Noname manuscript No.
(will be inserted by the editor)

Ch. Tsitouras

Runge-Kutta interpolants for high precision computations.

Received: date / Accepted: date

Abstract Runge-Kutta (RK) pairs furnish approximations of the solution of an initial value problem at discrete points in the interval of integration. Many techniques for enriching these methods with continuous approximations have been proposed. Here we constuct $8-$th and $9-$th order interpolation methods for a recently appeared RK pair of orders 9(8). It is the first time presented in the literature such a high accuracy dense output methods for use at quadruple precision, i.e. $32-33$ decimal digits of accuracy. Extended numerical results justify our effort.

Keywords Initial Value Problem · Dence Output · Quadruple precision.

Mathematics Subject Classification (2000) MSC 65L05 · 65L06

1 Introduction

We consider the numerical solution of the non-stiff initial value problem,

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

where the function $f : \Re \times \Re^m \to \Re^m$ is assumed to be as smooth as necessary. Traditionally, explicit embedded Runge-Kutta methods produce an approximation to the solution of (1) only at the end of each step. However, many applications require a continuous approximation to $y(x)$. These include differential equations with deviating arguments, problems with discontinuities or singularities, delay differential equations and the need for the numerical solution at a dense set of output points for graphical representation of the solution.

About 25 year ago a great interest for Runge-Kutta interpolation was emerged. Gladwell [11] and Horn [15,16] were the first to introduce solutions on this subject. Gladwell proposed standard Hermite interpolation over many steps while Horn after an interesting modification of the underlying Runge-Kutta method, produced the so called scaled Runge-Kutta method. The latter methods provided continuous extension to the solution, making use of the intriguing nature of Runge-Kutta methods. A little later Shampine [19] and Enright et. al. [10] proposed a kind of one-step Hermite interpolation using off-step points as extra stages of the basic method.

At the same time Dormand and Prince [7,8] presented formulas for their Runge-Kutta-(Nyström) pairs. In the begining of the $90'$s Tsitouras and Papageorgiou [22] proposed using two step interpolants since their truncation error is kept low. Interesting add-ons in the subject were done by Calvo et. al. [6], Owren and Zennaro [17] and Gladwell et al. [12], Sharp and Verner [20] among others.

Ch. Tsitouras
Department of Applied Sciences, TEI of Chalkis, GR34400, Greece
Tel.: +30-6974321039
Fax: +30-2228023766
E-mail: tsitoura@teihal.gr

All the above mentioned papers had dealt with low order methods. The construction of higher order ones is very complicated and many extra stages have to be inserted for achieving accuracy of the same magnitude with the basic formula. Bogacki and Shampine [2] constructed interpolants of order 5 for the Prince-Dormand 8(7) pair [18], that do not involve any additional evaluations. They restricted the step-size of the pair in order the interpolant can follow the accuracy of the underlying pair.

Verner [24,25] presented compact theory how to derive scaled extensions of orders up to nine. However the numbers he gives in [24] for his 9(8) Runge-Kutta pair [23], were accurate to $10-12$ digits only. Baker et. al. [1] gave 8−th order continuous extensions for the Prince-Dormand 8(7) pair [18]. Although they give in their site[1] coefficients in $32-38$ digits, they satisfy the corresponding Truncation errors only to 16 digits. So their suggestion is valid only for double precision computations.

There are some other interesting papers in this field but definitively nothing important has been published the last decade. We fill that there is some interest [13] in designing high order scaled extensions of Runge-Kutta pairs for use in quadruple precision arithmetic. Thus we decided to produce 8−th and 9−th order interpolants for our 9(8) pair and give the result in assuredly $32-33$ digits of accuracy.

Such an accuracy is useful in astronomical applications like LISA program which is space mission to be launched jointly by ESA[2] and NASA[3] around 2017 with the aim of detecting gravitational waves in the frequency range 0.1 mHz to 0.1 Hz and thus opening a completely new field of astronomy.

LISA consists of 3 spacecraft independently flying in orbits around the sun, similar to the Earth orbit but trailing behind the earth by about 20 degree (50 million km). The 3 spacecraft form a triangle with 5 million km armlength. Laser interferometry with pm accuracy in the relevant frequency range will monitor the distance between all three spacecraft, which employ drag-free technology to eliminate disturbing external forces such as solar radiation, magnetic fields etc. Ideally the orbits would evolve such that the triangle moves as a stiff formation around the sun, with an additional rotation around its center, but with constant armlengths and angles.

Orbital dynamics in the solar system cause disturbances to the triangular formation which are expected to be of the order of 1% variation in the armlengths, 1.5 degrees in the angles (nominally 60 degrees), and relative velocities of 15 m/s during a mission duration of 10 years. All these imperfections cause significant complications in the design of the laser interferometer and must hence be minimized. Since the spacecraft must follow pure gravitational orbits, station keeping maneuvers are not possible, and the only available degrees of freedom are the initial conditions of the orbits. Studying the LISA orbits we need integrators that reliably deliver a very high accuracy in the smallest possible computation time and thus are able to predict spacecraft orbits in the solar system to some meters precision over 10 years. Since the evaluation of the right-hand side of the corresponding differential equation is rather expensive (it involves looking up the positions of all planets in ephemerides files), high accuracy interpolation is attractive indeed [13].

2 Preliminaries

The general $s-$stage embedded Runge-Kutta pair of orders $p(p-1)$, for the approximate solution of the problem (1) can be defined by the following Butcher scheme [4,5]

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

where $A \in \Re^{s\times s}$, is strictly lower triangular, $b^T,\ \hat{b}^T,\ c \in \Re^s$ with $c = A \cdot e$, $e = [1,\ 1, \cdots, 1]^T \in \Re^s$. The vectors $\hat{b},\ b$ define the coefficients of the $(p-1)-$th and $p-$th order approximations respectively.

Starting with a given value $y(x_0) = y_0$, this method produces approximations at the mesh points $x_0 < x_1 < x_2 < \cdots < x_f$. Throughout this paper, we assume that local extrapolation is applied,

[1] http://www.scm.tees.ac.uk/j.r.dormand/rkcoeff.dat
[2] http://lisa.esa.int/
[3] http://lisa.jpl.nasa.gov/

hence the integration is advanced using the $p-$th order approximation. For estimating the error, two approximations are evaluated at each step x_n to $x_{n+1}=x_n+h_n$. These are:

$$\hat{y}_{n+1}=y_n+h_n\sum_{j=1}^{s}\hat{b}_j f_j \ \text{ and } \ y_{n+1}=y_n+h_n\sum_{j=1}^{s}b_j f_j,$$

where

$$f_i=f(x_n+c_ih_n,\ y_n+h_n\sum_{j=1}^{i-1}a_{ij}f_j),\ \ i=1,2,\cdots,s. \tag{2}$$

The local error estimate $E_n=\|y_n-\hat{y}_n\|$ of the $(p-1)-$th order Runge-Kutta pair is used for the automatic selection of the step size. Given a Tolerance $TOL>E_n$, the algorithm:

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

furnishes the next step length. In case $TOL<E_n$ then we reject the current step and try again with the left side of above formula being h_n.

Let $y_n(x)$ be the solution of the local initial value problem,

$$y_n'(x)=f(x,y_n(x)),\ x\geq x_n,\ y_n(x_n)=y_n.$$

Then E_{n+1} is an estimate of the error in the local solution $y_n(x)$ at $x=x_{n+1}$. The local truncation error t_{n+1} associated with the higher order method is

$$t_{n+1}=y_{n+1}-y_n(x_n+h_n)=\sum_{q=1}^{\infty}h_n^q\sum_{i=1}^{\lambda_q}T_{qi}P_{qi}=h_n^{p+1}\Phi(x_n,y_n)+O(h_n^{p+1}),$$

where $T_{qi}=Q_{qi}-\xi_{qi}/q!$ with Q_{qi} algebraic functions of $A,\ b,\ c$ and ξ_{qi} positive integers. P_{qi} are differentials of f evaluated at (x_n,y_n) and $T_{qi}=0$ for $q=1,2,\cdots,p$ and $i=1,2,\cdots,\lambda_q$. λ_q is the number of elementary differentials for each order and coincides with the number of rooted trees of order q. It is known that

$$\lambda_1=1,\ \lambda_2=1,\ \lambda_3=2,\ \lambda_4=4,\ \lambda_5=9,\ \lambda_6=20,\ \lambda_7=48\cdots,\ \text{etc [3]}.$$

The set $T^{(q)}=\{T_{q1},\ T_{q2},\cdots,T_{q,\lambda_q}\}$ is formed by the $q-$th order truncation error coefficients. It is usual practice a $(q-1)-$th order method to have minimized

$$\|T^{(q)}\|_2=\sqrt{\sum_{j=1}^{\lambda_q}T_{qj}^2}.$$

The derivation of a dense formula may be achieved in two ways. One is that developed by Horn [15,16]. This is an extension of the idea of an embedded pair where a third formula must be added to the Runge-Kutta pair. The resulting method is designed to integrate from x_n to $x_n+th_n,\ t\in(0,1)$ yielding non-mesh approximations to the solution $y(x_n+th_n)$ using the function evaluations of the original Runge-Kutta pair as the core of the new system, and is called a Scaled-Runge-Kutta method. The Scaled-Runge-Kutta method may be defined using the following tableau of coefficients,

$$\begin{array}{c|c} \widetilde{c} & \widetilde{A} \\ \hline & \widetilde{b}(t) \end{array}$$

where $\widetilde{A}\in\Re^{\widetilde{s}\times\widetilde{s}}$, $\widetilde{b}(t),\ \widetilde{c}\in\Re^{\widetilde{s}}$ and $\widetilde{s}\geq s$. Actually the left upper corner of $\widetilde{A}$ is the matrix of coefficients for the underlying pair A, and the first s elements of $\widetilde{c}$ is formed by original vector c. The quantities $\widetilde{b}(t)$ are polynomials of t of a suitable degree. Thus the dense approximation is then described by

$$y(x_n+th_n)\simeq\widetilde{y}_{n+t}=y_n+h_n\sum_{j=1}^{\widetilde{s}}\widetilde{b}_j(t)\widetilde{f}_j,$$

with $\widetilde{f}$ evaluated as in (2).

The extra coefficients $\widetilde{c},\ \widetilde{A},\ (i > s)$ needed for the scaled formula are independent of t in the case of a piecewise dense output method, while they depend on t if we want an intermediate approximation at a prescribed point. The derivation of the coefficients depends on the solution of the scaled equations of condition. The scaled equations of $q-$th order have the form:

$$\widetilde{T}_{qi} = \widetilde{Q}_{qi} - t^q \xi_{qi}/q! = 0$$

with $\widetilde{Q}$ algebraic functions of $\widetilde{A},\ \widetilde{b}$ and $\widetilde{c}$.

So for a fourth order continuous extension of a Runge-Kutta method we must satisfy the eight order conditions[4]:

$$\widetilde{b}\widetilde{e} = t, \qquad \widetilde{b}\widetilde{c} = \tfrac{t^2}{2}, \qquad \tfrac{1}{2}\widetilde{b}\widetilde{c}^2 = \tfrac{t^3}{6}, \qquad \widetilde{b}\widetilde{A}\widetilde{c} = \tfrac{t^3}{6},$$

$$\tfrac{1}{6}\widetilde{b}\widetilde{c}^3 = \tfrac{t^4}{24}, \quad \tfrac{1}{2}\widetilde{b}\widetilde{A}\widetilde{c}^2 = \tfrac{t^4}{24}, \quad \widetilde{b}\left(\widetilde{c} * \left(\widetilde{A}\widetilde{c}\right)\right) = \tfrac{t^4}{8}, \quad \widetilde{b}\widetilde{A}^2\widetilde{c} = \tfrac{t^4}{24}.$$

In consequence

$$\widetilde{T}_{1,1} = \widetilde{b}\widetilde{e} - t,\ \widetilde{T}_{2,1} = \widetilde{b}\widetilde{c} - t^2/2 \cdots,\ \widetilde{T}_{4,1} = \frac{1}{6}\widetilde{b}\widetilde{c}^3 - \frac{t^4}{24} \cdots,\ \widetilde{T}_{4,4} = \widetilde{b}\widetilde{A}^2\widetilde{c} - \frac{t^4}{24} \cdots$$

For our convenience we will use tildes in the rest of the paper only for b, s, e, and T's.

A second way of supplying a Runge-Kutta method with a dense formula is by constructing an interpolating polynomial of Hermite type. Shampine[19], Gladwell et al. [12] and Enright et al. [10], have proposed continuous extensions for Runge-Kutta pairs using this type of interpolation. The construction is based on values from an integration step. Therefore, assuming that the step $[x_n,\ x_{n+1}]$ has been completed successfully, the only available values are $y_n,\ y'_n = f_1,\ y_{n+1}$ and $y'_{n+1} = f_1$ of the next step. Assuming that another $g-4$ proper approximations of $y(x)$ or $y'(x)$ in the interval of interest are available, then a Hermite interpolation of degree $O(h^g)$ (usually $g = p+1$ or $g = p$) can be written as

$$U(x_n + th_n) = d_1(t)y_n + d_2(t)y_{n+1} + \sum_{j=3}^{k} d_j(t)y_{n+t_j} + h_n\left(d_{k+1}(t)y'_n + d_{k+2}(t)y'_{n+1} + \sum_{j=k+3}^{g} d_j(t)y'_{n+t_j}\right)$$

where $y_{n+t_3},\ y_{n+t_4}, \cdots, y_{n+t_k}$ are $O(h^g)$ approximations of the corresponding values $y(x_n + t_3h_n)$, $y(x_n + t_4h_n),\ \cdots\ y(x_n + t_kh_n)$ while $y'_{n+t_{k+3}},\ y'_{n+t_{k+4}}, \cdots, y'_{n+t_g}$ are $O(h^{g-1})$ approximations of the corresponding derivatives $y'(x_n + t_{k+3}h_n),\ y'(x_n + t_{k+4}h_n),\ \cdots\ y'(x_n + t_gh_n)$, and the polynomials $d_i(t),\ i = 1, 2, \cdots, g$ are of degree $g-1$.

The general algorithm for deriving this type of interpolation is given by the following formula:

$$U(x_n + th_n) = \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \\ \vdots \\ \\ \vdots \\ t^{g-2} \\ t^{g-1} \end{bmatrix}^T \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0\ 0\ 0\ 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1\ 1\ 1\ 1 & 1 & 1 \\ 1 & t_1 & t_1^2 & \cdots & & \cdots & t_1^{g-1} \\ & \vdots & & & & \vdots & \\ 1 & t_k & t_k^2 & \cdots & & \cdots & t_k^{g-1} \\ 0 & 1 & 0 & 0 & 0\ 0\ 0\ 0 & 0 & 0 \\ 0 & 1 & 2 & \cdots & \cdots\ \cdots & g-2 & g-1 \\ 0 & 1 & 2t_{k+3} & 3t_{k+3}^2 & \cdots\ \cdots & (g-2)\,t_{k+3}^{g-3} & (g-1)\,t_{k+3}^{g-2} \\ & \vdots & & & & \vdots & \\ 0 & 1 & 2t_g & 3t_g^2 & & (g-2)\,t_g^{g-3} & (g-1)\,t_g^{g-2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} y_n \\ y_{n+1} \\ y_{n+t_1} \\ \vdots \\ y_{n+t_k} \\ h_n y'_n \\ h_n y'_{n+1} \\ h_n y'_{n+t_{k+3}} \\ \vdots \\ h_n y'_{n+t_g} \end{bmatrix}$$

In the present paper we are interested in deriving eighth and ninth order interpolants for the 16$-$stages Runge-Kutta pair of orders 9(8) given in [21]. That pair outperformed all others, when high accuracy was required. So it is useful to embed such interpolants in this pair. The eighth order scaled extension uses four extra stages raising 25% the computational cost per step. The ninth order interpolant uses nine extra stages adding a 56% of evaluations per step.

[4] " $*$ " is a component-wise multiplication of vectors, and if $v = [v_1,\ v_2, \cdots]^T$ and $u = [u_1,\ u_2, \cdots]^T$ then $v * u = [v_1u_1,\ v_2u_2,\ v_3u_3, \cdots]^T$. Similarly $v^2 = v * v$, $v^3 = v * v * v$, etc.

3 Eighth order interpolant

Verner [24], claims that only five function evaluations are needed for derivation of eighth order interpolant for an 16−stages, 9(8) pair. One of them is actually the function f_1 of the next step, so it is not an extra one. Effectively only four extra stages are needed.

As we mention in the previous section there are two ways for constructing this interpolation. Classical Hermite interpolation needs six extra stages per step. One of them is used for constructing a seventh order ($O(h^8)$) scaled extension. Then the four data available at grids and five additional derivative evaluations of seventh order may form an eighth order interpolation. So it seems unattractive to implement the interpolation with this technique.

In order to proceed with our implementation, we try every additional stage to attach the highest possible stage order. Finally after only four additions we reach the required accuracy.

According to the previous notification we have $\widetilde{s} = 21$. The first real additional stage is the eighteenth. Let's note

$$a_{18} = [a_{18,1}, 0, 0, 0, 0, 0, 0, a_{18,8}, \cdots, a_{18,17}, 0, 0, 0, 0]$$

and $\widetilde{e} = [1, 1, \cdots, 1]^T \in \Re^{21}$. After setting the useless coefficients $a_{18,13} = a_{18,15} = 0$, then the 10 free parameters ($a_{18,1}$, $a_{18,8}, \cdots$, $a_{18,17}$ and c_{18}), may help to achieve seventh stage order. This means that we try so $f_{18} = y'(x_n + c_{18}h_n) + O(h^8)$. Thus we require:

$$\begin{array}{lllll} a_{18}\widetilde{e} = c_{18}, & a_{18}c = \frac{c_{18}^2}{2}, & a_{18}c^2 = \frac{c_{18}^3}{3}, & a_{18}c^3 = \frac{c_{18}^4}{4}, & a_{18}c^4 = \frac{c_{18}^5}{5}, \\ a_{18}c^5 = \frac{c_{18}^6}{6}, & a_{18}c^6 = \frac{c_{18}^7}{7}, & a_{18}A^3c^3 - \frac{c_{18}^7}{840}, & a_{18}A^2c^4 - \frac{c_{18}^7}{210}, & a_{18}Ac^5 - \frac{c_{18}^7}{42}. \end{array}$$

All the equations are linear in a_{18i}'s. We solve nine of them for a_{18i}'s and the tenth equation reduces to a polynomial in c_{18}.

The nineteenth stage offers twelve free coefficients after setting $a_{19,13} = 0$. Namely $a_{19,1}$, $a_{19,8}, \cdots$, $a_{19,18}$ and c_{19}. We satisfy all seventh order conditions and 58 equations of a total 115 conditions of eighth order. The equations are:

$$\begin{array}{cccc} a_{19}\widetilde{e} = c_{19}, & a_{19}c = \frac{c_{19}^2}{2}, & a_{19}c^2 = \frac{c_{19}^3}{3}, & a_{19}c^3 = \frac{c_{19}^4}{4}, \\ a_{19}c^4 = \frac{c_{19}^5}{5}, & a_{19}c^5 = \frac{c_{19}^6}{6}, & a_{19}c^6 = \frac{c_{19}^7}{7}, & a_{19}c^7 = \frac{c_{19}^8}{7} \\ \multicolumn{4}{c}{a_{19}A^3c^3 = \frac{c_{19}^7}{840}, a_{19}A^2c^4 = \frac{c_{19}^7}{210}, a_{19}Ac^5 = \frac{c_{19}^7}{42}, a_{19}\left(c * \left(Ac^5\right)\right) = \frac{c_{19}^8}{48}.} \end{array} \tag{3}$$

Again the eleven equations are linear on $a_{19,i}$'s and can be solved for them. The final equation is a polynomial on c_{19}.

The twentieth stage offers one coefficient more after setting the useless $a_{20,13} = 0$. Then we solve the equations (3) with the obvious modification for a_{20} and we conclude solving the additional equation

$$a_{20}\left(c * \left(A^2c^4\right)\right) = \frac{c_{20}^8}{240},$$

which reduces to a polynomial on c_{20}. The parameters of the 20−th stage satisfy seventh stage order and 63 equations of eighth order.

The twenty-first stage offers one additional coefficient (here again $a_{21,13} = 0$) which is used to solve the extra equation

$$a_{21}\left(c * \left(A^3c^3\right)\right) = \frac{c_{21}^8}{960}.$$

After solving this the 21−st stage is of eighth stage order. Thus $f_{21} = y'(x_n + c_{21}h_n) + O(h^9)$. Demanding this accuracy for f_{21} we get an extra free stage for implementing a ninth order Hermite-type interpolation. Observe that Verner's corresponding stage does not achieve eighth stage order [24].

The values of c that satisfy the equations needed are:

$$c_{18} = \frac{299812227498768379}{404366309217245371}, \quad c_{19} = \frac{1097291963064184168}{1282743498329518525},$$

$$c_{20} = \frac{128408850005288355}{200075732634920597}, \quad c_{21} = \frac{1141960682686836856}{2245632444134769313}.$$

The form of the scaled extension is:

$$y\left(x_n + th_n\right) \simeq \widetilde{y}_{n+t} = \; y_n + h_n \cdot \sum_{i=1}^{21} \widetilde{b}_i(t) \cdot f_i,$$

with $\widetilde{b}_i(t) = \sum_{j=0}^{8} \widetilde{b}_{ij} \cdot t^j$ and

$$\widetilde{b}_2(t) = \widetilde{b}_3(t) = \widetilde{b}_4(t) = \widetilde{b}_5(t) = \widetilde{b}_6(t) = \widetilde{b}_7(t) = 0.$$

For determining the coefficients of the fifteen non zero polynomials $\widetilde{b}$ we have to solve the following fourteen equations of condition:

$\widetilde{b}\widetilde{e} = t,\quad \widetilde{b}c = \frac{t^2}{2},\quad \widetilde{b}c^{\,2} = \frac{t^3}{3},\quad \widetilde{b}c^{\,3} = \frac{t^4}{4},\quad \widetilde{b}c^{\,4} = \frac{t^5}{5},$

$\widetilde{b}c^{\,5} = \frac{t^6}{6},\quad \widetilde{b}c^6 = \frac{t^7}{7},\quad \widetilde{b}c^7 = \frac{t^8}{8},\quad \widetilde{b}A^3c^3 = \frac{t^7}{840},\quad \widetilde{b}A^2c^4 = \frac{t^7}{210},$

$\widetilde{b}Ac^{\,5} = \frac{t^7}{42},\quad \widetilde{b}A^4c^{\,3} = \frac{t^8}{6720},\quad \widetilde{b}\left(c * \left(A^2c^{\,4}\right)\right) = \frac{t^8}{240},\quad \widetilde{b}\left(c * \left(Ac^{\,5}\right)\right) = \frac{t^8}{48}.$

These equations are linear in $\widetilde{b}$ and can be solved simultaneously leaving one polynomial as free parameter. This polynomial (say $\widetilde{b}_{15}$) has 9 coefficients. Two of them are needed for satisfying C^0 continuity. For this property we ask:

$$\widetilde{b}_i(0) = 0,\; i = 1, 2, \cdots, 21, \text{ and } \widetilde{b}_i(1) = b_i,\; i = 1, 2, \cdots, 16,\; \widetilde{b}_i(1) = 0,\; i = 17, \cdots, 21.$$

Polynomial $\widetilde{b}_{15}(t)$ contributes with only one parameter in the left set of the above equations since all polynomials $b_i,\ i = 1, 2, \cdots$ share no constant coefficient.

Then we proceed determining another two coefficients of the free polynomial for C^1 continuity:

$$\frac{d\widetilde{b}_1\,(t)}{dt}|_{t=0} = 1,\; \frac{d\widetilde{b}_i\,(t)}{dt}|_{t=0} = 0,\, i = 2, 3, \cdots, 21$$
$$\frac{d\widetilde{b}_{17}\,(t)}{dt}|_{t=1} = 1,\; \frac{d\widetilde{b}_i\,(t)}{dt}|_{t=1} = 0,\, i \neq 17$$

Finally five coefficients remain for minimizing the truncation error coefficients of ninth order. Since these terms depend on t, we integrate the Euclidean norm of them in the interval $[0,\ 1]$

$$\int\limits_{t=0}^{t=1} \|\widetilde{T}^{(9)}\|_2 dt = \int\limits_{t=0}^{t=1} \left(\sqrt{\widetilde{T}^2_{9,1}\,(t) + \widetilde{T}^2_{9,2}\,(t) + \cdots \widetilde{T}^2_{9,286}\,(t)}\right) dt \tag{4}$$

The coefficients of extended matrix A can be found in Appendix Tables 3 and 4, while the coefficients of the polynomials $\widetilde{b}$ are listed in Tables 5, 6, 7, 8, 9, 10, 11 and 12.

In Figure 1, we plot the value $\|\widetilde{T}^{(9)}\|_2$ as a function of t and we observe that is kept under the corresponding value of the underlying 8−th order method, $\|T^{(9)}\|_2 \approx 1.25 \cdot 10^{-5}$.

4 Ninth order interpolant

The ninth order scaled Runge-Kutta method needs five extra stages according to Verner[24]. But a great loss in the accuracy of the coefficients is experienced during this procedure. Classical Hermite interpolation would need six derivative evaluations more in order to reach the desired ten data information in the interval $[x_n, x_{n+1}]$. But our gain from the 8−th order interpolation is that f_{21} is of eighth order. So using the always available

$$y_n,\; y_n' = f_1,\; y_{n+1}\;\; y_{n+1}' = f_1 \text{ of the next step}$$

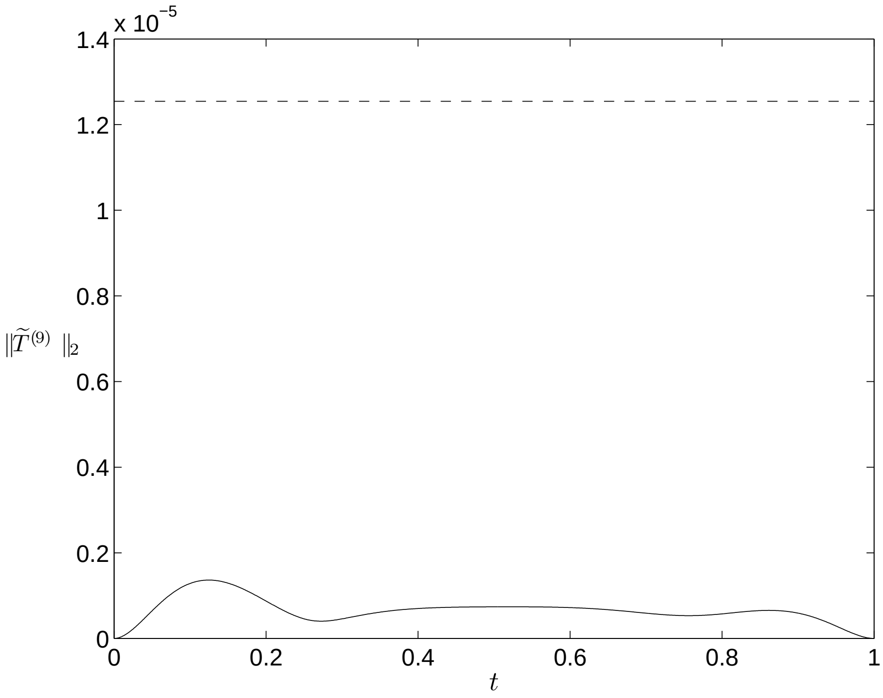

Fig. 1 $\|\widetilde{T}^{(9)}\|_2$ for 8-th order interpolant as function of t. The dashed line corresponds to the underlying method of 8−th order

and the eighth stage order approximations:

$$f_{21} = f(x_n + c_{21}h_n, y_n + h_n \sum_{j=1}^{j=20} a_{21j} f_j) = y'(x_n + c_{21}h_n) + O(h^9),$$

$$f_i = f(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{j=21} \widetilde{b}_j(c_i) f_j) = y'(x_n + c_i h_n) + O(h^9),\ 26 \geq i \geq 22,$$

we may form the interpolant[5]

$$\begin{aligned} y\left(x_n + th_n\right) \simeq \widetilde{\widetilde{y}}_{n+t} = {} & d_1 y_n + h_n d_2 y'_n + h_n d_3 f_{21} + h_n d_4 f_{22} + h_n d_5 f_{23} \\ & + h_n d_6 f_{24} + h_n d_7 f_{25} + h_n d_8 f_{26} + d_9 y_{n+1} + h_n d_{10} y'_{n+1}. \end{aligned} \tag{5}$$

The polynomial coefficients of this interpolant are given by,

$$\begin{bmatrix} d_1(t) \\ d_2(t) \\ d_3(t) \\ d_4(t) \\ d_5(t) \\ d_6(t) \\ d_7(t) \\ d_8(t) \\ d_9(t) \\ d_{10}(t) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 1 \; t \; t^2 \; \cdots \; t^9 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2c_{21} & 3c_{21}^2 & 4c_{21}^3 & 5c_{21}^4 & 6c_{21}^5 & 7c_{21}^6 & 8c_{21}^7 & 9c_{21}^8 \\ 0 & 1 & 2c_{22} & 3c_{22}^2 & 4c_{22}^3 & 5c_{22}^4 & 6c_{22}^5 & 7c_{22}^6 & 8c_{22}^7 & 9c_{22}^8 \\ 0 & 1 & 2c_{23} & 3c_{23}^2 & 4c_{23}^3 & 5c_{23}^4 & 6c_{23}^5 & 7c_{23}^6 & 8c_{23}^7 & 9c_{23}^8 \\ 0 & 1 & 2c_{24} & 3c_{24}^2 & 4c_{24}^3 & 5c_{24}^4 & 6c_{24}^5 & 7c_{24}^6 & 8c_{24}^7 & 9c_{24}^8 \\ 0 & 1 & 2c_{25} & 3c_{25}^2 & 4c_{25}^3 & 5c_{25}^4 & 6c_{25}^5 & 7c_{25}^6 & 8c_{25}^7 & 9c_{25}^8 \\ 0 & 1 & 2c_{26} & 3c_{26}^2 & 4c_{26}^3 & 5c_{26}^4 & 6c_{26}^5 & 7c_{26}^6 & 8c_{26}^7 & 9c_{26}^8 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}^{-1}.$$

[5] For the ninth order interpolant we use double tildes to distinguish it from eighth order scaled extension.

Since

$$\widetilde{\widetilde{y}}_{n+t} = d_1 y_n + h_n d_2 f_1 + h_n d_3 f_{23} + h_n d_4 f_{24} + h_n d_5 f_{25} + h_n d_6 f_{26}$$
$$+ h_n d_7 f_{27} + h_n d_8 f_{28} + d_9 (y_n + h_n \sum_{j=1}^{16} b_j f_j) + h_n d_{10} f_{17},$$

the formula (5) can be transformed to a scaled Runge-Kutta like scheme:

$$y\left(x_n + t h_n\right) \simeq \widetilde{\widetilde{y}}_{n+t} = \; y_n + h_n \cdot \sum_{i=1}^{26} \widetilde{\widetilde{b}}_i(t) \cdot f_i,$$

with $\widetilde{\widetilde{s}} = 26,\ \widetilde{\widetilde{b}}_i(t) = \sum\limits_{j=0}^{9} \widetilde{\widetilde{b}}_{ij} \cdot t^j$ and

$$\widetilde{\widetilde{b}}_2(t) = \widetilde{\widetilde{b}}_3(t) = \widetilde{\widetilde{b}}_4(t) = \widetilde{\widetilde{b}}_5(t) = \widetilde{\widetilde{b}}_6(t) = \widetilde{\widetilde{b}}_7(t) = \widetilde{\widetilde{b}}_{18}(t) = \widetilde{\widetilde{b}}_{19}(t) = \widetilde{\widetilde{b}}_{20}(t) = 0.$$

Also observe that:

$$\begin{aligned} \widetilde{\widetilde{b}}_1 &= d_2 + b_1 d_9, \\ \widetilde{\widetilde{b}}_i &= d_9 b_i,\ i = 8, \cdots, 16, \\ \widetilde{\widetilde{b}}_{17} &= d_{10}, \\ \widetilde{\widetilde{b}}_i &= d_{i-18},\ i = 21, \cdots, 26. \end{aligned}$$

and

$$\begin{aligned} a_{ij} &= \widetilde{b}_j(c_i),\ i > 21, \\ a_{ij} &= 0,\ j > 21 \end{aligned}$$

This latter form give us the ability to analyze the 10−th order truncation error coefficients of the new interpolant. A choice of c's that minimizes $\int\limits_{t=0}^{t=1} \|\widetilde{\widetilde{T}}^{(10)}\|_2 dt$ is:

$$c_{22} = \frac{1}{23},\ c_{23} = \frac{4}{21},\ c_{24} = \frac{7}{24},\ c_{25} = \frac{9}{14},\ c_{26} = \frac{8}{11}.$$

In Figure 2, we plot the value $\|\widetilde{\widetilde{T}}^{(10)}\|_2$ as a function of t and we observe that is kept under the corresponding value of the underlying method, $\|T^{(10)}\|_2 \approx 3.61 \cdot 10^{-7}$.

5 Numerical Results

We run the Runge-Kutta pair for the 25 DETEST [14,9] problems and for tolerances 10^{-12}, 10^{-14}, $\cdots, 10^{-22}$. Quadruple precision arithmetic was used by an INTEL Visual Fortran Compiler 9.0, on a Pentium IV computer running Windows XP Professional at 3.4GHz.

The numbers found in the appendix are as high as 5000 for most of $\widetilde{b}$'s while the final value of all these polynomials is always smaller than 1/2 for every $t \in [0, 1]$. This may not affect the results for tolerances greater than $10^{-22} - 10^{-23}$, but it is generally known that additions or subtractions of big numbers must be avoided. Especially here where small final values are produced adding big ones. All these polynomials have roots no far from unit circle. The biggest modulus is about 1.2. So we may write $\widetilde{b}$'s as a product with factors evaluated using additions of small numbers. For example

$$\widetilde{b}_{21} \approx 633.37878381 t^2 (t - 0.18112285)(t - 0.50852519)(t^2 - 1.50210002 t + 0.58245960)(t - 1)^2.$$

For our tests we used this form which reduces the roundoff errors.

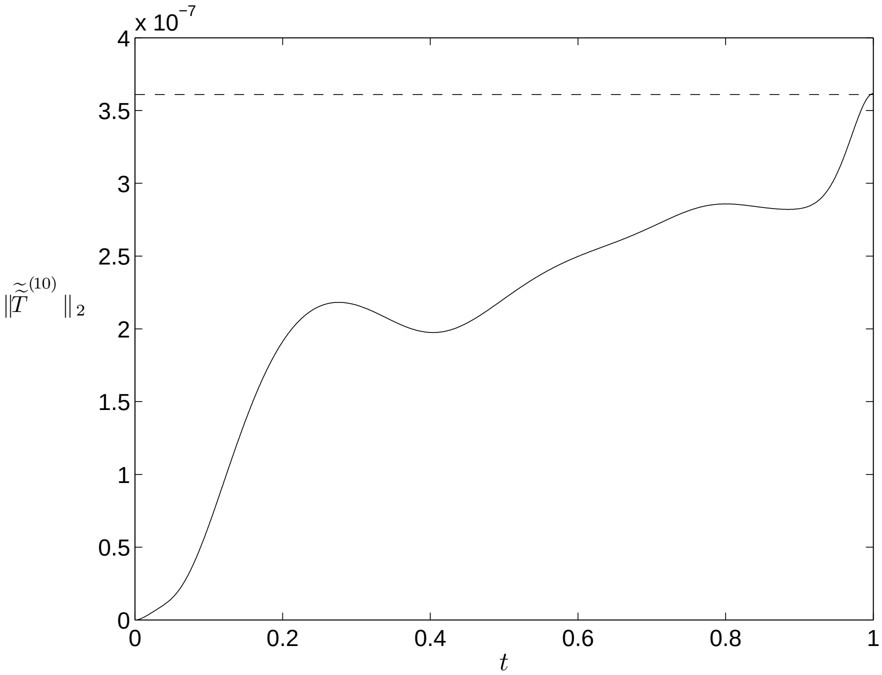

Fig. 2 $\|\widetilde{\widetilde{T}}^{(10)}\|_2$ for 9-th order interpolant as function of t. The dashed line corresponds to the underlying method of 9−th order

For both interpolants we computed the max-norm of the global error at 10 interpolation points $x_n + \frac{i}{10} \cdot h_n,\ i = 1, 2, \cdots, 10$ within its step $[x_n, x_{n+1}]$ generated by the Runge-Kutta pair. An almost exact approximation of the solution was computed internally by the same pair for a stringent tolerance each time. We recorded the ratio of the maximum global error that occurred at these interpolation points divided by the maximum global error that occurred at the grid points $[x_n,\ n \geq 1]$. This ratio shows how the global error produced by the interpolant relates to that of the associated formula. All the ratios are at least 1 since the interpolation points include the grid points.

From Table 1, we observe that the ratios for the 8−th order formula are in average as high as 11.2. For the 9−th order formula it is only 1.16 as shown in Table 2. An asymptotical estimation is given when taking in account only the results for tolerance 10^{-22}, where the ratios then are in average only 1.003 for the ninth order interpolant and 17.4 for the eighth order one. The latter figure falls to only 9.8 if we ignore problem $B3$. This means that in average, one decimal digit is gained by increasing the order of the interpolant. Thus the five extra stages justify this cost.

In case we use more conservative safety factor at step-size control algorithm i.e.

$$h_{n+1} = 0.8 \cdot h_n \cdot (TOL/EST)^{1/9},$$

we get better accuracy at a cost of more function evaluations. The overall efficiency does not affected more than ±1%. The corresponding average in ratios for the 9−th order interpolant raises to 1.23 and the eighth order interpolant looses relatively accuracy too, and the average increases to 16.0. This is not in contradiction with the observations of Bogachi and Shampine [2] who shortened the step length for their lower order interpolant to follow the accuracy of the method. For this case the accuracies of the interpolants in relation to the underlying method are worse but the absolute accuracy of the interpolation was better than before.

Finally we tried another type of 8−th order scaled extension requiring a natural choice

$$f_{21} = \widetilde{y}'_{n+c_{21}} = f(x_n + c_{21}h_n, y_n + h_n \sum_{j=1}^{20} \widetilde{b}_j(c_{21}) f_j).$$

For this we must satisfy,

$$a_{21j} = \widetilde{b}_j(c_{21}),\ j = 1, 2, \cdots 20.$$

This was achieved by spending only one of the free parameters of the polynomial $\widetilde{b}_{15}$ to get $\widetilde{b}_{21}(c_{21}) = 0 = a_{21,21}$. The four remaining parameters of $\widetilde{b}_{15}$ are used for minimization of the norm (4). The scaled Runge-Kutta method constructed this way gave similar results with the method proposed here (correlation factor 0.997) but it was in average slightly (say 5%) worse.

Table 1 Error ratios for $O(h^9)$ interpolant over DETEST.

	10^{-12}	10^{-14}	10^{-16}	10^{-18}	10^{-20}	10^{-22}
A1	4.46	3.87	6.66	5.26	6.11	6.48
A2	39.7	42.5	43.9	44.5	45.0	45.4
A3	1.03	1.01	1.06	1.01	1.61	3.36
A4	9.70	11.4	16.2	15.8	18.5	23.3
A5	4.16	3.89	2.30	3.82	3.51	5.29
B1	1.01	1.02	1.02	1.27	1.68	3.27
B2	6.31	5.75	6.15	4.99	6.04	6.04
B3	16.0	24.8	52.5	61.0	192	190
B4	1.01	1.01	1.01	1.01	1.04	1.09
B5	2.83	7.11	15.3	15.6	16.0	16.1
C1	3.27	3.42	3.71	3.92	4.05	4.14
C2	2.56	3.66	5.13	3.81	3.12	3.18
C3	7.17	4.75	6.50	6.45	6.67	6.80
C4	7.17	5.41	6.57	6.45	6.67	6.80
C5	3.14	4.93	6.46	3.71	2.68	2.38
D1	1.03	1.01	1.01	1.01	1.01	1.01
D2	1.00	1.00	1.00	1.00	1.00	1.00
D3	1.00	1.00	1.00	1.00	1.00	1.00
D4	1.00	1.00	1.00	1.00	1.00	1.00
D5	1.00	1.00	1.00	1.00	1.00	1.00
E1	1.17	1.95	2.37	2.70	2.89	3.02
E2	17.9	11.2	18.0	31.4	44.2	47.4
E3	11.6	3.03	2.24	3.04	2.74	2.48
E4	19.8	33.7	19.8	24.9	57.7	49.5
E5	2.74	2.81	2.87	2.89	2.90	2.91

6 Appendix

The coefficients for the 21−stage scaled scheme of eighth order. The numbers for the first 17 stages and $\hat{b}$ can be found in [21]. Missing coefficients are zero.

References

1. T. S. Baker, J. R. Dormand, J. E. Gilmore, P. J. Prince, Continuous approximation with embedded Runge-Kutta methods, Appl. Numer. Math., 22, 51-62 (1996)
2. P. Bogacki and L. F. Shampine, Interpolating high-order Runge-Kutta formulas, Comput. Math. Applic., 20, 15-24 (1990)
3. J. C. Butcher, Coefficients for the study of Runge-Kutta integration processes, J. Austr. Math. Soc., 3, 185-201 (1963)
4. J. C. Butcher, Implicit Runge-Kutta processes, Math. Comput., 18, 50-64 (1964)
5. J. C. Butcher, On Runge-Kutta processes of high order, J. Austral. Math. Soc., 4, 179-194 (1964)
6. M. Calvo, J. I. Montijano and L. Randez, A fifth order interpolant for the Dormand and Prince Runge-Kutta method, J. Comput. Appl. Math., 29, 91-100 (1990)
7. J. R. Dormand and P. J. Prince, Runge-Kutta triples, Comput. Math. Appl., 12, 1007-1017 (1986)
8. J. R. Dormand and P. J. Prince, Runge-Kutta-Nystrom triples, Comput. Math. Appl., 13, 937-949 (1987)
9. W. H. Enright and J. D. Pryce, Two Fortran Packages for Assessing Initial Value Methods, ACM TOMS, 13, 1-27 (1987)
10. W. H. Enright, K. R. Jackson, S. P. Norsett and P. G. Thomsen, Interpolants for Runge-Kutta formulas, ACM TOMS, 12, 193-218 (1986)

Table 2 Error ratios for $O(h^{10})$ interpolant over DETEST.

	10^{-12}	10^{-14}	10^{-16}	10^{-18}	10^{-20}	10^{-22}
A1	2.88	1.53	3.98	1.00	1.00	1.00
A2	1.10	1.08	1.06	1.04	1.03	1.01
A3	1.00	1.00	1.00	1.00	1.00	1.00
A4	1.04	1.00	1.00	1.00	1.00	1.00
A5	1.01	1.00	1.00	1.00	1.00	1.00
B1	1.00	1.00	1.00	1.00	1.00	1.00
B2	2.78	3.56	1.00	1.00	1.00	1.00
B3	3.61	2.63	2.16	1.00	1.02	1.00
B4	1.00	1.00	1.00	1.00	1.00	1.00
B5	1.00	1.00	1.00	1.00	1.00	1.00
C1	1.00	1.00	1.00	1.00	1.00	1.00
C2	3.29	1.83	1.00	1.00	1.00	1.00
C3	1.48	1.05	1.00	1.00	1.00	1.00
C4	1.48	1.00	1.00	1.00	1.00	1.00
C5	1.00	1.00	1.00	1.00	1.00	1.00
D1	1.00	1.00	1.00	1.00	1.00	1.00
D2	1.00	1.00	1.00	1.00	1.00	1.00
D3	1.00	1.00	1.00	1.00	1.00	1.00
D4	1.00	1.00	1.00	1.00	1.00	1.00
D5	1.00	1.00	1.00	1.00	1.00	1.00
E1	1.00	1.00	1.00	1.00	1.00	1.00
E2	1.01	1.01	1.10	1.04	1.13	1.06
E3	1.00	1.00	1.00	1.00	1.00	1.00
E4	1.93	1.00	1.00	1.00	1.00	1.00
E5	1.00	1.00	1.00	1.00	1.00	1.00

Table 3 The 18-th and 19-th rows of matrix A.

i	a_{18i}	a_{19i}
1	0.011432648061626831484473526087708431	0.013665530501599309301667313024545549
8	−0.25910015109834286167182847913811052	−0.30703861145677111921271332186217225
9	0.26112279003556667442851494524453053	0.25209531563572099623089165079301459
10	0.13603667739783981942567006544587345	0.13108277539906970515311706137497176
11	0.21092629039215214634081204784167201	0.21767739046525384968495567501048449
12	0.38561044391652919964360639985117800	0.44086627030905856048873271907826814
14	−0.0066120753217415785069185723457661197	0.036713802867286798088711574468169594
15	0	−0.00045263428475531956819180895665607424
16	−0.025491612884353702222118053281571882	−0.0075219364917745459172150727107695996
17	0.027512197954444024676795158249911643	0.0038178169251348210565771097937455971
18		0.074520142349051742400516673742958887

11. I. Gladwell, Initial Value Routines in the NAG Library, ACM TOMS, 5, 386-400 (1979)
12. I. Gladwell, L. F. Shampine, L. S. Baca and R. W. Brankin, Practical aspects of interpolation in Runge-Kutta codes, SIAM J. Sci. Stat. Comput., 8, 322-341 (1987)
13. G. Heinzel, Private communications (2006)
14. T. E. Hull, W. H. Enright, B. M. Fellen and A. E. Sedgwick, Comparing numerical methods of ordinary differential equations, SIAM J. Numer. Anal., 9, 603-637 (1972)
15. M. K. Horn, Scaled Runge-Kutta Algorithms for handling dense output, Rept. DFVLR-FB 81-13, Oberpfattenhonfen, FRG (1981)
16. M. K. Horn, Fourth and fifth-order, scaled Runge-Kutta algorithms for treating dense output, SIAM J. Numer. Anal., 20, 558-568 (1983)
17. B. Owren and M. Zennaro, Derivation of Efficient Continuous Explicit Runge-Kutta Methods, SIAM J. Sci. Stat. Comput., 13, 1488-1501 (1992)
18. P. J. Prince and J. R. Dormand, High order embedded Runge-Kutta formulae, J. Comput. Appl Math., 7, 67-75 (1981)
19. L. F. Shampine, Interpolation for Runge-Kutta methods, SIAM J. Numer. Anal., 22, 1014-1027 (1985)
20. P. W. Sharp and J. H. Verner, Generation of high-order interpolants for explicit Runge-Kutta pairs, ACM TOMS, 24, 13-29 (1998)
21. Ch. Tsitouras, Optimized explicit RungeKutta pair of orders 9(8), Appl. Numer. Math., 38, 123-134 (2001)

Table 4 The 20-th and 21-st rows of matrix A.

i	a_{20i}	a_{21i}
1	0.014268631497102842453094099400571112	0.013154167847783023947787964293933914
8	−0.24769768181800691975188712358712399	−0.34064115650712902848454767806522149
9	0.23928330553460883713418088067890939	0.22802117967789421808187939123391075
10	0.12957572153718905740865524351757453	0.13228168138941728498745754543959846
11	0.22078879969474376788381528067483098	0.21597957165624687772098521823642410
12	0.36428486485822390774749367125266646	0.51134550887345867838216854890276241
14	0.0078316665904274240545668452142219710	0.13917551936854006335051797420701257
15	0.000066960723091763590089655760622436567	−0.0031263601232930086179660228833873655
16	−0.014794348323796745679860880371768882	0.026185109965277331546702382098973556
17	0.013398599455567821157189637776957067	−0.033417534733963503459105659672079059
18	−0.085614642709593809011748490099335927	0.070815965257295717659359220856222485
19	0.00040934649811497276726885169474090132	−0.20353982182996909215211745048425036
20		−0.24770863310964368488502368393967138

Table 5 The coefficients forming $\widetilde{b}_1$ and $\widetilde{b}_8$. Notice that especially here $\widetilde{b}_{1,1} = 1$

j	$\widetilde{b}_{1j}$	$\widetilde{b}_{8j}$
2	−11.973451765066176560361311018226	43.2795603711957663314640687334784
3	61.8659433341341457414546419674870	−388.643615010266767652241553817573
4	−172.191284938824966462131456884724	1375.36134238344386516425494248983
5	278.420601619157308482597286774406	−2531.74963787167223481762103223525
6	−262.245961078958633613414158509672	2582.71150241910928426677685824245
7	133.625523142029815164255629875140	−1389.71139150818241656595496272621
8	−28.4864612916517081421757960333860	308.548158769451961760738188103935

Table 6 The coefficients forming $\widetilde{b}_9$ and $\widetilde{b}_{10}$.

j	$\widetilde{b}_{9j}$	$\widetilde{b}_{10j}$
2	−35.949229157515841308693212522485	14.507768855561170577399510684864115
3	344.66007145267323793304701239612813	−102.87334506907882369949499181653257
4	−1277.8231161852588793034483568957163	325.73620934962310722589632000654494
5	2438.4531969414645043419415426301071	−564.41475401881182444339789500457423
6	−2556.6772649423957779426859557911790	554.71818449744478179630222860534829
7	1403.5145365712080540414467219761373	−290.79278746482763407938150032377714
8	−315.94918029416959328896002709962148	63.246729432600696379368409963863892

Table 7 The coefficients forming $\widetilde{b}_{11}$ and $\widetilde{b}_{12}$.

j	$\widetilde{b}_{11j}$	$\widetilde{b}_{12j}$
2	−2.1094695519024646598908782765362982	−100.79375011459739760440959274234003
3	55.988251954788337958191095236032033	912.99125305463925121088944961549543
4	−264.61413210699096652631586537232704	−3280.6831578813275712931303068459504
5	565.65822385064758881538103386820147	6130.8565731590057666459772117731737
6	−633.38091626406568414487604426248265	−6333.8123210002439365588445068616605
7	362.74969708930989760924384724524554	3440.7580426866649922779331052966786
8	−84.067848703326167615235487768568199	−768.92110825120410413420983634383248

22. Ch. Tsitouras and G. Papageorgiou, Runge-Kutta interpolants based on values from two successive integration steps, Computing, 43, 255-266 (1990)
23. J. H. Verner, Explicit Runge-Kutta methods with estimates of the local truncation error, SIAM J. Numer. Anal., 15, 772-790 (1978)
24. J. H. Verner, Differentiable interpolants for High-order Runge-Kutta methods, Report 1990-9, Dept of Mathematics and Statistics, Queen's University, Ontario Canada (1990)
25. J. H. Verner, Differentiable interpolants for high-order Runge-Kutta methods, SIAM J. Numer. Anal., 30, 1446-1466 (1993)

Table 8 The coefficients forming $\widetilde{b}_{13}$ and $\widetilde{b}_{14}$.

j	$\widetilde{b}_{13j}$	$\widetilde{b}_{14j}$
2	−25.440507894872984550500436028067427	−53.044523766668953990865348337508229
3	231.56721238732705115521555888748081	472.06731168152030494976872425589296
4	−848.88135982430316686960349357413705	−1690.9474431179622415653053379847580
5	1619.4601461226664780921682769195806	3162.7606215432605215020698402851890
6	−1701.9954814871161810970517626589562	−3273.1311367235474224182518161777664
7	936.37628107675280541866295974978827	1780.7160806537701180827154030095558
8	−211.03212391286593233692541965030535	−398.29399587384786752327503119748342

Table 9 The coefficients forming $\widetilde{b}_{15}$ and $\widetilde{b}_{16}$.

j	$\widetilde{b}_{15j}$	$\widetilde{b}_{16j}$
2	0.069532904885953224280647815866529446	−16.273632561649042928632428258068406
3	−0.40836619698511108536380100612280752	144.74446484804145579419767803825056
4	0.64705882352941176470588235294117647	−520.09470500289185114515502593304132
5	0.13333333333333333333333333333333333	976.29689823342511601988001923608264
6	−1.2439024390243902439024390243902439	−1013.5010685886085561530538390289585
7	1.12	552.66779813432517968595841956802430
8	−0.31818181818181818181818181818181818	−123.80749699812617224093675910616022

Table 10 The coefficients forming $\widetilde{b}_{17}$ and $\widetilde{b}_{18}$.

j	$\widetilde{b}_{17j}$	$\widetilde{b}_{18j}$
2	20.625755176470875784534420237574253	−18.817476339237677341178965829020625
3	−186.72916690449931142805675512305950	53.566894625827769435485862257588867
4	684.34772688713531787857778116836424	250.05378570526817398980893871953673
5	−1313.8347634052464295313536128170233	−1346.8731688430031688644084422339689
6	1399.2597557186831307119727893837642	2334.7953069475911713312956770826992
7	−785.51482530649694191111861080006356	−1784.1158652809583151589572986554593
8	181.84551783395335849544398795044363	511.39052318451204660795422865862401

Table 11 The coefficients forming $\widetilde{b}_{19}$ and $\widetilde{b}_{20}$.

j	$\widetilde{b}_{19j}$	$\widetilde{b}_{20j}$
2	97.969961339224266808139818666364240	53.970113384226006859242965978811408
3	−844.22675982692345551884024227615845	−344.55913008769394871758868793496691
4	2908.4744283151958608474176819245470	709.14543312174029933838356908142433
5	−5186.9509883693411284355665884018688	−331.18678744759622490444568444043506
6	5068.1629115433598517009851177503704	−656.83707406929786479256385513892002
7	−2576.0565401402080847395788986107157	869.62774812753690937741613107541400
8	532.62698713869268933744311094746128	−300.16030302891517716044443862132775

Table 12 The coefficients forming $\widetilde{b}_{21}$

j	$\widetilde{b}_{21j}$	j	$\widetilde{b}_{21j}$
2	33.979349119946499359470740895293897	6	4493.1774654670702271573117063893514
3	−410.01102024350413607666399067994239	7	−2654.9642977809243792026409466797540
4	1801.4692144716236069560447277474610	8	633.37878381307778804303305201453814
5	−3897.0294948472896062365552896869481		