} = q_{11} - 1 = 0$, $t_{21} = q_{21} - \frac{1}{2} = 0$, $t_{31} = q_{31} - \frac{1}{6} = 0$, $t_{32} = q_{32} - \frac{1}{6} = 0$ we conclude to the order conditions we have to satisfy for constructing a third order method. The order conditions up to fourth order are listed in first column of Table I.

In this table we denote by c^i the componentwise multiplication $\underbrace{c\cdot c\cdots c}_{i}$ (we assume $c^0 = e$), for which we allow a higher order of precedence over the regular (matrix-to-matrix or matrix-to-vector) multiplication (dot product). Moreover, the same symbol will be used here to denote both type of multiplication. Whenever both type of multiplication are found simultaneously in a relation and there is a possible conflict, we distinguish the order of precedence, by the proper use of parenthesis. We also define $C = \mathrm{diag}(c)$.

TABLE I The equations of condition of RK methods, for orders 1–4

Equation	*Equation tree-form*	*Elementary differential*
$t_{11} = b \cdot e - 1$		
$t_{21} = b \cdot c - \frac{1}{2}$		
$t_{31} = \frac{1}{2} b \cdot c^2 - \frac{1}{6}$		
$t_{32} = b \cdot A \cdot c - \frac{1}{6}$		
$t_{41} = \frac{1}{6} b \cdot c^3 - \frac{1}{24}$		
$t_{42} = \frac{1}{2} b \cdot A \cdot c^2 - \frac{1}{24}$		
$t_{43} = b \cdot C \cdot A \cdot c - \frac{1}{8}$		
$t_{44} = b \cdot A^2 \cdot c - \frac{1}{24}$		

2.2 Trees and rooted trees

Equation (6) has the form

$$y(x_{n+1}) - y_{n+1} = \sum_{i=1}^{\infty} \sum_{\tau \in T_i} h^i \frac{1}{\sigma(\tau)} \left(\Phi(\tau) - \frac{1}{\gamma(\tau)} \right) F(\tau),$$

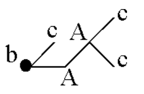
where T_i is the set of rooted trees of order i [13], σ, γ are integer-valued functions of τ, Φ is a certain composition of A, b, c, the skeleton of which depends only on τ and F is an elementary differential [3].

We then consider that a RK method is of order p if and only if

$$X(\tau) = \frac{1}{\sigma(\tau)} \left(\Phi(\tau) - \frac{1}{\gamma(\tau)} \right) = 0, \quad \text{for every } \tau \in T_i, \quad \text{for } i = 1(1)p.$$

The above relation defines a set of order conditons, which are linear in the components of b and nonlinear in the components of A, c (see, for example, Hairer *et al.* [8] or Butcher [3]). In the following the symbol $T^{(i)}$ denotes a vector whose elements are all the elements of the set $X(T_i)$ in some prescribed (but otherwise arbitrary) order.

The unique matching between a rooted tree τ and an order condition, comes clear after putting b at root, A at internal nodes c at leaves and using a prefix multiplication. Then we produce the single $\Phi(\tau)$ from the order conditions. For example using the following tree τ we name its nodes

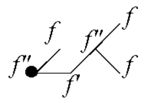


and we produce $\Phi(\tau) = bCA^2c^2$.

TABLE II The number of order conditions for systems and scalar autonomous equation

Order	1	2	3	4	5	6	7	8	9	10
System	1	1	2	4	9	20	48	115	286	719
Scalar aut.	1	1	2	3	5	7	11	15	22	30

In the same sense we derive the corresponding elementary differential. We put at every node (including root and the leaves) $f^{(k)}$ where k is the number of successors of the node. Using the same tree and putting the derivatives

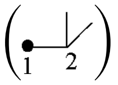
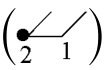
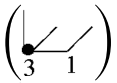
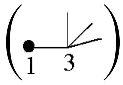


we conclude to the elementary differential $F(\tau) = f''(f,\ f'f''(f,\ f))$.

The two columns at right of Table I, show this relation for order conditions up to four.

The number of equations of condition (equals the number of rooted trees) for orders up to ten are given in the first row of Table II.

Now observe that $F(t_{42}) = f'f''(f,\ f) \neq f''(f'f,\ f) = F(t_{43})$ for systems of ODEs enforcing two separate equations t_{42} and t_{43}. But this is not necessary in the scalar case since $f' \cdot f''(f,\ f) = f'' \cdot f' \cdot f^2 = f''(f' \cdot f,\ f)$, and these equations may combine in $\tilde{t}_{42} = t_{42} + t_{43}$.

If for every tree τ we write down the degrees of derivatives at the corresponding node, we observe that we arrive at a connection of trees with the unrestricted partitions of a number [13]. So t_{42}  corresponds to $1 \cdot 2$ while t_{43}  corresponds to $2 \cdot 1$. These two trees resulted in $\tilde{t}_{42}$ indeed. Similarly $t_{52} = \frac{1}{2}bC^2Ac - \frac{1}{20}$  corresponding to $3 \cdot 1$ and $t_{53} = \frac{1}{6}bAc^3 - \frac{1}{120}$ , corresponding to $1 \cdot 3$ combine to $\tilde{t}_{52} = \frac{1}{2}bC^2Ac - \frac{1}{20} + \frac{1}{6}bAc^3 - \frac{1}{120}$. The rest order conditions follow this way and are listed in Table III.

The enumeration of order conditions is based in the relevant theory of unrestricted partitions of a number [13, pp 122]. The number of order conditions up to tenth order are listed in the last row of Table II.

3 CONSTRUCTION OF THE NEW PAIRS

The most popular RK pairs share orders five and four. $12 + 7 = 19$ equations of condition have to be solved then. If we choose a six stages method is possible to use the last stage in such a way that this evaluation can be taken over as the first evaluation for the next step [7]. This technique is called FSAL (First Stage As Last) pair, and thereby the number of evaluation per step will be reduced to five. The free parameters for solving the 19 equations are 20 ($c_2, c_3, c_4, c_5, a_{32}, a_{42}, a_{43},\ a_{52}, a_{53}, a_{54}, b_1, b_2, b_3, b_4, b_5, \hat{b}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4$ and $\hat{b}_5$). The other coefficients are restricted by the assumptions $Ae = c, c_6 = 1,\ a_{6j} = b_j,\ j = 1, 2, 3, 4, 5$ (FSAL), while we fix $\hat{b}_6 = \frac{1}{40} \neq b_6 = 0$.

We tried to solve the problem using a Levenberg-Marquardt method with line search. The resulted method NEW5(4)a accurate to 15–16 digits can by found in Table IV.

TABLE III The equations of condition for scalar autonomous RK methods

$$
\begin{aligned}
\tilde{t}_{11} &= b \cdot e - 1 \\
\tilde{t}_{21} &= b \cdot c - \tfrac{1}{2} \\
\tilde{t}_{31} &= \tfrac{1}{2} b \cdot c^2 - \tfrac{1}{6} \\
\tilde{t}_{32} &= b \cdot A \cdot c - \tfrac{1}{6} \\
\tilde{t}_{41} &= \tfrac{1}{6} b \cdot c^3 - \tfrac{1}{24} \\
\tilde{t}_{42} &= \tfrac{1}{2} b \cdot A \cdot c^2 - \tfrac{1}{24} + b \cdot C \cdot A \cdot c - \tfrac{1}{8} \\
\tilde{t}_{43} &= b \cdot A^2 \cdot c - \tfrac{1}{24} \\
\tilde{t}_{51} &= \tfrac{1}{24} b \cdot c^4 - \tfrac{1}{120} \\
\tilde{t}_{52} &= \tfrac{1}{2} bC^2Ac - \tfrac{1}{20} + \tfrac{1}{6} bAc^3 - \tfrac{1}{120} \\
\tilde{t}_{53} &= \tfrac{1}{2} b \cdot C \cdot A \cdot c^2 - \tfrac{1}{30} \\
\tilde{t}_{54} &= b \cdot C \cdot A^2 \cdot c - \tfrac{1}{30} + b \cdot A \cdot C \cdot A \cdot c - \tfrac{1}{40} + \tfrac{1}{2} b \cdot A^2 \cdot c^2 - \tfrac{1}{120} + \tfrac{1}{2} b \cdot ((Ac) \cdot (Ac)) - \tfrac{1}{40} \\
\tilde{t}_{55} &= b \cdot A^3 \cdot c - \tfrac{1}{120} \\
\tilde{t}_{61} &= \tfrac{1}{120} b \cdot c^5 - \tfrac{1}{720} \\
\tilde{t}_{62} &= \tfrac{1}{6} \cdot b \cdot C^3 \cdot A \cdot c^2 - \tfrac{1}{72} + \tfrac{1}{24} \cdot b \cdot a \cdot c^4 - \tfrac{1}{720} \\
\tilde{t}_{63} &= \tfrac{1}{4} \cdot b \cdot C^2 \cdot A \cdot c^2 - \tfrac{1}{72} + \tfrac{1}{6} \cdot b \cdot C \cdot A \cdot c^3 - \tfrac{1}{144} \\
\tilde{t}_{64} &= \tfrac{1}{2} bC(Ac)^2 - \tfrac{1}{48} + \tfrac{1}{2} bC^2A^2c - \tfrac{1}{72} + \tfrac{1}{6} bA^2c^3 - \tfrac{1}{720} + \tfrac{1}{2} bAC^2Ac - \tfrac{1}{120} \\
\tilde{t}_{65} &= bCACAc - \tfrac{1}{48} + \tfrac{1}{2} b \cdot ((Ac^2) \cdot (Ac)) - \tfrac{1}{72} + \tfrac{1}{2} bCA^2c^2 - \tfrac{1}{144} + \tfrac{1}{2} bACAc^2 - \tfrac{1}{180} \\
\tilde{t}_{66} &= \tfrac{1}{2} bA^3c^2 - \tfrac{1}{720} + bA^2CAc - \tfrac{1}{240} + bACA^2c - \tfrac{1}{180} + bCA^3c - \tfrac{1}{144} + \tfrac{1}{2} bA(Ac)^2 - \tfrac{1}{240} \\
&\quad + b((Ac)(A^2c)) - \tfrac{1}{72} \\
\tilde{t}_{67} &= b \cdot A^4 \cdot c - \tfrac{1}{720}
\end{aligned}
$$

It is worth mentioning that conventional RK pairs of orders 5(4) share 6 stages. Even a single fifth order RK method needs six stages to be derived.

A common practice in RK literature for construction of pairs, is the simultaneous minimization of the norm of the coefficients of principal truncation error $\|T^{(p+1)}\|_2$ that promises some better performance among pairs of order p. In the previous case there is only one parameter more than equations we have to solve so it seems difficult to derive a good $\|T^{(p+1)}\|_2$. According to Shampine [14] a measure of efficiency of RK methods is

$$\text{eff} = \text{stages } \cdot \|T^{(p+1)}\|_2^{1/(p+1)}. \tag{7}$$

TABLE IV The coefficients of the new 5 stages NEW5(4) a pair

$a_{21} = 7.983935319765683D - 1$	$a_{31} = 1.202381595746123D - 1$
$a_{32} = 1.128649860170427D - 1$	$a_{41} = 2.369003675496253D - 1$
$a_{42} = 4.087329938001282D - 2$	$a_{43} = 4.053316066041420D - 1$
$a_{51} = 3.942557940083695D - 1$	$a_{52} = -6.463834165307711D - 1$
$a_{53} = -4.156640553306520D - 1$	$a_{54} = 1.6338978367814070$
$c_2 = 7.983935319765683D - 1$	$c_3 = 2.331031455916550D - 1$
$c_4 = 6.831052735337801D - 1$	$c_5 = 9.661061589283534D - 1$
$c_6 = 1$	$b_1 = 6.417799939883591D - 2$
$b_2 = -7.247079043141412D - 2$	$b_3 = 3.787268997297880D - 1$
$b_4 = 4.899267581974183D - 1$	$b_5 = 1.396391331053720D - 1$
$b_6 = 0$	$\hat{b}_1 = 6.619132135710427D - 2$
$\hat{b}_2 = -8.196722114333793D - 2$	$\hat{b}_3 = 3.733280325768971D - 1$
$\hat{b}_4 = 5.056592903053327D - 1$	$\hat{b}_5 = 1.117885769040039D - 1$
$\hat{b}_6 = 2.5D - 2$	$a_{6j} = b_j,\ j = 1, 2, 3, 4, 5$

TABLE V The coefficients of the new 6 stages NEW5(4)b pair

$a_{21} = 1.847335922100224D - 1$	$a_{31} = -1.645069880057942D - 1$
$a_{32} = 9.194285585489098D - 1$	$a_{41} = 1.587175467729586D - 1$
$a_{42} = 1.962853625884246D - 1$	$a_{43} = 5.676905153972640D - 1$
$a_{51} = 3.743387591600372D - 1$	$a_{52} = -2.643740896944088D - 1$
$a_{53} = 1.771817353585664D - 1$	$a_{54} = -9.321767771586958D - 2$
$a_{61} = 2.502170241513760D - 1$	$a_{62} = 2.907083508501376D - 1$
$a_{63} = 5.339090024278675D - 2$	$a_{64} = 1.033290279145638D - 1$
$a_{65} = -3.529823979691512D - 1$	$a_{7j} = b_j,\ j = 1, 2, 3, 4, 5, 6,\ c_7 = 1$
$c_2 = 1.847335922100224D - 1$	$c_3 = 7.549215705431156D - 1$
$c_4 = 9.226934247586472D - 1$	$c_5 = 1.939287271083252D - 1$
$c_6 = 3.446629051897131D - 1$	$b_1 = 9.339100081427762D - 2$
$b_2 = 2.612750383228366D - 1$	$b_3 = 2.804013355360740D - 1$
$b_4 = 1.417598731551612D - 1$	$b_5 = -2.144996937291904D - 1$
$b_6 = 4.376724459008408D - 1$	$b_7 = 0,\ \hat{b}_7 = 2.5D - 2$
$\hat{b}_1 = 1.105115855660626D - 1$	$\hat{b}_2 = 3.895429703363805D - 2$
$\hat{b}_3 = 2.857445777154804D - 1$	$\hat{b}_4 = 1.055499389515022D - 1$
$\hat{b}_5 = -3.338705571936740D - 2$	$\hat{b}_6 = 4.676266564526840D - 1$

So it is possible to accept an increase in the number of stages if we ensure a considerable reduction in the norm of the error. We tried a seven stage FSAL method sharing 27 free parameters. The new pair NEW5(4)b, with minimal $\|T^{(6)}\|_2$ can be found in Table V.

This method was not derived using a constrained minimization procedure. Here we also preferred the Levenberg-Marquardt method with line search, after we multiplied the order conditions by a large number.

The main characteristics of our new methods and the most popular 5(4) pair [6], due to Dormand and Prince [4], can be found in Table VI.

Observing (7) we see that the efficiency measure for the new pair is smaller than the corresponding value of DP5(4), so some better performance for the former pair is expected.

The derivation of higher order pairs becomes very difficult. For a 6(4) pair we have to solve 29 equations of condition, so 7 stages are needed in order to get the required 34 coefficients. Seven stages are enough for a conventional 6(4) pair also [16]. Perhaps we may gain 1–2 stages for higher order pairs but the resulting nonlinear system to be solved can not be simplified. The common assumptions valid for standard pairs do not apply in this case. For example, in the general case setting, $Ac = \frac{1}{2}c^2$, we drop equations $t_{43} = b \cdot C \cdot A \cdot c - \frac{1}{8} = \frac{1}{2}b \cdot c^3 - \frac{1}{8}$ to equation $t_{41} = \frac{1}{6}b \cdot c^3 - \frac{1}{24}$. This does not work in scalar autonomous cases since t_{43} is already mixed with t_{42}.

TABLE VI The main characteristics of the pairs presented here

Method	**Stages*	$\|T^{(6)}\|_2$	*eff*	I_R	B_2	C_2	D_∞
NEW5(5)a	5	1.9×10^{-3}	1.75	-3.2	3.1	1.5	1.6
NEW5(4)b	6	8.2×10^{-6}	0.85	-3.5	0.7	0.7	0.9
DP5(4)	6	3.8×10^{-4}	1.61	-3.3	1.6	1.9	11.6

I_R: real stability interval
*: effective number of

$$B_2 = \frac{\|T^{(6)}\|_2}{\|T^{(5)}\|_2}, \quad C_2 = \frac{\|T^{(6)} - \hat{T}^{(6)}\|_2}{\|T^{(5)}\|_2}, \quad D_\infty = \max(\|A\|_\infty, \|b\|, \|\hat{b}\|_\infty, \|c\|_\infty).$$

4 NUMERICAL RESULTS

Our new seven stages FSAL pair is compared along with its main competitor DP5(4) on a set of relevant problems.

The problems chosen are:

4.1 Problem 1. Detest problem A1 [5]:

$$y' = -y, \quad y(0) = 1, \;\; x \in [0, 20]$$

with analytic solution $y(x) = \mathrm{e}^{-x}$.

4.2 Problem 2. Detest problem A2:

$$y' = -\frac{y^3}{2}, \quad y(0) = 1, \;\; x \in [0, 20]$$

with analytic solution $y(x) = 1\big/\big(\sqrt{1+x}\big)$.

4.3 Problem 3. Detest problem A3:

$$y' = -\frac{y}{4}\left(1 - \frac{y}{20}\right), \quad y(0) = 1, \;\; x \in [0, 20]$$

with analytic solution $y(x) = 20\big/\big(1 + 19\,\mathrm{e}^{-x/4}\big)$.

4.4 Problem 4:

$$y' = -y^{3/2}, \quad y(0) = 4, \;\; x \in [0, 20]$$

with analytic solution $y(x) = 4/(1+x)^2$.

Both pairs were run at tolerances $10^{-3}, 10^{-4}, \dots, 10^{-11}$, for all problems. A comparison based on the works of Enright and Pryce [5] (see also Sharp [15]), was carried out. Briefly, let us assume that the global error satisfies the relation $ge = C \cdot \mathrm{TOL}^E$ and that its value is known for several tolerances. The values of E and C can easily be found in the sense of a least squares approximation. These values are then used, with linear interpolation, in order to estimate the number of derivative evaluations required to achieve a prescribed accuracy. We present the efficiency gains of some of the older methods in relation to the new ones, for the respective problems and the expected accuracies, counted in units of 1%, in tables. The numbers in these tables are the ratios in function evaluations cost of the two pairs being tested. The larger value is always being divided by the smaller value and the efficiency gain is formed by subtracting 1 from this ratio. The gain is multiplied by -1 whenever the first pair is more efficient than the second. Subsequently the result is multiplied by 100 and rounded to the nearest integer.

TABLE VII DP5(4)P against NEW5(4)

Log expected accuracy	*1*	*2*	*3*	*4*	
−2					
−3					
−4	14	6	32		
−5	18	11	14	22	
−6	21	14	28	24	
−7	23	13	42	29	
−8	25	16	58	30	
−9	29	17	76	31	
−10	33	15	88	32	
−11	35	14		32	
	25	13	48	28	29%

The final row is the average efficiency for a problem for all expected accuracies, while the most right number is the average efficiency for all problems.

This type of test was verified by us in a number of recent articles [10–12, 16]. The cumulative results can be found in Table VII.

By interpreting the results it is obvious that the new method is especially suited for scalar autonomous problems and clearly outperforms DP5(4). The average gain is by no means remarkable for pairs of the same order.

References

[1] Butcher, J.C. (1964) Implicit Runge-Kutta processes, *Math. Comput.* **18**, 50–64.

[2] Butcher, J.C. (1964) On Runge-Kutta processes of high order, *J. Austral. Math. Soc.* **4**, 179–194.

[3] Butcher, J.C. (1987) *The Numerical Analysis of ODEs: Runge-Kutta and General Linear Methods* (Wiley, Chichester).

[4] Dormand, J.R. and Prince, P.J. (1980) A family of embedded Runge-Kutta formulae, *J. Comput. Appl. Math.* **6**, 19–26.

[5] Enright, W. and Pryce, J.D. (1987) Two FORTRAN packages for assessing initial value methods, *ACM Trans. Math. Software,* **13**, 1–27.

[6] Matlab (1999) *Using Matlab, Version 5*, The MathWorks inc., Natick.

[7] Fehlberg, E. (1969) Low order classical Runge-Kutta formulas with step-size control and their application to some heat transfer problems, NASA Tech. Rep. TR R-315, G. C. Marsal Space Flight Center, Ala. 35812.

[8] Hairer, E., Norsett, S.P. and Wanner, G. (1987) *Solving Ordinary Differential Equations I* (Springer-Verlag, Berlin).

[9] Lambert, J.D. (1991) *Numerical Methods for ODEs* (Wiley, Chichester).

[10] Papakostas, S.N. and Tsitouras, Ch. (1997) Highly continuous interpolants for one step ODE solvers and their application to Runge-Kutta methods, SIAM J. Numer. Anal. **34** , 22–47.

[11] Papakostas, S.N. and Tsitouras, Ch. (1999) High algebraic order, high phase-lag order Runge-Kutta and Nystrm pairs, *SIAM J. Sci. Comput.* **21**, 747–763.

[12] Papakostas, S.N., Tsitouras, Ch. and Papageorgiou, G. (1996) A general family of Runge-Kutta pairs of orders 6(5), *SIAM J. Numer. Anal.* **33**, 917–926.

[13] Riordan, J. (1958) *An Introduction to Combinatorial Analysis* (Wiley, New York).

[14] Shampine, L.F. (1986) Some practical Runge-Kutta formulas, *Math. Comput.* **46**, 135–150.

[15] Sharp, P. (1991) Numerical comparisons of some explicit Runge-Kutta pairs, *ACM Trans. Math. Soft.* **17**, 387–409.

[16] Tsitouras, Ch. and Papakostas, S.N. (1999) Cheap error estimation for Runge-Kutta methods, *SIAM J. Sci. Comput.* **20**, 2067–2088.