Optimized explicit Runge-Kutta pair of orders 9(8).

CH. TSITOURAS

ABSTRACT. A fully explicit algorithm for deriving a Runge-Kutta pair of orders 9(8) is presented in this paper. After that an optimal pair is given, which is found to outperform all other existing Runge-Kutta pairs when it is applied in quadruple presision.

Keywords : Initial Value Problems, High order, Embedded pairs, steepest descent.

AMS classification : 65L05, 65L06, 65K10.

1. INTRODUCTION

Explicit Runge-Kutta (RK) pairs are widely used for the numerical solution of the initial value problem

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

where $f : \mathbb{R} \times \mathbb{R}^m \mapsto \mathbb{R}^m$. 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, $\hat{b}^T$, $c \in \mathbb{R}^s$ and $A \in \mathbb{R}^{s \times s}$ 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

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

and

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

with

$$f_{ni} = f(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{ij} f_{nj}) \text{ for } i = 1, 2, .., s.$$

From this embedded form 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 [7],

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

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.

2. Pairs of orders 9(8)

A RK method applied to an autonomous system of differential equations of the type (1) is said to be of algebraic order p if and only if

$$X(\tau) = 0, \ \forall \tau \in T_i, \ \text{for } i = 1, 2, \ldots, p, \tag{2}$$

where T_i is the set of ith order (rooted) trees and

$$X(\tau) = \frac{1}{\sigma(\tau)} \left(\Phi(\tau) - \frac{1}{\gamma(\tau)} \right).$$

σ,γ are integral functions of τ (symmetry and density function, respectively, in the terminology introduced by Butcher, see [3]) and Φ is a certain composition of A, b, c. In what follows the symbol $T^{(i)}$ denotes a vector with elements all the elements of the set $X(T_i)$ in some arbitrary order.

In the case of a 9(8) pair, equation (2) is expanded in 486 nonlinear algebraic equations that must be satisfied by its higher order method and 200 equations by its lower order method. All methods of order higher than four that have been constructed so far , as well as those that we consider in this article obey the simplifying assumption $Ae = c$, $e = (1, 1, \ldots, 1)^T \in \mathbb{R}^s$.

All currently known pairs of orders 9(8) (except that derived by Fehlberg [4]) use 16 stages. In such a case the number of available free coefficients $a, b, \widehat{b}, c$, is only 187. Since the number of unknowns is less than the number of equations and mainly because some of the latter equations are strongly nonlinear with respect to the elements of A, it is necessary to apply some sort of simplifying assumptions for their solution. Then we derive a smaller set of equations to be solved for a sufficient number of unknowns. In addition a minimisation of $\left\| T^{(10)} \right\|_2$ is needed for achieving more efficient pairs. So the greater the number of free parameters (among the 187 ones) the better the changes for smaller $\left\| T^{(10)} \right\|_2$.

Table 1: The main characteristics of the pairs appeared in this paper.

Pair	s	p	$\left\|T^{(p+1)}\right\|_2$	B_2	C_2	S_R	D_∞
PD87 [12]	13	8	$4.51 \cdot 10^{-6}$	2.24	2.27	-5.16	16.6
P87[8]	13	8	$7.35 \cdot 10^{-7}$	2.03	2.03	-5.90	11.7
Fe89 [4]	17	9	$1.58 \cdot 10^{-6}$	30.01	30.05	-3.00	27.5
V89 [15]	16	9	$6.11 \cdot 10^{-5}$	1.98	2.31	-4.19	627
V89a [16]	16	9	$1.37 \cdot 10^{-5}$	2.54	2.52	-3.91	425
V89b [16]	16	9	$8.19 \cdot 10^{-5}$	2.12	2.12	-3.65	66.5
New98	16	9	$3.64 \cdot 10^{-7}$	3.22	3.22	-3.94	26.2
Ha10(6) [14]	18	10	$5.27 \cdot 10^{-6}$	–	–	-2.70	1.05

s :stages, p :order,
S_R: Real Stability Interval,
$D_\infty = \max\left(\max_{i,j} |a_{ij}|, \|b\|_\infty, \|c\|_\infty\right)$,
$B_2 = \left\|\widehat{T}^{(p+1)}\right\|_2 / \left\|\widehat{T}^{(p)}\right\|_2$, $C_2 = \left\|T^{(p+1)} - \widehat{T}^{(p+1)}\right\|_2 / \left\|\widehat{T}^{(p)}\right\|_2$ [12], not applied for Ha10(6).

The first who constructed a 9(8) pair was E. Fehlberg [4], at a cost of 17 function evaluations per step. That pair had the disadvantage of being quadrature defective. Some of the 9th order truncation coefficients for the lower order formula were equal to zero, because Fehlberg used the free parameters c_5, c_8, c_{11}, c_{13} for minimization of $\left\|\widehat{T}^{(9)}\right\|_2$ since the methods were advanced by the value $\hat{y}_{n+1}$ then. In consequence there are initial value problems where the estimated error might be equal to zero. This method was forgotten through the years passed but the small $\left\|T^{(10)}\right\|_2$ achieved by its 9th order formula is promising. In addition we may drop the 15th stage which is useless for the higher order formula and embed a 6th and a 3rd order formulas at no cost. Then we may implement a 16 stage 9(6)3 triple using some modification of the step size algorithm used for the 8(5)3 triple given by Hairer, Norsett and Wanner [6]. In alternative a 9(6) or possibly a 9(7) pair may by constructed using step size algorithm introduced by Tsitouras and Papakostas [14]. Of course a fair comparison demands the construction of 15, or even possibly 14 stages 9(.) pairs or triples simultaneously.

Later, J. H. Verner [15], indeed dropped the useless stage and manage to embed an eighth order formula. He used $c_2, c_5, c_9, c_{10}, c_{11}, c_{13}, c_{14}, a_{11,6}$ as free parameters ignoring the significance of b_{16}, or even $\widehat{b}_{15}, \widehat{b}_{16}$. He derived the pair V89 setting $a_{16,15} = a_{11,6} = b_{15} = \widehat{b}_{15} = \widehat{b}_{16} = 0$. Only the condition $a_{16,15} = 0$ is obligatory for this family.

The resulting pair gave not very satisfactory results in comparison to the pair PD87 of Prince and Dormand [12], or other 8(7) pairs.

Recently J. H. Verner [16], derived very interesting families of 9(8) and 8(7) pairs with $a_{s,s-1} \neq 0$, gaining a free parameter. This was done reducing the stage order of internal stages, something that has been tested successfully on 6(5) pairs in the past, [9, 10]. He presented there two pairs, V98a and V98b whose major characteristics are given in Table 1. Unfortunately the algorithm for such a family involves an implicit evaluation of an intermediate coefficient. According to our experience it is much more fruitful handling an explicit algorithm [13], than handling an implicit one [10].

The main disadvantage of all 9(8) pairs presented until now, is its rather big value of $\left\|T^{(10)}\right\|_2$. Verner admits that there is only a marginal improvement over the other known pairs, [16]. In this paper we extended the family given in [15], including the parameters $b_{16}, \widehat{b}_{15}$ and $\widehat{b}_{16}$. We found that these extra parameters helped considerably the production of a pair with minimized $\left\|T^{(10)}\right\|_2$.

2.1. The explicit algorithm for deriving an 16-stage pair of orders 9(8)..

In the following algorithm, whenever c is a vector, we denote by c^i the componetwise multiplication $\underbrace{c \cdot c \cdot \cdots c}_{i}$ (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). We also define $C = diag\,(c)$, and I the identity matrix of a proper dimension.

Select $c_2, c_5, c_9, c_{10}, c_{11}, c_{13}, c_{14}, a_{116}, b_{16}, \widehat{b}_{15}$ and $\widehat{b}_{16}$ as free parameters.

Set $a_{1615} = 0$, $b_i = \widehat{b}_i = 0$ for $i = 2, 3, \ldots, 7$, $a_{i2} = 0$, for $i = 4, 5, \ldots, 16$, $a_{i3} = 0$, for $i = 6, 5, \ldots, 16$, and $a_{i4} = a_{i5} = 0$, for $i = 8, 5, \ldots, 16$.

Choose $c_7 = c_9\left(\frac{4}{5} - 2\frac{\sqrt{6}}{15}\right)$ or $c_7 = c_9\left(\frac{4}{5} + 2\frac{\sqrt{6}}{15}\right)$.

$c_8 = \frac{4}{3}c_9$, $c_6 = \frac{c_8(4c_7 - 3c_8)}{2(3c_7 - 2c_8)}$, $c_4 = \frac{c_6(4c_5 - 3c_6)}{2(3c_5 - 2c_6)}$, $c_3 = \frac{2}{3}c_4$,

$a_{32} = c_3^2/(2c_2)$, $a_{43} = c_4^2/(2c_3)$,

$a_{53} = 3c_5^2(3c_4 - 2c_5)/(4c_4^2)$, $a_{54} = -c_5^2(c_4 - c_5)/c_4^2$.

Solve $(Ac)_6 = c_6^2/2$, $(Ac^2)_6 = c_6^3/3$ for a_{64} and a_{65}.

Solve $(Ac)_7 = c_7^2/2$, $(Ac^2)_7 = c_7^3/3$, $(Ac^3)_7 = c_7^4/4$ for a_{74}, a_{75} and a_{76}.

Solve $(Ac)_8 = c_8^2/2$, $(Ac^2)_8 = c_8^3/3$, for a_{86} and a_{87}.

Solve $(Ac)_9 = c_9^2/2$, $(Ac^2)_9 = c_9^3/3$, $(Ac^3)_9 = c_9^4/4$ for a_{96}, a_{97} and a_{98}.

Evaluate c_{12} from equation[1],

$$\int_0^1 p(x)\frac{(1-x)^3}{3!}dx \cdot \int_0^1 \overline{p}(x)\,(1-x)\,dx - \int_0^1 p(x)\frac{(1-x)^2}{2!}dx \cdot \int_0^1 \overline{p}(x)\frac{(1-x)^2}{2!}dx = 0$$

[1] $p(x) = x\,(x - c_8)\,(x - c_9)\,(x - c_{10})\,(x - c_{11})$, $\overline{p}(x) = (x - c_{12})\,p(x)$, see [15].

Derive b_i, $i=8,9,\ldots,15$, from equations $bc^{i-7}=1/(i-6)$.
Derive $\widehat{b}_i$, $i=8,9,\ldots,14$, from equations $\widehat{b}c^{i-7}=1/(i-6)$.
Solve $(b(A+C-I))_{14}=\left(\widehat{b}(A+C-I)\right)_{14}=0$, for $a_{15,14}$ and $a_{16,14}$.
Solve $(Ac^i)_{10}=c_{10}^{i+1}/(i+1)$, $i=1,2,3,4$ for $a_{10,6}, a_{10,7}, a_{10,8}$ and $a_{10,9}$.
Solve $(Ac^i)_{11}=c_{11}^{i+1}/(i+1)$, $i=1,2,3,4$ for $a_{11,7}, a_{11,8}, a_{11,9}$ and $a_{11,10}$.
Substitute $a_{12,6}, a_{13,6}$ and $a_{14,6}$ from the equations $(b(C-I)A)_6=0$, $\left(\widehat{b}(C-I)A\right)_6=0$ and $(b(C-I)(C-c_{14}I)A)_6=0$.
Solve $(b(A+C-I))_6=\left(\widehat{b}(A+C-I)\right)_6=0$, for $a_{15,6}$ and $a_{16,6}$.
Substitute $a_{12,7}, a_{13,7}$ and $a_{14,7}$ from the equations $(b(C-I)A)_7=0$, $\left(\widehat{b}(C-I)A\right)_7=0$ and $(b(C-I)(C-c_{14}I)A)_7=0$.
Solve $(b(A+C-I))_7=\left(\widehat{b}(A+C-I)\right)_7=0$, for $a_{15,7}$ and $a_{16,7}$
Solve $(Ac^i)_{12}=c_{12}^{i+1}/(i+1)$, $i=1,2,3,4$ for $a_{12,8}, a_{12,9}, a_{12,10}$ and $a_{12,11}$.
Derive $a_{14,13}$ from equation

$$\begin{aligned} &b(C-I)A(C-c_{10}I)(C-c_9I)(C-c_8I)(C-c_{12}I)(C-c_{11}I)c= \\ &=\textstyle\int_0^1(x-1)\int_0^x(x-c_{10})(x-c_9)(x-c_8)(x-c_{20})(x-c_{11})\,xdxdx. \end{aligned}$$

Solve $(b(A+C-I))_{13}=\left(\widehat{b}(A+C-I)\right)_{13}=0$, for $a_{15,13}$ and $a_{16,13}$.
Derive $a_{13,12}$ and $a_{14,12}$ from equations

$$\begin{aligned} &\widehat{b}(C-I)A(C-c_9I)(C-c_8I)(C-c_{11}I)(C-c_{10}I)c= \\ &=\textstyle\int_0^1(x-1)\int_0^x(x-c_9)(x-c_8)(x-c_{11})(x-c_{10})\,xdxdx, \end{aligned}$$

and

$$\begin{aligned} &b(C-I)(C-c_{12}I)A(C-c_{10}I)(C-c_9I)(C-c_8I)(C-c_{11}I)c= \\ &=\textstyle\int_0^1(x-1)(x-c_{12})\int_0^x(x-c_{10})(x-c_9)(x-c_8)(x-c_{11})\,xdxdx. \end{aligned}$$

Solve $(Ac^i)_{13}=c_{13}^{i+1}/(i+1)$, $i=1,2,3,4$ for $a_{13,8}, a_{13,9}, a_{13,10}$ and $a_{13,11}$.
Solve $(Ac^i)_{14}=c_{14}^{i+1}/(i+1)$, $i=1,2,3,4$ for $a_{14,8}, a_{14,9}, a_{14,10}$ and $a_{14,11}$.
Solve $(Ac^i)_j=\frac{1}{i+1}c_j^{i+1}$, $j=15,16$ $i=1,2,3,4$, and $bCAc^5=\frac{1}{48}$, $\widehat{b}CAc^5=\frac{1}{48}$ for a_{ij}, $i=15,16$, $j=8,9,10,11,12$.

Finally we get $b_1=1-\sum_{i=2}^s b_i$, $\widehat{b}_1=1-\sum_{i=2}^s\widehat{b}_i$ and $a_{i1}=c_i-\sum_{i=2}^{i-1}a_{ij}$, for $i=2,3,\ldots,16$.

3. Derivation of 9(8) pairs

We first implement the above explicit algorithm in the matlab function
function f = rk98(x)

Table 2: The 24 truncation error coefficients that produce $\left\|T^{(10)}\right\|_2$.

$e_1 = 35990.79(bc^9 - 1/10)/362880$	$e_2 = 916.92(bCAc^7 - 1/80)/5040$
$e_3 = 343.94(bC^2Ac^6 - 1/70)/1440$	$e_4 = 198.57(bCA^2c^6 - 1/560)/720$
$e_5 = 175.44(bC^3Ac^5 - 1/60)/720$	$e_6 = 78.46(bC^2A^2c^5 - 1/420)/240$
$e_7 = 45.30(bCA^3c^5 - 1/3360)/120$	$e_8 = 48.37(bAC^2A^2c^4 - 1/900)/48$
$e_9 = 45.30(bCACAc^5 - 1/480)/120$	$e_{10} = 21.63(bC^2A^3c^4 - 1/2100)/48$
$e_{11} = 12.49(bCA^4c^4 - 1/16800)/24$	$e_{12} = 1.41(bCA^7c - 1/403200)$
$e_{13} = 12.49(bCACA^2c^4 - 1/2400)/24$	$e_{14} = 2.45(bC^2A^6c - 1/50400)/2$
$e_{15} = 5.48(bAC^2A^5c - 1/21600)/2$	$e_{16} = 1.41(bCACA^5c - 1/57600)$
$e_{17} = 17.32(bC^3A^3c^3 - 1/1200)/36$	$e_{18} = 7.75(bC^2A^4c^3 - 1/8400)/12$
$e_{19} = 4.47(bCA^5c^3 - 1/67200)/6$	$e_{20} = 5.48(bC^3A^4c^2 - 1/3600)/12$
$e_{21} = 2.45(bC^2A^5c^2 - 1/25200)/4$	$e_{22} = 1.41(bCA^6c^2 - 1/201600)/2$
$e_{23} = 4.47(bCACA^3c^3 - 1/9600)/6$	$e_{24} = 1.41(bCACA^4c^2 - 1/28800)/2$

$$f = \left\|T^{(10)}\right\|_2 = \sqrt{\sum_{i=1}^{24} e_i^2}$$

with input the eleven free parameters and output the euclidean norm of the tenth order truncation error of the method. There is no need to evaluate all 719 truncation error coefficients of 10th order. Only 24 of them are enough for consisting the desired value $f = \left\|T^{(10)}\right\|_2 = \sqrt{\sum_{i=1}^{24} e_i^2}$, where e_i are given in Table 2. Writing rk98 correctly we have to compute powers of A or c only once, so we may get the result in a very short time.

Then we have to use an optimization method in order to minimize f. We prefer the modified steepest descent algorithm described briefly in Vrahatis, Androulakis and Manoussakis, [18]. Programming it in matlab is an easy task so we present the program in Figure 1.

We ran that program taking 1000 random choices for the starting vector. Refining the first outputs we used them again as inputs in another try, and we concluded to the method NEW9(8) given in the appendix. Actually matlab gave coefficients with only $14-15$ correct digits, but even $4-5$ digits would be enough for detecting a minimum. Observe that most of the free parameters in the appendix (like $c_{11} = 7/9, c_{14} = 39/40$ etc.) are simple fractions. Rounding the output parameters to the nearest fraction by a factor of 10^{-4}, we are then able to produce a 35-digits version of the method using a symbolic manipulation package. The value $\left\|T^{(10)}\right\|_2$ is very satisfactory while

Figure 1: The steepest descent matlab program.

```
function xnew=steepest(fun,x);

len=length(x);l0=512;k=-1;m=1;
dd=sqrt(eps);dx=dd*eye(len);e=1e-8;
mit=10;f=feval(fun,x);ff=f'*f;gf=zeros(len,1);
for i=1:len, % gradient evaluation
   gf(i)=(feval(fun,x+dx(:,i))-f)/dd;
end;

while k<mit & sqrt(gf'*gf)>e,
      k=k+1;l=l0;m=1;
      xtemp=x-l*gf;ftemp=feval(fun,xtemp);
      while ftemp-f>-0.5*l*ff & m<30,
            m=m+1;l=l0/(2^(m-1));
            xtemp=x-l*gf;ftemp=feval(fun,xtemp);
      end;
      x=xtemp;f=feval(fun,x);ff=f'*f;
      for i=1:len, % gradient evaluation
         gf(i)=(feval(fun,x+dx(:,i))-f)/dd;
      end;
end;
xnew=x;
```

other characteristics (like the small D_∞) are acceptable. Notice that since $b_{16} = \widehat{b}_{16}$, we save the first and the last function evaluation after a step rejection, so only 14 stages are wasted then.

4. Numerical results

We implemented the numerical results the way we did in a series of papers lately, [10, 11, 13, 14]. We choose for testing the most efficient 8(7) pairs, PD87 and P87 [8], and the most promising 9(8) pairs, our NEW9(8) and V89a which coefficients were kindly given by Prof. J. H. Verner [17]. These methods were run in quadruple precision for the 25 problems of the set DETEST [7], and for tolerances $10^{-12}, 10^{-14}, 10^{-16}, \cdots, 10^{-24}$. After that we compare every pair with NEW9(8), notifying the percentage difference in the number of function evaluations required for

Table 3: Efficiency gains of NEW9(8) relative to PD8(7), for the range of tolerances 10^{-12}, 10^{-14}, …, 10^{-24}.

log global error	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5	D1	D2	D3	D4	D5	E1	E2	E3	E4	E5
−10																				1					
−12						1											1	2	2	2					
−14	−3		0	1	3	3	−3	−2	6	1	−4	−5	−5	−5	−2	2	2	3	3	3	−2	0	3	1	3
−16	−3	6	2	2	3	5	−3	−1	8	2	−3	−3	−4	−4	−1	3	3	4	3	3	−1	2	5	2	4
−18	−2	7	2	4	4	7	−3	1	9	3	−2	−3	−3	−3	0	4	4	5	4	4	0	3	6	3	5
−20	−2	8	4	5	5	9	−2	2	11	4	−2	−2	−2	−2	1	6	5	6	5	4	0	4	8	4	6
−22	−1	9	5	7	6	11	−2	3	13	5	−1	−2	−2	−2	3	7	7	7	6		1	6	10	5	7
−24	−1	10	6	9	7		−1	5		7	0	−1	−1	−1							2	8		7	9
−26	0						0						0	0											
27%	−2	8	3	5	5	6	−2	1	9	4	−2	−3	−2	−2	1	4	4	4	4	3	0	4	6	4	6

Table 4: Efficiency gains of NEW9(8) relative to P8(7), for the range of tolerances 10^{-12}, 10^{-14}, …, 10^{-24}.

log global error	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5	D1	D2	D3	D4	D5	E1	E2	E3	E4	E5
−10																				−1					
−12						0											0	−1	0	0					
−14	0		2	4	1	1	0	1	4	2	−1	−2	−1	−1	2	2	0	0	1	0	1	0	1	2	0
−16	1	1	2	6	1	3	1	1	6	3	0	−1	0	0	3	3	1	1	1	1	2	1	2	3	0
−18	2	2	2	9	2	5	1	2	7	4	1	0	1	1	4	4	2	2	2	2	3	3	3	4	0
−20	2	2	2	12	2	7	2	3	9	5	2	1	2	2	5	5	3	2	2	3	4	4	4	6	1
−22	3	3	3	15	2	10	3	4	11	6	3	2	3	3	6	6	4	3	3		5	6	6	7	2
−24	4	3	3		3		4	5			5	3	3	3	7							8			3
−26																									
29%	2	2	2	9	2	4	2	3	7	4	2	1	1	1	4	4	2	1	1	1	3	4	3	5	1

achieving a given maximum global error over the range of integration. This percentage is called efficiency gain, and it is recorded for each problem and accuracy in Tables 3,4 and 5 respectively, in units of 10%. In these tables positive numbers mean that the first of the two methods is superior. The final row gives the mean value of efficiency gain for each problem. The final row's first number is the average efficiency gain for all problems. The empty places are due to the unavailability of data for the respective errors, see [10] for more details.

Interpreting the results we observe that NEW9(8) clearly outperforms the 8(7) pairs. PD8(7) pair seem to perform slightly better than P8(7), because of its excellent results on the 6 constant coefficients linear DETEST problems A1, B2, C1 − 4. The values $bA^{k-2}c - 1/k!$, $k = 9, 10$ are very small for PD8(7) so it behaves like a higher order method for this type of problems. The long stability intervals play no role when

Table 5: Efficiency gains of NEW9(8) relative to V9(8)a, for the range of tolerances 10^{-12}, 10^{-14}, …, 10^{-24}.

log global error	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5	D1	D2	D3	D4	D5	E1	E2	E3	E4	E5
−10																				3					
−12						0											5	5	5	4					
−14	2		2	3	5	1	1	3	1	3	2	1	2	2	4	8	5	6	5	4	3	2	1	3	2
−16	2	3	2	3	5	2	2	3	1	3	2	1	2	2	4	8	5	6	5	4	3	3	2	4	2
−18	2	3	3	4	4	3	2	4	2	3	3	2	3	3	5	8	5	6	5	4	3	4	2	4	2
−20	3	3	3	5	4	4	3	4	2	4	3	2	3	3	5	9	5	6	5		3	5	2	4	2
−22	3	3	4	6	4	5	3	4	2	4	3	3	3	3	5	9	5				3	6	3	4	3
−24	3	3		7			3	5	3	5	4	3	3	3	5						3	8	3	5	3
−26		3																							
36%	2	3	3	5	4	2	2	4	2	4	3	2	3	3	5	8	5	6	5	4	3	5	2	4	2

the methods are applied at so stringent tolerances and the stepsizes selected are very small. On the other problems P8(7) seems to be much better than PD8(7).

The V89a pair was chosen because of its small truncation error, but it hardly follows the results of the 8(7) pairs. It is in total $5-6\%$ less efficient even than Fe89. The defect in the error estimator of the latter pair do not affect the results of any of the 25 test problems. Dropping the 15th stage of Fe89 we may conclude to a pair (or triple) almost 10% more efficient than V89a. The results of V89b are not satisfactory either. In [14], the method Ha10(6) was presented and it was based on a tenth order method of Hairer [5]. That pair was the most efficient one, among all the pairs tested in [14], on high tolerances. According to our tests even Ha10(6) is 16% less efficient than New9(8) when run in the tolerances we used.

Finally we conclude that the NEW9(8) pair seem to be the better one for use in quadruple precision at high tolerances. Some better performance is possible if someone derive optimal 9(.) pairs or triples at a cost of $14-15$ stages.

5. Appendix

The coefficients of the new pair.

$c_2 = 0.020408163265306122448979591836734$ $c_3 = 0.088132939149981030086379159392415$
$c_4 = 0.13219940872497154512956873908862$ $c_5 = 0.42857142857142857142857142857142$
$c_6 = 0.53647553922432876813951009998132$ $c_7 = 0.22542922268043313662239466192343$
$c_8 = 0.63492063492063492063492063492063$ $c_9 = 0.47619047619047619047619047619047$
$c_{10} = 1.0555555555555555555555555555555$ $c_{11} = 0.77777777777777777777777777777777$
$c_{12} = 0.14741696242609947604784015871016$ $c_{13} = 0.9375$
$c_{14} = 0.975$ $c_{15} = 1$
$c_{16} = 1$

$b_1 = 0.04153556008805959168895898921867 2$ | $b_8 = -0.42522808741698055893278455437035 5$

$b_9 = 0.49112696294176088418706222862956 3$

$b_{10} = 0.45417824175884742543736768126380 4$

$b_{11} = 1.00603264909442806518378187219689 6$

$b_{12} = 0.23969807142877259184285842967673 0$

$b_{13} = -4.45549129773140746622918860353288 1$

$b_{14} = 9.28897877570610182445225059259350 5$

$b_{15} = -6.41006164510035158839953740490670 2$

$b_{16} = 10/13$

$\hat{b}_1 = 0.03999250134104311539086473895855 2$

$\hat{b}_8 = 0.04011325788584545964379727040448 6$

$\hat{b}_9 = 0.39251270956252331424179468475966 0$

$\hat{b}_{10} = -1.05119709476448575216749921113279 2$

$\hat{b}_{11} = -0.25050832988640247910532087799040 9$

$\hat{b}_{12} = 0.24654282889396906337851737662775 2$

$\hat{b}_{13} = 6.83996511237872594872767778721551 1$

$\hat{b}_{14} = -16.02665175464198790087906253807353 0$

$\hat{b}_{15} = 10.0$

$\hat{b}_{16} = 10/13$

$a_{2,1} = 0.02040816326530612244897959183673 4$

$a_{3,1} = -0.10216872744876831477697236065353 0$

$a_{3,2} = 0.19030166659874934486335152004594 5$

$a_{4,1} = 0.03304985218124288628239218477215 5$

$a_{4,3} = 0.09914955654372865884717655431646 7$

$a_{5,1} = 0.94392638321712918835566706728119 7$

$a_{5,3} = -3.63011506309348203707711430007311 0$

$a_{5,4} = 3.11476010844778142015001866136334 1$

$a_{6,1} = 0.02056923328616261711782848556408 5$

$a_{6,4} = 0.26048241831974029875523407230877 8$

$a_{6,5} = 0.25542388761842585226644754210846 2$

$a_{7,1} = 0.04318483705109263028835194255758 6$

$a_{7,4} = 0.17692848398076896455208429450024 9$

$a_{7,5} = 0.00771541462187698379431579576727 1$

$a_{7,6} = -0.00239951297330544201235737090167 2$

$a_{8,1} = 0.07054673721340388007054673721340 3$

$a_{8,6} = 0.23898607155585223812703202690113 0$

$a_{8,7} = 0.32538782615137880243734187080610 0$

$a_{9,1} = 0.07068452380952380952380952380952 3$

$a_{9,6} = 0.11469816948658278225718464710149 6$

$a_{9,7} = 0.32428992575151245583805344813659 8$

$a_{9,8} = -0.03348214285714285714285714285714 2$

$a_{10,1} = 0.38280395688657407407407407407407 4$

$a_{10,6} = -26.28104570763124221540974003468081 9$

$a_{10,7} = -1.74842464344900469817050687889942 7$

$a_{10,8} = 8.7107466796875$

$a_{10,9} = 19.99147527006172839506172839506172 8$

$a_{11,1} = 0.05775788905284689042365660435243 2$

$a_{11,6} = 0.76$

$a_{11,7} = 0.40185532879155600502450439723337 3$

$a_{11,8} = 0.06626589626062798854099077293240 4$

$a_{11,9} = -0.51506969549767873886526582816840 5$

$a_{11,10} = 0.00696835917042563265389183142797 3$

$a_{12,1} = 0.06994103035906395715686489597582 1$

$a_{12,6} = -0.12061660605654704907947500980244 2$

$a_{12,7} = 0.13269630173144586830109110576218 3$

$a_{12,8} = 0.20315325380776279220591082994878 1$

$a_{12,9} = -0.07548640175023889467177231137902 5$

$a_{12,10} = 0.00378910368643901129564314061820 5$

$a_{12,11} = -0.06605971935182620916042249241335 7$

$a_{13,1} = 0.6878836016002817993752515099013 03$ $a_{13,6} = -3.6588085273664515171969468970569 76$

$a_{13,7} = 5.8455033785646698903141027760598 91$ $a_{13,8} = 5.4114127520046888588448636967093 00$

$a_{13,9} = -1.5820959882696735689928941097123 96$ $a_{13,10} = 0.0994524886570629966778884663541 95$

$a_{13,11} = -1.3295934871052344040987715493039 22$ $a_{13,12} = -4.5362542180853440549234938929513 94$

$a_{14,1} = 0.8075602959459265637003617465986 99$ $a_{14,6} = -7.8357501395579927721982493581512 31$

$a_{14,7} = 6.1742000259412297701231844231469 54$ $a_{14,8} = 7.1113110527837991487238222838460 58$

$a_{14,9} = 1.1264139826521842683585607765065 18$ $a_{14,10} = 0.1138838501589788637237573235699 77$

$a_{14,11} = -1.4347746786079132075288717350390 82$ $a_{14,12} = -5.0643147421228928984648638095452 09$

$a_{14,13} = -0.0235296471933197364377016509326 85$

$a_{15,1} = 0.8101674645254722523232876762637 53$ $a_{15,6} = -11.9284124567795589827279131601200 63$

$a_{15,7} = 5.3643128984903605546259124213852 67$ $a_{15,8} = 8.0599544597595117957572286355321 22$

$a_{15,9} = 4.6180634390017214837581602867140 57$ $a_{15,10} = 0.1128601659181894505188796244218 55$

$a_{15,11} = -1.3152077069450029030173879131808 65$ $a_{15,12} = -4.6898923999036091255542977392384 43$

$a_{15,13} = 0.0067213745543296816530362403580 01$ $a_{15,14} = -0.0385672386214142073369060721356 86$

$a_{16,1} = 0.7081867006810967019347772392865 65$ $a_{16,6} = -11.3511240435918967394614365824112 66$

$a_{16,7} = 4.4398368527636195666513863031280 39$ $a_{16,8} = 7.1605289668558467803306883326942 10$

$a_{16,9} = 4.9447522890661813895395199119760 94$ $a_{16,10} = 0.0981953561086798742592703501940 55$

$a_{16,11} = -1.0238357391589517664465203400316 95$ $a_{16,12} = -3.9351856990315794334197222354183 54$

$a_{16,13} = -0.0218626037455164619891723846941 87$ $a_{16,14} = -0.0194920799474799113987905947234 60$

References

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

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

[3] J. C. Butcher, The Numerical Analysis of Ordinary Differential Equations, *John Wiley & Sons Inc.,* 1987, New York.

[4] E. Fehlberg, Classical Fifth-, Sixth-, Seventh-, and Eighth-Order Runge-Kutta formulas with stepsize control, *NASA TR R*-287(1969), G. C. Marshall Space Flight Center, Juntsville, Ala.

[5] E. Hairer, A Runge-Kutta method of order 10, *J. Inst. Math. Appl.* **21**(1978) 47-59.

[6] E. Hairer, S. P. Norsett and G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Second edition, Springer-Verlag, Berlin Heidelberg, 1993.

[7] 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) 603-637.

[8] S. N. Papakostas, Algebraic analysis and development of numerical ODE solvers of the Runge-Kutta types(in Greek), *PhD desertation*, National Technical University, Athens, 1996.

[9] S. N. Papakostas, Ch. Tsitouras and G. Papageorgiou, A new family of efficient 6(5) Runge-Kutta pairs, *Hermis 92* E. Lipitakis, ed. Athens, 1992, pp.515-516.

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

[11] S. N. Papakostas and Ch. Tsitouras, High phase-lag order Runge-Kutta and Nyström pairs, *SIAM J. Sci. Comput.* **20** (1999)

[12] P. J. Prince and J. R. Dormand, High order embedded Runge-Kutta formulae, *J. Comput. Appl. Math.* **7**(1981) 67-75.

[13] Ch. Tsitouras, A parameter study of explicit Runge-Kutta pairs of orders 6(5), *Appl. Math. Lett.* **11**(1998) 65-69.

[14] Ch. Tsitouras and S. N. Papakostas, Cheap Error Estimation for Runge-Kutta pairs, *SIAM J. Sci. Comput.* **20** (1999)

[15] J. H. Verner, Explicit Runge-Kutta methods with estimates of the local truncation error, *SIAM J. Numer. Anal.* **15** (1978) 772-790.

[16] J. H. Verner, High-order explicit Runge-Kutta pairs with low stage order, *Appl. Numer. Math.* **22**(1996) 345-357.

[17] J. H. Verner, private communication, 1999.

[18] M. N. Vrahatis, G. S. Androulakis and G. E. Manoussakis, A new unconstrained optimization method for imprecise function and gradient values, *J. Math. Anal. Appl.* **197** (1996) 586-607.