Vol. 44 No. 1 IPSJ Journal Jan. 2003

Regular Paper

Two Stage Explicit Runge-Kutta Type Method Using Second and Third Derivatives

Toshinobu Yoshida† and Harumi Ono††

A two stage explicit Runge-Kutta type method for solving non-stiff initial-value problems of autonomous ordinary differential equations is proposed. The method uses first- to third-order derivatives of the solution in the first stage, and second-order pseudo-derivatives in the second stage; which are the product of the Jacobian matrix of the equations and a vector which is the linear combination of the first-order derivatives and all values obtained in the first stage. In these stages, the derivatives and the pseudo-derivatives are assumed to be computed using automatic differentiation. Consequently, these computations can be performed quite easily and efficiently. The order conditions of the method are solved, and the parameters in the method are shown as functions of a free parameter. This is followed by the presentation of the D^{2}RK245 formulas, the fifth-order formula, and the fourth-order formula which is embedded in the fifth-order formula. The leading truncation error terms of these formulas as functions of the free parameter are discussed. Finally, numerical examples are presented to compare the accuracy, CPU time and step control of the proposed method with conventional methods.

1. Introduction

We consider non-stiff initial value problems of autonomous differential equations of the form:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y)\ ,\ y(t_0) = y_0 \qquad (1)$$

where y and f are vectors and f is assumed to be sufficiently smooth.

We propose a two-stage explicit Runge-Kutta type method for solving Eq. (1). The method uses first- to third-order derivatives of the solution in the first stage. The second stage involves the use of the product of the Jacobian matrix $f_y(y)$ and a vector $\tilde{f}$, which is the linear combination of the first-order derivatives $f(y)$ and all values obtained in the first stage. We refer to this product as second-order pseudo-derivatives.

Methods using second-order derivatives have already been proposed by Shintani [13),14)]. He proposed r-stage methods which require one calculation of the first-order derivatives $\dot{y} = f(y)$ and r calculations of the second-order derivatives $\ddot{y} = f_y(y)f(y)$. It has been shown that explicit methods of order $r+2$ exist for $r = 1, 2, 3, 4,$ and 5. Mitsui also proposed $(1, q)$-stage method using q calculations of the second-order derivatives [7)]. Furthermore, Kastlunger and Wanner have proposed a general form of Runge-Kutta type s-stage q-derivative methods [6)].

Toda derived two types of five-stage fifth-order limiting formulas [15)]. In one of them, by taking the limit as the distance between the last two abscissas approaches zero, the form of the pseudo-derivative appears. Ono et al. have proposed explicit two-stage Runge-Kutta type DRK234 formulas [8),9)]. These formulas, which use the second-order derivatives in the first stage and second-order pseudo-derivatives in the sencond stage, achieve fourth-order accuracy in which third-order formula is embedded. They have also shown that three-stage methods using second-order derivatives in the first stage and second-order pseudo-derivatives in the sencond and third stages can not have pairs of formulas one of which is embedded in the other.

Ono has proposed the Runge-Kutta type seventh-order limiting formula, RKD7 [10)]. It is the limiting case where the second and third abscissas approach the first one, and last two abscissas approach each other. In these limits, second- and third-order derivatives appear in the first stage, and the pseudo-derivatives in the last stage. There are no other methods using the second-order pseudo-derivatives that we are aware of.

The proposed two-stage method can achieve fifth-order accuracy in which fourth-order formula is embedded. Therefore, we name the method D^{2}RK245.

This method assumes that the derivatives

† The University of Electro-Communications
†† Current address: 3-22-11, Hachimanyama, Setagaya, Tokyo 156-0056

and the pseudo-derivatives are computed using automatic differentiation [5),12),17)], which can produce the exact, efficient and compact codes for these derivatives.

In the following section we introduce general formulas which illustrate the method, and solve the order conditions. Then we show that these formulas are fifth-order formula with embedded fourth-order formula. Section three illustrates methods for the computation of the derivatives. Section four presents numerical examples to compare the accuracy, the CPU time and step control of the proposed formulas with those of conventional formulas.

2. Two-stage Formulas Using Second and Third Derivatives

In this section we present general formulas for the proposed method, and then derive the order conditions and show that the formulas can not be sixth-order. Next, we investigate the leading truncation error terms of these formulas, and show that there is a free parameter in the fifth-order formula, and a free parameter in the fourth-order formula. This is followed by a discussion of the values of the parameters for minimizing the truncation errors. Finally, we introduce formulas using actual values of these parameters.

2.1 Formulas

We consider the following formulas:

$$
\begin{aligned}
y_n &= y(t_n) \\
f_1 &= f(y_n) = \dot y(t_n), \\
\dot f_1 &= \left.\frac{\mathrm{d}f}{\mathrm{d}t}\right|_{y=y_n} = f_y(y_n) f_1 = \ddot y(t_n), \\
\ddot f_1 &= \left.\frac{\mathrm{d}^2 f}{\mathrm{d}t^2}\right|_{y=y_n} = \dddot y\,(t_n), \\
y_2 &= y_n + h a_{21} f_1 \\
&\quad + h^2 \bar a_{21} \dot f_1 + h^3 \bar{\bar a}_{21} \ddot f_1, \\
f_2 &= f(y_2), \qquad\qquad (2) \\
\tilde f_2 &= f_2 + \alpha_{21} f_1 + h \bar\alpha_{21} \dot f_1 + h^2 \bar{\bar\alpha}_{21} \ddot f_1, \\
\dot{\tilde f}_2 &= f_y(y_2) \tilde f_2, \\
\hat y_{n+1} &= y_n + h(\hat b_1 f_1 + \hat b_2 f_2) \\
&\quad + h^2(\hat{\bar b}_1 \dot f_1 + \hat{\bar b}_2 \dot{\tilde f}_2) + h^3 \hat{\bar{\bar b}}_1 \ddot f_1, \\
y_{n+1} &= y_n + h(b_1 f_1 + b_2 f_2) \\
&\quad + h^2(\bar b_1 \dot f_1 + \bar b_2 \dot{\tilde f}_2) + h^3 \bar{\bar b}_1 \ddot f_1, \\
E &= y_{n+1} - \hat y_{n+1}
\end{aligned}
$$

where $\hat y_{n+1}$ is assumed to be a lower order formula embedded in y_{n+1}. E is an estimation of the truncation error of $\hat y_{n+1}$. It should be noted that $\dot f_1$ and $\ddot f_1$ are the second- and third-order derivative in the first-stage, and $\dot{\tilde f}_2$ is the second-order pseudo-derivative in the second-stage.

2.2 Order Conditions

We expand y_{n+1} around $t = t_n$ up to the h^6-th term, and compare it with the Taylor expansion of $y(t_n+h)$. Although there are twenty sixth-order elementary derivatives [1)] in the Taylor expansion of $y(t_n+h)$, the coefficients of four of these twenty terms are zero in the expansion of y_{n+1}. Therefore the formulas (2) can not be sixth-order formulas.

In order for the solution y_{n+1} to be of the fifth-order, the corresponding terms must be equal up to the h^5-th term, as follows:

$$
\begin{aligned}
&h^1 \mathbf{f}: \quad b_1 + b_2 = 1 \\
&h^2 \mathbf{f}_j \mathbf{f}^j: \quad \bar b_1 + b_2 a_{21} + \bar b_2(\alpha_{21}+1) = \frac{1}{2} \\
&h^3 \mathbf{f}_{jk} \mathbf{f}^j \mathbf{f}^k: \quad \bar{\bar b}_1 + \frac{1}{2} b_2 a_{21}^2 \\
&\qquad + \bar b_2 a_{21}(\alpha_{21}+1) = \frac{1}{6} \\
&h^3 \mathbf{f}_j \mathbf{f}^j_k \mathbf{f}^k: \quad \bar{\bar b}_1 + b_2 \bar a_{21} \\
&\qquad + \bar b_2(\bar\alpha_{21} + a_{21}) = \frac{1}{6} \\
&h^4 \mathbf{f}_{jkl} \mathbf{f}^j \mathbf{f}^k \mathbf{f}^l: \quad \frac{1}{6} b_2 a_{21}^3 \\
&\qquad + \frac{1}{2} \bar b_2 a_{21}^2 (\alpha_{21}+1) = \frac{1}{24} \\
&h^4 \mathbf{f}_{jk} \mathbf{f}^j_l \mathbf{f}^l \mathbf{f}^k: \quad b_2 a_{21} \bar a_{21} \\
&\qquad + \bar b_2(a_{21}(\bar\alpha_{21} + a_{21}) \\
&\qquad + \bar a_{21}(\alpha_{21}+1)) = \frac{1}{8} \\
&h^4 \mathbf{f}_j \mathbf{f}^j_{kl} \mathbf{f}^k \mathbf{f}^l: \quad b_2 \bar{\bar a}_{21} \\
&\qquad + \bar b_2 \left(\bar{\bar\alpha}_{21} + \frac{1}{2} a_{21}^2\right) = \frac{1}{24} \\
&h^4 \mathbf{f}_j \mathbf{f}^j_k \mathbf{f}^k_l \mathbf{f}^l: \quad b_2 \bar{\bar a}_{21} \\
&\qquad + \bar b_2(\bar{\bar\alpha}_{21} + \bar a_{21}) = \frac{1}{24} \\
&h^5 \mathbf{f}_{jklm} \mathbf{f}^j \mathbf{f}^k \mathbf{f}^l \mathbf{f}^m: \quad \frac{1}{24} b_2 a_{21}^4 \\
&\qquad + \frac{1}{6} \bar b_2 a_{21}^3 (\alpha_{21}+1) = \frac{1}{120} \\
&h^5 \mathbf{f}_{jkl} \mathbf{f}^j_m \mathbf{f}^m \mathbf{f}^k \mathbf{f}^l: \quad \frac{1}{2} b_2 a_{21}^2 \bar a_{21} \qquad\qquad (3) \\
&\qquad + \frac{1}{2} \bar b_2 a_{21}(a_{21}(\bar\alpha_{21} + a_{21}) \\
&\qquad + 2\bar a_{21}(\alpha_{21}+1)) = \frac{1}{20}
\end{aligned}
$$

$$
\begin{aligned}
&h^5\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l\mathbf{f}^k_m\mathbf{f}^m : \quad \frac{1}{2}b_2\bar{a}_{21}^2 \\
&\qquad +\bar{b}_2\bar{a}_{21}(\bar{\alpha}_{21}+a_{21}) = \frac{1}{40} \\
&h^5\mathbf{f}_{jk}\mathbf{f}^j_{lm}\mathbf{f}^l\mathbf{f}^m\mathbf{f}^k : \quad b_2a_{21}\bar{\bar{a}}_{21} \\
&\qquad +\bar{b}_2\left(a_{21}\left(\bar{\bar{\alpha}}_{21}+\frac{1}{2}a_{21}^2\right)\right. \\
&\qquad \left. +\bar{\bar{a}}_{21}(\alpha_{21}+1)\right) = \frac{1}{30} \\
&h^5\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l_m\mathbf{f}^m\mathbf{f}^k : \quad b_2a_{21}\bar{\bar{a}}_{21} \\
&\qquad +\bar{b}_2(a_{21}(\bar{\bar{\alpha}}_{21}+\bar{a}_{21}) \\
&\qquad +\bar{\bar{a}}_{21}(\alpha_{21}+1)) = \frac{1}{30} \\
&h^5\mathbf{f}_j\mathbf{f}^j_{klm}\mathbf{f}^k\mathbf{f}^l\mathbf{f}^m : \quad \frac{1}{6}\bar{b}_2a_{21}^3 = \frac{1}{120} \\
&h^5\mathbf{f}_j\mathbf{f}^j_{kl}\mathbf{f}^k_m\mathbf{f}^m\mathbf{f}^l : \quad \bar{b}_2a_{21}\bar{a}_{21} = \frac{1}{40} \\
&h^5\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_{lm}\mathbf{f}^l\mathbf{f}^m : \quad \bar{b}_2\bar{\bar{a}}_{21} = \frac{1}{120} \\
&h^5\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_l\mathbf{f}^l_m\mathbf{f}^m : \quad \bar{b}_2\bar{\bar{a}}_{21} = \frac{1}{120}.
\end{aligned}
$$

where $\mathbf{f}^i_{j_1j_2\cdots j_n}$ denotes an nth-order elementary differential [1)] of the ith component of f, and the summation convention is used. This non-linear system of equations can be solved using a free parameter c_2 ($c_2 \neq 0$) as follows:

$$
\begin{aligned}
&a_{21} = c_2, \quad \bar{a}_{21} = \frac{c_2^2}{2}, \quad \bar{\bar{a}}_{21} = \frac{c_2^3}{6}, \\
&\alpha_{21} = (3-5c_2), \ \bar{\alpha}_{21} = c_2(3-5c_2), \qquad (4) \\
&\bar{\bar{\alpha}}_{21} = \frac{c_2^2(3-5c_2)}{2},
\end{aligned}
$$

$$
\begin{aligned}
b_1 &= \frac{5c_2^4-5c_2+3}{5c_2^4}, \\
\bar{b}_1 &= \frac{10c_2^3-15c_2+8}{20c_2^3}, \qquad (5) \\
\bar{\bar{b}}_1 &= \frac{10c_2^2-15c_2+6}{60c_2^2}, \\
b_2 &= \frac{5c_2-3}{5c_2^4}, \quad \bar{b}_2 = \frac{1}{20c_2^3}.
\end{aligned}
$$

In the case of $c_2 = 3/5$, $\alpha_{21}, \bar{\alpha}_{21}$, and $\bar{\bar{\alpha}}_{21}$ are all zero, and $\tilde{\tilde{f}}_2$ becomes an ordinary derivative:

$$\tilde{\tilde{f}}_2 = f_y(y_2)\tilde{f}_2 = f_y(y_2)f_2. \qquad (6)$$

The order conditions for the embedded fourth-order solution $\hat{y}_{n+1}$ are obtained by making the corresponding coefficients of Taylor expansion of $\hat{y}_{n+1}$ and $y(t_n+h)$ equal up to the h^4-th term, as follows:

$$
\begin{aligned}
&h^1\mathbf{f} : \quad \hat{b}_1 + \hat{b}_2 = 1, \\
&h^2\mathbf{f}_j\mathbf{f}^j : \quad \hat{\bar{b}}_1 + c_2\hat{b}_2 + (4-5c_2)\hat{\bar{b}}_2 = \frac{1}{2}, \\
&h^3\mathbf{f}_{jk}\mathbf{f}^j\mathbf{f}^k, h^3\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k : \qquad (7) \\
&\qquad \hat{\bar{\bar{b}}}_1 + \frac{1}{2}c_2^2\hat{b}_2 + c_2(4-5c_2)\hat{\bar{b}}_2 = \frac{1}{6}, \\
&h^4\mathbf{f}_{jkl}\mathbf{f}^j\mathbf{f}^k\mathbf{f}^l, h^4\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l\mathbf{f}^k, \\
&h^4\mathbf{f}_j\mathbf{f}^j_{kl}\mathbf{f}^k\mathbf{f}^l, h^4\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_l\mathbf{f}^l : \\
&\qquad \frac{1}{6}c_2^3\hat{b}_2 + \frac{1}{2}c_2^2(4-5c_2)\hat{\bar{b}}_2 = \frac{1}{24}
\end{aligned}
$$

where we use the parameters a_{21}, $\bar{a}_{21}$, $\bar{\bar{a}}_{21}$, α_{21}, $\bar{\alpha}_{21}$, and $\bar{\bar{\alpha}}_{21}$ given in Eq. (4).

This non-linear system of equations can be solved using the parameter $\hat{\bar{b}}_2$ as follows:

$$
\begin{aligned}
\hat{b}_1 &= \frac{4c_2^3-1}{4c_2^3} + 3\frac{4-5c_2}{c_2}\hat{\bar{b}}_2 \\
\hat{b}_2 &= \frac{1}{4c_2^3} - 3\frac{4-5c_2}{c_2}\hat{\bar{b}}_2 \qquad (8) \\
\hat{\bar{b}}_1 &= \frac{2c_2^2-1}{4c_2^2} + 2(4-5c_2)\hat{\bar{b}}_2 \\
\hat{\bar{\bar{b}}}_1 &= \frac{4c_2-3}{24c_2} + \frac{1}{2}c_2(4-5c_2)\hat{\bar{b}}_2
\end{aligned}
$$

where c_2 is the same parameter as in Eq. (4).

If $\hat{\bar{b}}_2 = \bar{b}_2$, then $\hat{b}_1 = b_1$, $\hat{\bar{b}}_1 = \bar{b}_1$, $\hat{\bar{\bar{b}}}_1 = \bar{\bar{b}}_1$, and $\hat{b}_2 = b_2$. In this case $\hat{y}_{n+1}$ is the same solution as the fifth-order solution y_{n+1}. Thus, for $\hat{\bar{b}}_2 \neq \bar{b}_2$ we obtain the fourth-order formula embedded in the fifth-order formula.

We note that if $c_2 = 4/5$ and $\hat{\bar{b}}_2 \neq \bar{b}_2$, then $\hat{b}_1 = b_1$, $\hat{\bar{b}}_1 = \bar{b}_1$, $\hat{\bar{\bar{b}}}_1 = \bar{\bar{b}}_1$, $\hat{b}_2 = b_2$, and $E = h^2(\bar{b}_2 - \hat{\bar{b}}_2)\tilde{\tilde{f}}_2$. Hence, the estimation E of the truncation error of $\hat{y}_{n+1}$ becomes very simple.

2.3 Truncation Errors

This section examines the leading truncation error terms of y_{n+1} and $\hat{y}_{n+1}$. We define the truncation error of a term in the expansion of y_{n+1} as the difference between the coefficient of the term and the coefficient of the corresponding term in the Taylor expansion of $y(t_n+h)$. We define a relative error of the term as the ratio of the truncation error to the coefficient of the Taylor expansion. The twenty h^6 terms in the Taylor expansion are divided into four groups. The first group consists of the terms $\mathbf{f}_{jklmn}\mathbf{f}^j\mathbf{f}^k\mathbf{f}^l\mathbf{f}^m\mathbf{f}^n$, $\mathbf{f}_{jklm}\mathbf{f}^j_n\mathbf{f}^n\mathbf{f}^k\mathbf{f}^l\mathbf{f}^m$, $\mathbf{f}_{jkl}\mathbf{f}^j_m\mathbf{f}^m\mathbf{f}^k_n\mathbf{f}^n\mathbf{f}^l$, $\mathbf{f}_{jkl}\mathbf{f}^j_{mn}\mathbf{f}^m\mathbf{f}^n\mathbf{f}^k\mathbf{f}^l$, $\mathbf{f}_{jk}\mathbf{f}^j_{lm}\mathbf{f}^l\mathbf{f}^m\mathbf{f}^k_n\mathbf{f}^n$, $\mathbf{f}_{jkl}\mathbf{f}^j_m\mathbf{f}^m_n\mathbf{f}^n\mathbf{f}^k\mathbf{f}^l$, and $\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l_m\mathbf{f}^m\mathbf{f}^k_n\mathbf{f}^n$, which have a relative error of $|-10+24c_2-15c_2^2|/10$. The second group consists of

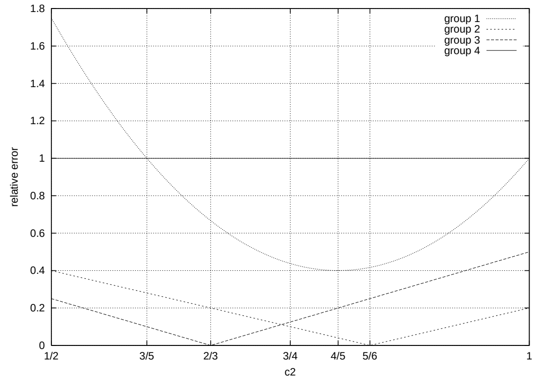

Fig. 1 Relative errors of h^6 terms.

the terms $\mathbf{f}_{jk}\mathbf{f}^j_{lmn}\mathbf{f}^l\mathbf{f}^m\mathbf{f}^n\mathbf{f}^k$, $\mathbf{f}_{jk}\mathbf{f}^j_{lm}\mathbf{f}^l_n\mathbf{f}^n\mathbf{f}^m\mathbf{f}^k$, $\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l_{mn}\mathbf{f}^m\mathbf{f}^n\mathbf{f}^k$, and $\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l_m\mathbf{f}^m_n\mathbf{f}^n\mathbf{f}^k$, which have a relative error of $|6c_2 - 5|/5$. The third group consists of the terms $\mathbf{f}_j\mathbf{f}^j_{klmn}\mathbf{f}^k\mathbf{f}^l\mathbf{f}^m\mathbf{f}^n$, $\mathbf{f}_j\mathbf{f}^j_{klm}\mathbf{f}^k_n\mathbf{f}^n\mathbf{f}^l\mathbf{f}^m$, $\mathbf{f}_j\mathbf{f}^j_{kl}\mathbf{f}^k_m\mathbf{f}^m\mathbf{f}^l_n\mathbf{f}^n$, $\mathbf{f}_j\mathbf{f}^j_{kl}\mathbf{f}^k_{mn}\mathbf{f}^m\mathbf{f}^n\mathbf{f}^l$, and $\mathbf{f}_j\mathbf{f}^j_{kl}\mathbf{f}^k_m\mathbf{f}^m_n\mathbf{f}^n\mathbf{f}^l$, which have a relative error of $|3c_2 - 2|/2$. The last group consists of the terms $\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_{lmn}\mathbf{f}^l\mathbf{f}^m\mathbf{f}^n$, $\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_{lm}\mathbf{f}^l_n\mathbf{f}^n\mathbf{f}^m$, $\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_l\mathbf{f}^l_{mn}\mathbf{f}^m\mathbf{f}^n$, and $\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_l\mathbf{f}^l_m\mathbf{f}^m_n\mathbf{f}^n$, which do not appear in the expansion of y_{n+1}. Therefore, the relative error of these terms is 1. The relative errors as a function of c_2 of the four groups are shown in **Fig. 1**.

From Fig. 1, we see that each curves decreases monotonously for $c_2 < 2/3$, and increases monotonously for $5/6 < c_2$.

Next, we examine the relative error of the forth-order solution $\hat{y}_{n+1}$. The nine h^5 terms of the Taylor expansion are divided into two groups. The first group consists of the terms $\mathbf{f}_{jklm}\mathbf{f}^j\mathbf{f}^k\mathbf{f}^l\mathbf{f}^m$, $\mathbf{f}_{jkl}\mathbf{f}^j_m\mathbf{f}^m\mathbf{f}^k\mathbf{f}^l$, $\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l\mathbf{f}^k_m\mathbf{f}^m$, $\mathbf{f}_{jk}\mathbf{f}^j_{lm}\mathbf{f}^l\mathbf{f}^m\mathbf{f}^k$, and $\mathbf{f}_{jk}\mathbf{f}^j_l\mathbf{f}^l_m\mathbf{f}^m\mathbf{f}^k$, which have a relative error of $(|4 - 5c_2|/4)|\hat{\bar{b}}_2 - \bar{b}_2|$. The second group consists of the terms $\mathbf{f}_j\mathbf{f}^j_{klm}\mathbf{f}^k\mathbf{f}^l\mathbf{f}^m$, $\mathbf{f}_j\mathbf{f}^j_{kl}\mathbf{f}^k_m\mathbf{f}^m\mathbf{f}^l$, $\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_{lm}\mathbf{f}^l\mathbf{f}^m$, and $\mathbf{f}_j\mathbf{f}^j_k\mathbf{f}^k_l\mathbf{f}^l_m\mathbf{f}^m$, which have a relative error of $|\hat{\bar{b}}_2 - \bar{b}_2|$.

If $c_2 = 4/5$, then the relative error of the first group is zero. Hence, there is a greater probability that the error estimation E will be zero. Although determining E is very simple, the fourth-order solution $\hat{y}_{n+1}$ for $c_2 = 4/5$ is not adequate for an embedded solution.

2.4 D^{2}RK245 Formulas

From the above discussion of the case when $c_2 = 4/5$, we do not choose $c_2 = 4/5$. Instead, we use $c_2 = 3/4$ in the interval $[2/3, 5/6]$ because it is a simple fraction, and the relative errors are small. In order to reduce the relative errors of $\hat{y}_{n+1}$, we use $\hat{\bar{b}}_2 = 1/9$. We call the resulting formulas D^{2}RK245, and write them as follows:

$$
\begin{aligned}
f_1 &= f(y_n) \\
\dot{f}_1 &= f_y(y_n) f_1, \\
\ddot{f}_1 &= \left.\frac{\mathrm{d}^2 f}{\mathrm{d}t^2}\right|_{y=y_n}, \\
y_2 &= y_n + \frac{3}{4} h f_1 + \frac{9}{32} h^2 \dot{f}_1 + \frac{9}{128} h^3 \ddot{f}_1, \\
f_2 &= f(y_2), \\
\tilde{f}_2 &= f_2 - \frac{3}{4} f_1 - \frac{9}{16} h \dot{f}_1 - \frac{27}{128} h^2 \ddot{f}_1, \\
\dot{\tilde{f}}_2 &= f_y(y_2) \tilde{f}_2, \qquad (9) \\
\hat{y}_{n+1} &= y_n + h\left(\frac{14}{27} f_1 + \frac{13}{27} f_2\right) \\
&\quad + h^2\left(\frac{1}{9}\dot{f}_1 + \frac{1}{9}\dot{\tilde{f}}_2\right) + h^3 \frac{1}{96}\ddot{f}_1, \\
y_{n+1} &= y_n + h\left(\frac{71}{135} f_1 + \frac{64}{135} f_2\right) \\
&\quad + h^2\left(\frac{31}{270}\dot{f}_1 + \frac{16}{135}\dot{\tilde{f}}_2\right) + h^3 \frac{1}{90}\ddot{f}_1, \\
E &= h\frac{1}{135}(f_1 - f_2) \\
&\quad + h^2\left(\frac{1}{270}\dot{f}_1 + \frac{1}{135}\dot{\tilde{f}}_2\right) + h^3 \frac{1}{1440}\ddot{f}_1.
\end{aligned}
$$

3. Computation of Derivatives

The proposed method assumes the use of automatic differentiation for the derivative computations.

In the formulas Eq. (2), $\dot{f}_1$ and $\dot{\tilde{f}}_2$ can be evaluated efficiently by employing the forward method of automatic differentiation [5),12),17)]. The forward method computes the product of the Jacobian matrix $f_y(y)$ and a vector v without computing the Jacobian matrix itself. The number of operations required to compute the product $f_y(y)v$ by this method is at most three times the number of operations required to compute $f(y)$.

The higher derivatives $\ddot{f}_1$ can be evaluated efficiently by using recursive computation of Taylor coefficients [3),12)]. The solution $y(t)$ of Eq. (1), and its derivative $f(y(t))$ can be expanded as follows:

$$y(t_n + h) = \beta_0 + \beta_1\ h + \beta_2\ h^2 + \cdots \qquad (10)$$

$$f(y(t_n + h)) = \gamma_0 + \gamma_1\ h + \gamma_2\ h^2 + \cdots \qquad (11)$$

where $\beta_0, \beta_1, \beta_2, \cdots, \gamma_0, \gamma_1, \gamma_2, \cdots$ are Taylor coefficients defined by

$$\beta_k = \frac{1}{k!} \left.\frac{\mathrm{d}^k y}{\mathrm{d}t^k}\right|_{t=t_n}, \quad \gamma_k = \frac{1}{k!} \left.\frac{\mathrm{d}^k f}{\mathrm{d}t^k}\right|_{t=t_n}. \tag{12}$$

Furthermore, we note that the following relation holds:

$$\beta_k = \frac{1}{k}\gamma_{k-1}, \quad \text{for } k = 1, 2, \cdots. \tag{13}$$

First, we determine the value of γ_0 ($= f(y(t_n))$) using β_0 ($= y(t_n)$). Next, using the relation (13), we set $\beta_1 = \gamma_0$, and compute γ_1 by the recursive method. The number of operations required by this step is at most three times the number of operations required to compute γ_0. Finally, we set $\beta_2 = \gamma_1/2$ and compute γ_2. The number of operations required by this step is at most five times the number of operations required to compute γ_0. Therefore, we can obtain $\ddot{f}_1 = 2\gamma_2$ with, at most, eight times the number of operations required to compute γ_0.

4. Numerical Examples and Conclusions

In this section we solve an ordinary differential equation C5 in DETEST[4)], which represents the motion of five outer planets about the sun, using the D^{2}RK245 formula, Taylor method[12)] and Dormand-Prince's seven-stage fifth-order formula with fourth-order embedded solution, DOPRI5[2)]. First, we compare the CPU time and the accuracy of these methods without step control. Then we solve the equation with step control using embedded formulas.

4.1 CPU Time and Accuracy

We integrate the equation C5 from $t = 0$ to $t = 20$ with step size $h = 2^k(k = 2, 1, 0, -1, -2, \cdots, -10)$. For each k, we compute the accumulated truncation error e which is defined by the root mean square of the errors at $t = 20$. The numerical computations were performed in quadruple precision Fortran on dual processors of alpha21264 (750 MHz) with 2 GB of RAM. The CPU time in seconds and the error e of each methods are listed in **Table 1**.

It can be seen that the D^{2}RK245 method results in a similar degree of accuracy as the Taylor method and the DOPRI5 method, besides the CPU time of our method are less than that of the other two methods.

In general, if the function f contains elementary functions such as square root, exponential, etc., then the conventional methods must evaluate these high-cost functions in every stage. However, the D^{2}RK245 method calculates these functions only twice, and computes the derivatives of the functions using only arithmetical operations. Therefore, the D^{2}RK245 method has a clear advantage in computational cost.

Table 1 CPU time and errors in the numerical solutions of C5.

$\log_2 h$	D^{2}RK245		Taylor		DOPRI5	
	time	$\log_2 \lvert e\rvert$	time	$\log_2 \lvert e\rvert$	time	$\log_2 \lvert e\rvert$
2	0.00	−6.86	0.00	−6.22	0.00	−5.62
1	0.00	−11.77	0.01	−11.55	0.01	−11.68
0	0.01	−16.74	0.01	−16.88	0.02	−17.70
−1	0.02	−21.74	0.02	−22.11	0.03	−23.54
−2	0.04	−26.74	0.04	−27.26	0.06	−29.14
−3	0.07	−31.74	0.09	−32.34	0.12	−34.50
−4	0.14	−36.74	0.19	−37.39	0.24	−39.70
−5	0.29	−41.74	0.36	−42.41	0.48	−44.80
−6	0.57	−46.74	0.73	−47.42	0.96	−49.85
−7	1.15	−51.74	1.44	−52.43	1.91	−54.88
−8	2.28	−56.74	2.92	−57.43	3.82	−59.89
−9	4.57	−61.74	5.79	−62.43	7.63	−64.90
−10	9.14	−66.74	11.55	−67.43	15.26	−69.90

Table 2 Results with step control.

Method	Tolerance	No.of steps	Percent decieved	Relative error
D^{2}RK245	10^{-3}	2	50.0	51.3
	10^{-6}	14	14.3	22.1
	10^{-9}	62	6.5	19.9
Taylor	10^{-3}	6	16.7	2.1
	10^{-6}	24	12.5	0.5
	10^{-9}	100	4.0	0.4
DOPRI5	10^{-3}	4	25.0	107.0
	10^{-6}	15	13.3	22.3
	10^{-9}	62	4.8	6.4

4.2 Step Control Using Embedded Formulas

We integrate the equation C5 from $t = 0$ to $t = 20$ with initial step size $h = 0.01$ and tolerances 10^{-3}, 10^{-6} and 10^{-9}. The step size is controled by the routines which use the difference of the fifth-order solution and forth-order solution based on the routines in Press et al.[11)]. These computations were performed in double precision gcc on a single pentium4 processor (2.2 GHz) with 1 GB of RAM. The number of steps, the percent of steps for which the local error exeeded the tolerance and the relative errors at $t = 20$ of each methods are listed in **Table 2**.

In this table, the relative error is $max_i||(y_i - y_{t,i})/y_{t,i}||_\infty/\tau$, where y is the numerical solution at $t = 20$, y_t is the true solution at $t = 20$ and τ is the tolerance.

In the D^{2}RK245 method, f_1, $\dot{f}_1$ and $\ddot{f}_1$ are evaluated for each step, f_2 and $\dot{f}_2$ are evaluated for each step and moreover for the decieved case where the local error exceeded the tolerance. In the Taylor method, the function f and it's derivatives are evaluated for each step. In the DOPRI5 method, the function f are called seven times for each step and six times for the decieved case.

The D^{2}RK245 method controls the step width as well as the DOPRI5 method.

Acknowledgments Professor Gerhard Wanner informed the authors of the higher-order formulas, for which the authors are deeply grateful to him. They would also like to thank the reviewers who gave them valuable comments. They also thank Professor Mamoru Hoshi for his helpful advice.

References

1) Butcher, J.C.: *The Numerical Analysis of Ordinary Differential Equations*, Wiley, New York (1987).
2) Dormand, J.R. and Prince, P.J.: A family of embedded Runge-Kutta formulae, *J. Computational and Applied Mathematics*, Vol.6, No.1, pp.19–26 (1980).
3) Hairer, E., Nørsett, S.P. and Wanner, G.: *Solving Ordinary Differential Equations I, Non-stiff Problems*, Springer-Verlag, Berlin (1993).
4) Hull, T.E., Enright, W.H., Fellen, B.M. and Sedgwick, A.E.: Comparing Numerical Methods for Ordinary Differential Equations, *SIAM J. Numer. Anal.*, Vol.9, No.4, pp.603–637 (1972).
5) Iri, M.: Simultaneous Computation of Functions, Partial Derivatives and Estimates of Rounding Errors — Complexity and Practicality, *Jpn. J. Appl. Math.*, Vol.1, pp.223–252 (1984).
6) Kastlunger, K.H. and Wanner, G.: Runge Kutta processes with multiple nodes, *Computing*, Vol.9, pp.9–24 (1972).
7) Mitsui, T.: Runge-Kutta Type Integration Formulas Including the Evaluation of the Second Derivative Part I, *Publ. RIMS, Kyoto Univ.*, Vol.18, pp.325–364 (1982).
8) Ono, H., Toda, H. and Iri, M.: Runge-Kutta type two stage imbedded formulas using the second derivatives (in Japanese), *Trans. IPS Jpn.*, Vol.28, pp.807–814 (1987).
9) Ono, H. and Toda, H.: Explicit Runge-Kutta methods using second derivatives, *Annals of Numer. Math.*, Vol.1, pp.171–182 (1994).
10) Ono, H. and Toda, H.: Runge-Kutta Type Seventh-order Limiting Formula, *J. Info. Proc.*, Vol.12, pp.286–298 (1989).
11) Press, W.H., et al.: *Numerical Recipes in C: the art of scientific computing*, Cambridge University Press, Cambridge (1988).
12) Rall, L.B.: *Automatic Differentiation Techniques and Applications*, Springer, Berlin Heidelberg (1981).
13) Shintani, H.: On One-step Methods Utilizing the Second Derivative, *Hiroshima Math. J.*, Vol.1, pp.349–372 (1971).
14) Shintani, H.: On Explicit One-step Methods Utilizing the Second Derivative, *Hiroshima Math. J.*, Vol.2, pp.353–368 (1972).
15) Toda, H.: On the truncation error of a limiting formula of Runge-Kutta methods (in Japanese), *Trans. IPS Jpn.*, Vol.21, pp.285–296 (1980).
16) Verner, J.H.: Families of Imbedded Runge-Kutta Methods, *SIAM J. Numer. Anal.*, Vol.16, pp.857–875 (1979).
17) Yoshida, T.: Automatic Derivative Derivation System (in Japanese), *Trans. IPS Jpn.*, Vol.30, pp.799–806 (1989).

(Received February 22, 2002)
(Accepted October 7, 2002)

Toshinobu Yoshida was born in 1951. He received his B.E. degree from the University of Electro-Communications in 1973, and his M.E. and D.E. degrees from the University of Tokyo in 1975 and 1978 respectively. He had worked in Chiba University as an assistant, and had worked in Gunma University as an associate professor. Since 1992 he had been in the University of Electro-Communications as an associate professor and has been as a professor since 2000. His current research interests are speech language processing, neural networks and numerical analysis. He is a member of IPSJ, IEICE, JSIAM, JNNS, and ASJ.

Harumi Ono was born in 1932. She received her B.S. degree from Ochanomizu University in 1954, and her D.E. degree from the University of Tokyo in 1985. She had been in Chiba University as an associate professor until March 1997. Her main research interest is numerical analysis. She is a member of IPSJ, JSIAM, and JSAS.