

## 合成関数の高速微分法とその導関数を含む Runge-Kutta 系の常微分方程式数値解法公式への応用†

伊 理 正 夫<sup>††</sup> 小 野 令 美<sup>†††</sup> 戸 田 英 雄<sup>††††</sup>

$m$  個の  $n$  変数関数  $x_1 = g_1(t_1, t_2, \dots, t_n)$ ,  $x_2 = g_2(t_1, t_2, \dots, t_n)$ ,  $\dots$ ,  $x_m = g_m(t_1, t_2, \dots, t_n)$  の、すべての  $t_i$  に関する偏導関数を関数計算と同時に求める効率の良い手法をさきに提案したが<sup>1)</sup>, さらに  $x_1, x_2, \dots, x_m$  の関数  $f$  が与えられ、その合成関数を  $F(t_1, t_2, \dots, t_n) = f(g_1, g_2, \dots, g_m)$  と置くとき、合成関数  $F$  の  $t_i$  に関する偏導関数を効率良く計算する手法について述べる。次に Runge-Kutta 系の常微分方程式数値解法公式に偏導関数を利用したものの性質を一般的に論じ、特に新しい2段4次公式を与える。この際、偏導関数を Runge-Kutta 系の極限公式に現れる形で用いることによって打切り誤差の最適化も行っている。将来上記偏導関数を効率良く計算する手法を自動的に実現するシステムが開発されれば、偏導関数まで取り入れた予測子・修正子法とともに常微分方程式の数値解法公式として有力なものとなる。

### 1. ま え が き

関数計算と同時に、すべての独立変数に関する偏導関数を計算する手法をさきに提案したが<sup>1), 2)</sup>, この手法を合成関数の偏導関数の計算法に適用する。

$t$  の関数  $x = g(t)$  と  $x$  の関数  $f(x)$  が与えられているとする。ここで  $t$  は  $n$  次元、 $x$  は  $m$  次元のベクトルである。このとき合成関数  $F(t) = f(x) = f(g(t))$  の第  $i$  成分  $F^{(i)}$  の、 $t$  の第  $j$  成分  $t_j$  に関する偏導関数は

$$\frac{\partial F^{(i)}}{\partial t_j} = \left( \frac{\partial f^{(i)}}{\partial x} \right) \cdot \left( \frac{\partial x}{\partial t_j} \right)$$

である。ここで  $f^{(i)}$  は  $f$  の第  $i$  成分で右辺の“ $\cdot$ ”はベクトルの内積を表す。

ベクトル  $\partial f^{(i)}/\partial x$  のすべての成分を同時に求める算法は、関数計算を単項あるいは2項演算に分解しておいて次の三つの stages に分けられる。stage 1: 計算過程に従って関数計算を行う。stage 2: 前記 stage 1 と並行して個々の演算に現れる一つまたは二つの変数に関する偏導関数を計算する。stage 3: 関数値から変数に至るすべての道を逆順にたどってすべての偏導関数を同時に計算する。

$\partial F^{(i)}/\partial t_j$  を求めるにはベクトル  $\partial f^{(i)}/\partial x$  の個々の成分は不要で前述の stage 2 と stage 3 は一つの stage 2' にまとめられて次のようになる。ここで、ベクトル  $\partial x/\partial t_j$  は何らか（上述の方法など）の手段で求められているとする。

$\partial x/\partial t_j$  の各成分を  $x$  の各成分のところに付随させて置く。stage 2 で中間変数の計算に用いた項に関する偏導関数を求めると同時にそれらの項に付随して置かれている値との積和を作ってその中間変数に付随させて置く (stage 2')。  $f^{(i)}$  に達すればそこに付随させて置かれた値が求める  $\partial F^{(i)}/\partial t_j$  である。

次に常微分方程式にこれを応用する。常微分方程式  $dy/dt = f(t, y)$  はベクトル  $y$  の各成分が1変数  $t$  の関数となっている特別の場合で、さらに  $\partial y/\partial t$  が今計算しようとしている関数値そのものになっているという特別の場合である。したがって stage 2' をそのまま用いることはできない。そこで一般の場合と同じ stage 2 を実行する。次に1と、今得られた関数値  $f$  を  $t$  と  $y$  に付随させて置き、関数を計算した順に道をたどって内積を計算する (stage 3')。

次に、このようにして求められる  $df/dt$  を利用した常微分方程式の数値解法を考える。代表的な解法として1段階法と多段階法が考えられるが、このうち1段階法の代表的解法である Runge-Kutta 系の公式について考察する。

$f$  の1階の偏導関数まで取り入れた Runge-Kutta 系公式は、既に新谷<sup>7), 8)</sup>, 三井<sup>9)</sup> らによって高次のものまで開発されているが、ここで与える公式の特徴は偏導関数を用いる部分の式の形をより一般化したことである。単一方程式の場合について書くと、導関数を

† Fast Differentiation of Composite Functions and Numerical Integration Formulas of the Runge-Kutta Type for Ordinary Differential Equations by MASAO IRI (Department of Mathematical Engineering and Instrumentation Physics, Faculty of Engineering, University of Tokyo), HARUMI ONO (Tokyo Metropolitan Noge Agricultural Upper Secondary School) and HIDEO TODA (Faculty of Engineering, Chiba University).

†† 東京大学工学部計数工学科

††† 東京都立農芸高等学校

†††† 千葉大学工学部

通常の

$$Df = \frac{\partial}{\partial t} f(t, y) + f(t, y) \cdot \frac{\partial}{\partial y} f(t, y)$$

という形ではなく、その段までに既に計算されている関数値と偏導関数の値まで取り入れ、 $i$  段目での値を

$$\overline{Df}_i = \frac{\partial}{\partial t} f(t, y) + (\sum_{k=1}^i \alpha_k f_k + h \sum_{k=1}^i \beta_k \overline{Df}_k) \cdot \frac{\partial}{\partial y} f(t, y)$$

によって計算する。ここで  $f$  の下つきの番号はその番号の段での値を表す。これはさきに発表した Runge-Kutta 系の極限公式<sup>5), 9)</sup> に現れる形のものである。この場合  $\overline{Df}_i$  は合成関数の微分ではないが、stage 3' の  $f$  を  $\overline{Df}_i$  の  $\frac{\partial}{\partial y} f(t, y)$  の係数で置き換えさえすればよい。

2段公式では  $Df$  の形でも4次公式にはなるが、自由パラメタは残らず、5次の誤差項は4段4次の古典的 Runge-Kutta 公式に比べほぼ2倍となる。しかし以下で与える公式のように  $\overline{Df}_2$  の形を採用することによって自由パラメタが残り、これを利用して5次の誤差項を最適化することができて、古典的 Runge-Kutta 公式とほぼ同精度の公式が得られる。

## 2. 合成関数の偏導関数の計算法

$m$  個の  $n$  変数関数

$$x_1 = g_1(t_1, t_2, \dots, t_n), x_2 = g_2(t_1, t_2, \dots, t_n), \dots,$$

$$x_m = g_m(t_1, t_2, \dots, t_n)$$

と  $x_1, x_2, \dots, x_m$  の関数  $f^{(1)}, f^{(2)}, \dots$  が与えられたとき、その合成関数を

$$F^{(1)}(t_1, t_2, \dots, t_n) = f^{(1)}(g_1, g_2, \dots, g_m),$$

$$F^{(2)}(t_1, t_2, \dots, t_n) = f^{(2)}(g_1, g_2, \dots, g_m),$$

...

と置けば、 $F^{(i)}$  の  $t_j$  に関する偏導関数は

$$\frac{\partial F^{(i)}}{\partial t_j} = \frac{\partial f^{(i)}}{\partial x_1} \cdot \frac{\partial x_1}{\partial t_j} + \frac{\partial f^{(i)}}{\partial x_2} \cdot \frac{\partial x_2}{\partial t_j} + \dots + \frac{\partial f^{(i)}}{\partial x_m} \cdot \frac{\partial x_m}{\partial t_j}$$

である。

さきに提案した手法<sup>1), 2)</sup> を用いるかあるいは問題自身の性質から、 $x_1, x_2, \dots, x_m$  の  $t_1, t_2, \dots, t_n$  に関する偏導関数は求められているとする。このとき  $\partial F^{(i)}/\partial t_j$  を求める手続きは次のようなものになる。

関数値を求める計算は、単項あるいは2項演算に分解され計算グラフに書かれているとする。このグラフの各頂点を中間変数と呼ぶ。また各中間変数に対しそ

の中間変数を求めるのに用いるすぐ下の1個(単項演算)あるいは2個(2項演算)の(中間)変数に関する偏導関数を要素的偏導関数と呼ぶ。

$\partial F^{(i)}/\partial t_j$  の計算にはベクトル  $(\partial f^{(i)}/\partial x_1, \partial f^{(i)}/\partial x_2, \dots, \partial f^{(i)}/\partial x_m)$  の個々の成分は要らないので、 $F^{(i)}$  と結ばれている  $x_k$  および中間変数に対して、ベクトル  $(\partial x_1/\partial t_j, \partial x_2/\partial t_j, \dots, \partial x_m/\partial t_j)$  と中間変数  $w$  の各  $x_i$  に関する偏導関数との内積を入れる場所  $D(w)$  を用意する。

$D(x_k) := \partial x_k/\partial t_j$  と置き計算グラフを下から順にたどって各中間変数を計算していくが、このときこれと並行して  $f^{(i)}$  と結ばれている中間変数  $w$  のところでは、 $w$  が2項演算  $w := u \odot v$  ならば  $\partial w/\partial t_j = (\partial w/\partial u) \cdot (\partial u/\partial t_j) + (\partial w/\partial v) \cdot (\partial v/\partial t_j)$  なので  $D(w) := (\partial w/\partial u) \cdot D(u) + (\partial w/\partial v) \cdot D(v)$  を計算する。 $w$  が単項演算  $w := \odot u$  ならば  $\partial w/\partial t_j = (\partial w/\partial u) \cdot (\partial u/\partial t_j)$  なので  $D(w) := (\partial w/\partial u) \cdot D(u)$  を計算する。 $f^{(i)}$  に達すれば  $D(f^{(i)}) := \partial F^{(i)}/\partial t_j$  である。

この手順から明らかのように、 $F$  のすべての成分の一つの  $t_j$  に関する偏導関数は、 $F$  のすべての成分を計算するのに要する総演算回数の高々定数倍の演算回数で求めることができる。特に常微分方程式に応用する場合  $n=1$  であれば  $F$  自身とその導関数をすべて計算するのに要する演算回数は、 $m$  がいくら大きくても、 $F$  の計算だけに要するもの的高々数倍で抑えられる( $m$  によらない)。

### 2.1 算 法

例えば、1変数  $t$  の関数  $x = \varphi(t)$ ,  $y = \psi(t)$  と  $x, y$  の関数  $f(x, y) = (xy + b)x + c$  が与えられ、 $dx/dt$ ,  $dy/dt$  は求められているとする。このとき合成関数  $F(t) = f(\varphi(t), \psi(t))$  の  $t$  に関する導関数を求める手続きは次のようになる。

$f$  を計算する過程を単項または2項演算の形にする。

$$v_1 = x \times y,$$

$$v_2 = v_1 + b,$$

$$v_3 = v_2 \times x,$$

$$f = v_3 + c.$$

初期設定:  $D(x) := dx/dt$ ,  $D(y) := dy/dt$ .

反復過程:

$$v_1 := x * y, \quad D(v_1) := (\partial v_1/\partial x) \cdot D(x) + (\partial v_1/\partial y) \cdot D(y),$$

$$v_2 := v_1 + b, \quad D(v_2) := (\partial v_2/\partial v_1) \cdot D(v_1),$$

$$v_3 := v_2 * x, \quad D(v_3) := (\partial v_3/\partial v_2) \cdot D(v_2) + (\partial v_3/\partial x) \cdot D(x),$$

$$f := v_3 + c, \quad dF/dt \equiv D(f) := (\partial f / \partial v_3) \cdot D(v_3).$$

以上の算法を

$$F^{(i)}(t_1, t_2, \dots, t_n) = f^{(i)}(x_1, x_2, \dots, x_m)$$

$$= f^{(i)}(g_1(t_1, t_2, \dots, t_n),$$

$$g_2(t_1, t_2, \dots, t_n), \dots,$$

$$g_m(t_1, t_2, \dots, t_n))$$

の場合について、 $f^{(i)}$  を計算する過程を単項または2項演算の形にした後の部分を述べると次のようになる。

1)  $\partial x_1 / \partial t_j, \partial x_2 / \partial t_j, \dots, \partial x_m / \partial t_j$  が既に求められている場合。

stage 2' :

初期設定 :

$f^{(i)}$  と直接あるいは中間変数を介してつながっているすべての  $x_k$  に対してそれぞれ

$$D(x_k) := \partial x_k / \partial t_j$$

を  $x_k$  に付随させて置く。

反復過程 :

計算過程に従って、 $x$  の側から順に各中間変数  $w$  に対し

$$w := u \odot v \text{ あるいは } w := \odot u$$

の計算を行う。その際  $f^{(i)}$  とつながっている  $w$  に対しては

$$D(w) := (\partial w / \partial u) \cdot D(u) + (\partial w / \partial v) \cdot D(v)$$

あるいは

$$D(w) := (\partial w / \partial u) \cdot D(u)$$

を並行して行い  $w$  に付随させて置く。

$f^{(i)}$  に達すれば  $D(f^{(i)})$  が求めるものである。

2)  $\partial x_1 / \partial t_j, \partial x_2 / \partial t_j, \dots, \partial x_m / \partial t_j$  が  $f^{(i)}$  の計算が終了した後でなければ求められない (常微分方程式などの) 場合。

stage 2 :

反復過程 1 :

計算過程に従って  $x$  の側から各中間変数  $w$  に対し

$$w := u \odot v \text{ あるいは } w := \odot u$$

の計算を行う。その際  $f^{(i)}$  とつながっている  $w$  に対しては

$$\partial w / \partial u \text{ と } \partial w / \partial v \text{ あるいは } \partial w / \partial u$$

を計算し  $w$  に付随させて置く。

stage 3' :

初期設定 :

$f^{(i)}$  とつながっているすべての  $x_k$  に対して

$$D(x_k) := \partial x_k / \partial t_j$$

を付随させて置く。

反復過程 2 :

$f^{(i)}$  とつながっている中間変数  $w$  に対し、計算過程の順に  $x$  の側から、さきに付随させておいた  $\partial w / \partial u, \partial w / \partial v$  を使って2項演算か単項演算かに従い

$$D(w) := (\partial w / \partial u) \cdot D(u) + (\partial w / \partial v) \cdot D(v)$$

あるいは

$$D(w) := (\partial w / \partial u) \cdot D(u)$$

を計算し、 $\partial w / \partial u, \partial w / \partial v$  と別に  $w$  に付随させて置く。

$f^{(i)}$  に達すれば  $D(f^{(i)})$  が求めるものである。

これは stage 2 の反復過程 1 までは一般の偏導関数を求める算法と同じもので stage 3' の初期設定と反復過程 2 だけが異なるものである。

### 3. 常微分方程式数値解法公式への利用

常微分方程式の初期値問題

$$dy/dt = f(t, y), \quad y(t_0) = y_0,$$

( $f, y$  は  $m$  次元ベクトル)

の数値解法公式で、 $f$  の一階の偏導関数まで取り入れた公式を考える。

#### 3.1 Runge-Kutta 系公式

常微分方程式の数値解法の一つの  $s$  段 ( $s$ -stage)

Runge-Kutta 公式は

$$\begin{cases} f_1 = f(t_n, y_n) \\ f_i = f(t_n + \alpha_i h, y_n + h \sum_{j=1}^{i-1} \beta_{ij} f_j), \\ \quad i = 2, 3, \dots, s, \\ y_{n+1} = y_n + h \sum_{i=1}^s \mu_i f_i \end{cases} \quad (3.1)$$

で表される。

前節の手法を用いると、この公式に含まれる関数  $f_i$  の計算を行う際、同時に  $df_i/dt$  が容易にかつ正確に求められるので、それを利用して次の形の公式を考える。

$$\begin{cases} f_1 = f(t_n, y_n), \\ \overline{Df}_1 = \frac{\partial}{\partial t} f(t_n, y_n) + f_{1 \cdot} \frac{\partial}{\partial y} f(t_n, y_n), \\ f_i = f(t_n + \alpha_i h, y_n + h \sum_{j=1}^{i-1} \beta_{ij} f_j \\ \quad + h^2 \sum_{j=1}^{i-1} b_{ij} \overline{Df}_j), \\ \overline{Df}_i = \frac{\partial}{\partial t} f(t_n + \alpha_i h, y_n + h \sum_{j=1}^{i-1} \beta_{ij} f_j \\ \quad + h^2 \sum_{j=1}^{i-1} b_{ij} \overline{Df}_j) \\ \quad + (\sum_{j=1}^{i-1} \gamma_{ij} f_j + h \sum_{j=1}^{i-1} c_{ij} \overline{Df}_j) \\ \quad \times \frac{\partial}{\partial y} f(t_n + \alpha_i h, y_n + h \sum_{j=1}^{i-1} \beta_{ij} f_j \\ \quad + h^2 \sum_{j=1}^{i-1} b_{ij} \overline{Df}_j), \\ \quad i = 2, 3, \dots, s, \end{cases} \quad (3.2)$$

$$| y_{n+1} = y_n + h \sum_{i=1}^s \mu_i f_i + h^2 \sum_{i=1}^s \lambda_i \overline{Df}_i.$$

このパラメタ  $\alpha_i, \beta_{ij}, b_{ij}, \gamma_{ij}, c_{ij}, \mu_i, \lambda_i$  を  $y_{n+1}$  と真の解関数の Taylor 展開が、なるべく  $h$  の高次の項まで、関数によらず一致するように選ぶ。

この公式の特徴は導関数を用いる部分の式を通常の

$$Df = \frac{\partial}{\partial t} f(t, y) + f(t, y) \cdot \frac{\partial}{\partial y} f(t, y) \quad (3.3)$$

という形でなく、それまでの段で得られている  $f_i, \overline{Df}_i$  まで取り入れた形

$$\begin{aligned} \overline{Df}_i = & \frac{\partial}{\partial t} f(t, y) + (\sum_{j=1}^s \gamma_{ij} f_j \\ & + h \sum_{j=1}^s c_{ij} \overline{Df}_j) \cdot \frac{\partial}{\partial y} f(t, y) \end{aligned} \quad (3.4)$$

で用いたことである。これはさきに発表した Runge-Kutta 系の極限公式に現れる形のものである<sup>5), 9)</sup>。

この(3.4)式のパラメタを  $\gamma_{ii}=1, \gamma_{ij}=0 (j \neq i), c_{ij}=0 (j=1, \dots, i-1)$  に選んだものが通常の  $Df$  であり、それに加えて公式(3.2)式の  $\alpha_2=0, \beta_{i1}=\alpha_i, \beta_{ij}=0 (j=1, 2, \dots, i), (i=2, 3, \dots, s), b_{i1}=0 (i=2, \dots, s), \mu_1=1, \mu_i=0 (i=2, \dots, s), \lambda_1=0$  と置いたものが新谷<sup>8)</sup>の陽的公式に相当する。また同じく(3.4)式で  $\gamma_{ii}=1, \gamma_{ij}=0 (j \neq i), c_{ij}=0 (j=1, \dots, i-1)$  に選んだ通常の  $Df$  を用いる公式で、陰的公式まで含めた一般化は三井<sup>3)</sup>によって考察されている。

ここでは陽的 Runge-Kutta 系公式に限定する。

(3.2)式で  $s=1$  と置くと

$$\begin{aligned} f_1 &= f(t_n, y_n) \\ \overline{Df}_1 &= \frac{\partial}{\partial t} f(t_n, y_n) + f_1 \cdot \frac{\partial}{\partial y} f(t_n, y_n) \\ y_{n+1} &= y_n + h \mu_1 f_1 + h^2 \lambda_1 \overline{Df}_1 \end{aligned}$$

となり、 $\overline{Df}_1$  は通常の  $Df_1$  と同じものである。 $\mu_1=1, \mu_2=1/2$  に選べば2次の公式となり、これは2次の Taylor 級数法にほかならない。

### 3.2 2段4次の Runge-Kutta 系公式

(3.2)式で  $s=2$  と置くと公式は次のように書ける：

$$\begin{aligned} f_1 &= f(t_n, y_n), \\ \overline{Df}_1 &= \frac{\partial}{\partial t} f(t_n, y_n) + f_1 \cdot \frac{\partial}{\partial y} f(t_n, y_n), \\ f_2 &= f(t_n + \alpha_2 h, y_n + h \beta_{21} f_1 + h^2 b_{21} \overline{Df}_1), \\ \overline{Df}_2 &= \frac{\partial}{\partial t} f(t_n + \alpha_2 h, y_n + h \beta_{21} f_1 \\ & \quad + h^2 b_{21} \overline{Df}_1) \\ & \quad + (\gamma_1 f_1 + \gamma_2 f_2 + h c_1 \overline{Df}_1) \\ & \quad \times \frac{\partial}{\partial y} f(t_n + \alpha_2 h, y_n + h \beta_{21} f_1 \end{aligned} \quad (3.5)$$

$$\begin{aligned} & + h^2 b_{21} \overline{Df}_1), \\ y_{n+1} &= y_n + h(\mu_1 f_1 + \mu_2 f_2) \\ & + h^2(\lambda_1 \overline{Df}_1 + \lambda_2 \overline{Df}_2) \end{aligned}$$

$y_{n+1}$  と真の解関数の Taylor 展開の同じ導関数の係数の差を  $\delta_{p,j}$  で表すと次のようになる：

$$\begin{aligned} h \text{ の項: } & \delta_{1,1} = \mu_1 + \mu_2 - 1, \\ h^2 \text{ の項: } & \delta_{2,1} = \mu_2 \alpha_2 + \lambda_1 + \lambda_2 - 1/2, \\ h^3 \text{ の項: } & \delta_{3,1} = \mu_2 \alpha_2^2 / 2 + \lambda_2 \alpha_2 - 1/6, \\ & \delta_{3,2} = \mu_2 b_{21} + \lambda_2 (\gamma_2 \alpha_2 + c_1) - 1/6, \\ h^4 \text{ の項: } & \delta_{4,1} = \mu_2 \alpha_2^3 / 6 + \lambda_2 \alpha_2^2 / 2 - 1/24, \\ & \delta_{4,2} = \lambda_2 \gamma_2 \alpha_2^2 / 2 - 1/24, \\ & \delta_{4,3} = \mu_2 \alpha_2 b_{21} + \lambda_2 (b_{21} + \alpha_2 (\gamma_2 \alpha_2 + c_1)) - 1/8, \\ & \delta_{4,4} = \lambda_2 \gamma_2 b_{21} - 1/24, \\ h^5 \text{ の項: } & \delta_{5,1} = \mu_2 \alpha_2^4 / 24 + \lambda_2 \alpha_2^3 / 6 - 1/120, \\ & \delta_{5,2} = \mu_2 \alpha_2^2 b_{21} / 2 + \lambda_2 (\alpha_2 b_{21} + \alpha_2^2 (\gamma_2 \alpha_2 + c_1)) / 2 \\ & \quad - 1/20, \\ & \delta_{5,3} = \lambda_2 \gamma_2 \alpha_2^3 / 2 - 1/30, \\ & \delta_{5,4} = 0 - 1/120, \\ & \delta_{5,5} = \mu_2 b_{21}^2 / 2 + \lambda_2 b_{21} (\gamma_2 \alpha_2 + c_1) - 1/40, \\ & \delta_{5,6}^{(1)} = \lambda_2 \gamma_2 \alpha_2 b_{21} - 1/40, \\ & \delta_{5,6}^{(2)} = \lambda_2 \gamma_2 \alpha_2 b_{21} - 1/30, \\ & \delta_{5,7} = \lambda_2 \gamma_2 \alpha_2^3 / 6 - 1/120, \\ & \delta_{5,8} = 0 - 1/120. \end{aligned}$$

また、これらが関数によらず成り立つために

$$\gamma_1 + \gamma_2 = 1, \quad \beta_{21} = \alpha_2$$

でなければならない。 $\delta_{5,6}^{(1)}$  と  $\delta_{5,6}^{(2)}$  は単一の方程式では加え合わせて一つの項となるものである。

これらの  $\delta_{p,j}$  を  $h$  のなるべく高次の項まで0にしようとする、 $h^4$  の項まで0にできて、パラメタは一つの自由パラメタ  $\alpha_2$  を用いて次のように表される：

$$\begin{aligned} \mu_2 &= (2\alpha_2 - 1) / 2\alpha_2^3, \quad \mu_1 = 1 - \mu_2, \\ \lambda_2 &= (3 - 4\alpha_2) / 12\alpha_2^2, \quad \lambda_1 = 1/2 - \mu_2 \alpha_2 - \lambda_2, \\ \gamma_2 &= 1 / (3 - 4\alpha_2), \quad \gamma_1 = 1 - \gamma_2, \quad \beta_{21} = \alpha_2, \\ b_{21} &= \alpha_2^2 / 2, \\ c_1 &= 2\alpha_2(1 - 2\alpha_2) / (3 - 4\alpha_2). \end{aligned} \quad (3.6)$$

$\overline{Df}_1$  は  $Df_1$  と同じものであるが、 $\overline{Df}_2$  も(3.3)式で考えても4次の公式にはなるが自由パラメタは残らず、5次の誤差項の係数の大きさは  $\sqrt{\sum \delta_{5,i}^2 / 8}$  の尺度でみると、普通用いられている4段4次 Runge-Kutta 公式のほぼ2倍になる。これは自由パラメタ  $\alpha_2$  を  $\alpha_2=1/2$  に選んだ特別な場合に相当している。これに反し、(3.4)式の形の  $\overline{Df}_2$  を用いたこの公式には一つの自由パラメタが残る。そこで次節に述べるように、これを用いて5次の誤差項の最適化を行うこと

ができる。

3.3 5次の誤差項の係数の最適化

(3.5)式のパラメタを  $\delta_{5,j}$  に代入すると次のようになる：

$$\begin{aligned} \delta_{5,1} &= -(10\alpha_2^2 - 15\alpha_2 + 6)/720, \\ \delta_{5,2} &= -(10\alpha_2^2 - 15\alpha_2 + 6)/120, \\ \delta_{5,3} &= (5\alpha_2 - 4)/120, \\ \delta_{5,4} &= -1/120, \\ \delta_{5,5} &= -(10\alpha_2^2 - 15\alpha_2 + 6)/240, \\ \delta_{5,6}^{(1)} &= (5\alpha_2 - 4)/120, \\ \delta_{5,6}^{(2)} &= (5\alpha_2 - 3)/120, \\ \delta_{5,7} &= (5\alpha_2 - 3)/360, \\ \delta_{5,8} &= -1/120. \end{aligned}$$

そこで、これらなるべく小さくするように  $\alpha_2$  を決める。  $|\delta_{5,4}|$  と  $|\delta_{5,8}|$  は定数なので、これを除いて考える。  $|\delta_{5,j}|$  の  $\alpha_2$  による変化の様子を図1に示す。図からも最適な  $\alpha_2$  は 7/10 付近と分かる。絶対値最大のものが最小となるように定めるとすれば、単一の方程式に対しては約  $\alpha_2 = 11/15$ 、連立方程式に対しては  $\alpha_2 = 7/10$  となる。これらの  $\alpha_2$  に対する  $\delta_{5,j}$  と尺度  $\sqrt{\sum \delta_{5,j}^2}$  / 項数 による比較を表1に示す。

表1から単一方程式で最良の  $\alpha_2 = 11/15$  に対するものは、連立方程式に対しても小さいので、自由パラメタ  $\alpha_2$  を  $\alpha_2 = 11/15$  に決める。

3.4 打ち切り誤差最良の Runge-Kutta 系2段4次公式

$O(h^5)$  の誤差項の係数を最適化した、 $f$  の1階の偏導関数を含む2段4次公式は次のもので、[DRK 24]

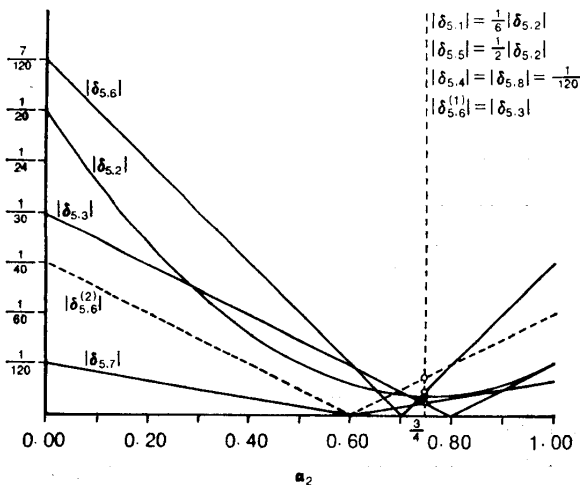


図1  $|\delta_{5,j}|$  の  $\alpha_2$  による変化の様子  
Fig. 1 Graph of  $|\delta_{5,j}(\alpha_2)|$ .

表1  $O(h^5)$  の誤差項の係数  
Table 1 Error coefficients of  $O(h^5)$  terms.

$\delta_{5,j}$	$\alpha_2 = 11/15$	$\alpha_2 = 7/10$	
$\delta_{5,1}$	$-17/32400 \approx -.525_{10} - 3$	$-1/1800 \approx -.556_{10} - 3$	
$\delta_{5,2}$	$-17/5400 \approx -.315_{10} - 2$	$-1/300 \approx -.333_{10} - 2$	
$\delta_{5,3}$	$-1/360 \approx -.278_{10} - 2$	$-1/240 \approx -.417_{10} - 2$	
$\delta_{5,4}$	$-1/120 \approx -.833_{10} - 2$		
$\delta_{5,5}$	$-17/10800 \approx -.157_{10} - 2$	$-1/600 \approx -.167_{10} - 2$	
$\delta_{5,6}$	$\delta_{5,6}^{(1)}$	$-1/360 \approx -.278_{10} - 2$	$-1/240 \approx -.417_{10} - 2$
	$\delta_{5,6}^{(2)}$	$1/180 \approx .556_{10} - 2$	$1/240 \approx .417_{10} - 2$
$\delta_{5,7}$	$1/540 \approx .185_{10} - 2$	$1/720 \approx .139_{10} - 2$	
$\delta_{5,8}$	$-1/120 \approx -.833_{10} - 2$		
$\sqrt{\frac{\sum \delta_{5,j}^2}{8}}$	.00462	.00464	
$\sqrt{\frac{\sum (\delta_{5,j}^{(k)})^2}{9}}$	.00473	.00480	

公式と名づける。

$$f_1 = f(t_n, y_n),$$

$$Df_1 = \frac{\partial}{\partial t} f(t_n, y_n) + f_1 \cdot \frac{\partial}{\partial y} f(t_n, y_n),$$

$$f_2 = f\left(t_n + \frac{11}{15}h, y_n + \frac{11}{15}hf_1 + \frac{121}{450}h^2 Df_1\right),$$

$$D_2 f = \frac{\partial}{\partial t} f\left(t_n + \frac{11}{15}h, y_n + \frac{11}{15}hf_1 + \frac{121}{450}h^2 Df_1\right)$$

$$+ \left(-14f_1 + 15f_2 - \frac{154}{15}h Df_1\right)$$

$$\cdot \frac{\partial}{\partial y} f\left(t_n + \frac{11}{15}h, y_n + \frac{11}{15}hf_1\right)$$

$$+ \frac{121}{450}h^2 Df_1,$$

$$y_{n+1} = y_n + \frac{h}{2662} (1087f_1 + 1575f_2)$$

$$+ \frac{h^2}{484} (27Df_1 + 5Df_2).$$

[DRK 24]

表2に、[DRK 24] 公式と古典的 Runge-Kutta 公式 [RK 4-4] の誤差項の比較を示す。表からも二つの公式はほぼ同精度であることが分かる。

3.5 予測子・修正子法

常微分方程式のもう一つの代表的な解法である予測子・修正子法について言えば、現在ほとんど使われていないが、やはり  $f$  の1階の導関数を取り入れた次の公式(文献6) p. 225)がある。

表 2 打ち切り誤差の主要項の係数の大きさの比較  
Table 2 Comparison of magnitudes of the coefficient of leading truncation error term.

Method	DRK 24	Classical Runge-Kutta RK4-4
$\sqrt{\frac{\sum \delta_{5,j}^2}{8}}$	.00462	.00420
$\sqrt{\frac{\sum (\delta_{5,j}^{(k)})^2}{9}}$	.00473	.00483

予測子:  $y_{n+1}^{(0)} = y_n + h(-f_n + 3f_{n-1})/2 + h^2(17Df_n + 7Df_{n-1})/12,$

改良子:  $\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} + \frac{31}{30}(y_n - y_{n+1}^{(0)}),$

$f_{n+1}^{(0)} = f(t_n + h, \bar{y}_{n+1}^{(0)}),$  (3.7)

修正子:  $y_{n+1}^{(j+1)} = y_n + h(f_{n+1}^{(0)} + f_n)/2 + h^2(-Df_{n+1}^{(j)} + Df_n)/12, (j=0, 1, 2, \dots).$

ここで

$$Df = \frac{\partial}{\partial t} f(t, y) + f(t, y) \cdot \frac{\partial}{\partial y} f(t, y)$$

である。

この公式の修正子は 2 段階的 Runge-Kutta 系公式と見なすこともできる。

$f$  の 1 階の導関数まで計算できる手法を用いれば、この公式は 4 次公式でありながら 1 段階 (1-step) なので、出発および途中の刻み幅変更について有利であり、能率の良い公式で、もっと使われてよい公式と思う。

3.6 偏導関数を含む式の計算法

$m$  元連立常微分方程式を

$$dy_i/dt = f^{(i)}(t, y_1, y_2, \dots, y_m), (i=1, 2, \dots, m)$$

と置く。

[DRK 24] 公式の  $\overline{Df_1}$  と予測子・修正子法 (3.7) 式の  $Df_n, Df_{n-1}, Df_{n+1}$  の第  $i$  成分は共に (3.3) 式の形すなわち

$$Df^{(i)} = \frac{\partial}{\partial t} f^{(i)}(t, y_1, y_2, \dots, y_m) + \sum_{k=1}^m f^{(k)}(t, y_1, y_2, \dots, y_m) \times \frac{\partial}{\partial y_k} f^{(i)}(t, y_1, y_2, \dots, y_m) \quad (3.8)$$

である。ここで  $f^{(k)}, y_k$  はベクトル  $f, y$  の第  $k$  成分である。また [DRK 24] 公式の  $\overline{Df_2}$  は

$$y_{k2} = y_k + \frac{11}{15} h f_1^{(k)} + \frac{121}{450} h^2 \overline{Df_1}^{(k)}$$

と書けば、その第  $i$  成分は

$$\begin{aligned} \overline{Df_2}^{(i)} &= \frac{\partial}{\partial t} f^{(i)}\left(t + \frac{11}{15}h, y_1, y_2, \dots, y_m\right) \\ &+ \sum_{k=1}^m \left(-14f_1^{(k)} + 15f_2^{(k)} - \frac{154}{15}h\overline{Df_1}^{(k)}\right) \\ &\times \frac{\partial}{\partial y_k} f^{(i)}\left(t + \frac{11}{15}h, y_1, y_2, \dots, y_m\right) \end{aligned} \quad (3.9)$$

である。したがってこれらの計算は次のように行う。

2.1 節 2) の算法で  $x_1, x_2, \dots, x_l$  を 1 変数  $t$  の関数  $t, y_1(t), y_2(t), \dots, y_m(t)$  と考える。

したがって  $Df^{(i)}$  を求める (3.8) 式の計算では初期設定で

$$\frac{\partial x_1}{\partial t} := \frac{dx}{dt} := 1, \quad \frac{\partial x_{k+1}}{\partial t} := \frac{dy_k}{dt} := f^{(k)}, (k=1, 2, \dots, m)$$

と置けばよい。(3.9) 式の  $\overline{Df_2}^{(k)}$  は合成関数としての導関数  $\partial F^{(k)}/\partial t$  ではないが、

$$\begin{aligned} \frac{\partial x_1}{\partial t} := \frac{dx}{dt} := 1, \quad \frac{\partial x_{k+1}}{\partial t} := \frac{dx_{k+1}}{dt} := -14f_1^{(k)} \\ + 15f_2^{(k)} - \frac{154}{15}h\overline{Df_1}^{(k)}, (k=1, 2, \dots, m) \end{aligned}$$

と置けば同じ算法で計算できる。

なお、ここではすべての  $i (i=1, \dots, m)$  について  $Df^{(i)}, \overline{Df_2}^{(i)}$  の値が必要なので、すべての中間変数について順に計算すればよい。また  $\overline{Df_2}^{(i)}$  に必要なベクトル  $f_2$  は stage 1 で得られたものなので、直前のベクトル  $f_1$  と  $\overline{Df_1}$  を蓄えておけばよい。

3.7 数値例

例 1) 単一方程式

$$dy/dt = -t^2 y^2/3, y(2) = 1.$$

刻み幅  $h$  を  $2^{-2}$  から  $2^{-16}$  まで変えて、古典的 Runge-Kutta 公式 [RK 4-4] と [DRK 24] 公式で積分したときの  $t=3$  における相対誤差を図 2 に示す。

例 2) 連立方程式

中心力場における質点の 2 次元運動の方程式

$$dx/dt = u, du/dt = -x/(x^2 + y^2)^{3/2},$$

$$dy/dt = v, dv/dt = -y/(x^2 + y^2)^{3/2},$$

$$x(0) = 3, u(0) = 0.3, y(0) = 0, v(0) = 0.2$$

を新しい独立変数  $s$  を導入した形で解く。ここで

$$\begin{aligned} dt/ds = g = 1/(1 + (dx/dt)^2 + (du/dt)^2 \\ + (dy/dt)^2 + (dv/dt)^2)^{1/2} \end{aligned}$$

である<sup>4)</sup>。

これを [DRK 24] 公式と [RK 4-4] 公式で  $s=0$  から 28.125 まで ( $t$  で言えば 0 から約 18.75 まで) 刻み幅  $h$  を  $2^{-3}$  から  $2^{-7}$  まで変えて積分したときの

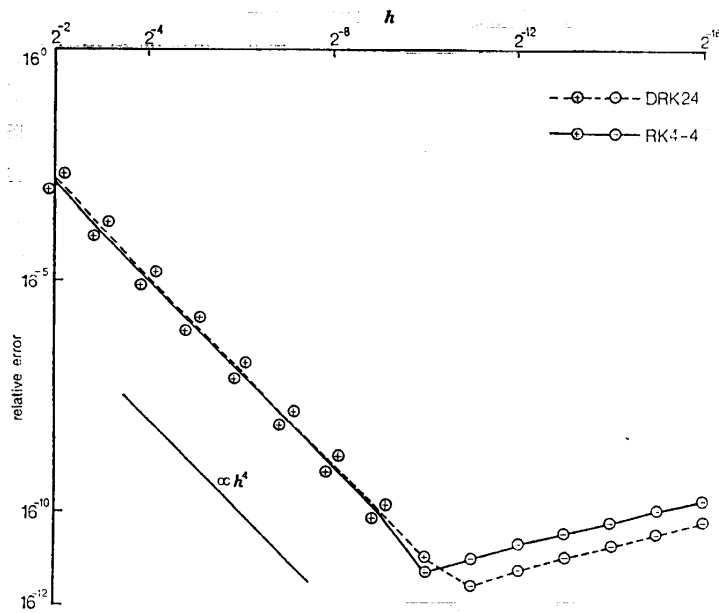


図 2  $y' = -\frac{t^2 y^3}{3}$ ,  $y(2) = 1$  の  $t = 3$  における相対誤差の比較  
 Fig. 2 Comparison of relative error for problem 1) at  $t = 3$ .

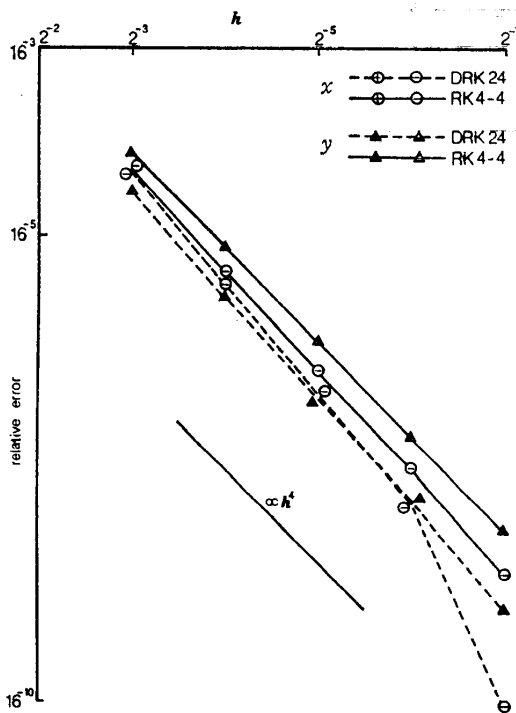


図 3 例 2) の  $s = 28.125$  における  $x$  と  $y$  の相対誤差の比較  
 Fig. 3 Comparison of relative error for problem 2) at  $s = 28.125$ .

最終ステップにおける  $x$  と  $y$  の相対誤差を 図 3 に示す。

図からも誤差が  $h^4$  に比例して小さくなり [DRK 24] 公式は 4 次公式であることが分かる。また局所打ち切り誤差は両公式ともほぼ同じであることが分かる。

計算の時間については、現在ではこの方式で偏導関数計算まで行えるシステムはないので、FORTRAN によるシミュレーションの結果では実測計算時間の比は表 3 のようであった。導関数を用いても関数のみを用いる場合に比べて、次数当たりの計算時間が特に不利にはなっていない点に注目すべきである。将来この方式のシステムが開発されればもっと改善されると思われる。

#### 4. むすび

多変数の合成関数の形で表される 1 変数のベクトル関数の導関数は、さきに提案した偏導関数を求める手法から容易に導かれる。計算の手間も関数計算の手間とほぼ同じである。

またこれを利用した Runge-Kutta 系数値解法公式 [DRK 24] は、導関数を用いる部分をより一般的な形にした結果打ち切り誤差に関して古典的 Runge-Kutta 公式とほぼ同精度が達成されている。

将来この方式による偏導関数計算まで行えるシステム (コンパイラなど) が提供されるようになれば、上記公式は簡単に推奨される公式となろう。またそのようなシステムがない現在でも、“計算グラフ”を人手で書けば、実際の計算に対して十分実用になる公式である。

謝辞 情報処理研修センタ事務局の方々には、計算機の使用について、特段の便宜を図っていただいた。ここに記して深甚なる謝意を表する次第である。

表 3 計算時間の比  
 Table 3 Ratio of C. P. U. times.

Problem	Method	
	Classical Runge-Kutta RK 4-4	DRK 24
Problem 1 65536 steps (single equation)	1	0.94
Problem 2 3375 steps (system of equations)	1	1.14

## 参考文献

- 1) Iri, M.: Simultaneous Computation of Functions, Partial Derivatives and Estimates of Rounding Errors—Complexity and Practicality—, *Jpn. J. Appl. Math.*, Vol. 1, No. 2, pp. 223-252 (1984).
- 2) 岩田憲和, 伊理正夫: 多変数関数の勾配の計算方法, 情報処理学会数値解析研究会資料, 7-1, pp. 1-9 (1983).
- 3) Mitsui, T.: Runge-Kutta Type Integration Formulas Including the Evaluation of the Second Derivative Part I, *Publ. RIMS, Kyoto Univ.*, Vol. 18, No. 1, pp. 325-364 (1982).
- 4) 森口繁一, 伊理正夫, 小林光夫: 常微分方程式の数値解法における難問対策へのある試み, 第8回プログラミング・シンポジウム報告集, 情報処理学会, pp. B 28-33 (1967).
- 5) 小野令美, 戸田英雄: 6個の関数計算による実質的6次のRunge-Kutta法, 情報処理学会論文誌, Vol. 23, No. 6, pp. 599-607 (1982).
- 6) Ralston, A. and Rabinowitz, P.: *A First Course in Numerical Analysis* (Second Edition), p. 556, McGraw-Hill, New York (1978) (戸田英雄, 小野令美(訳): 電子計算機のための数値解析の理論と応用, p. 515, プレイン図書出版, 東京 (1986)).
- 7) Shintani, H.: On One-step Methods Utilizing the Second Derivative, *Hiroshima Math. J.*, Vol. 1, No. 2, pp. 349-372 (1971).
- 8) Shintani, H.: On Explicit One-step Methods Utilizing the Second Derivative, *Hiroshima Math. J.*, Vol. 2, No. 2, pp. 353-368 (1972).
- 9) 戸田英雄: Runge-Kutta系のある極限公式の打ち切り誤差について, 情報処理学会論文誌, Vol. 21, No. 4, pp. 285-296 (1980).

(昭和60年11月6日受付)

(昭和61年1月17日採録)



伊理 正夫 (正会員)

昭和8年生。昭和30年東京大学工学部応用物理学科(数理学科)卒業。昭和35年同大学院博士課程修了。工学博士。九州大学工学部助手, 助教授(通信工学科), 東京大学助教授(工学部計数工学科)を経て, 現在同大教授。回路, グラフ, 数値計算, 言語などの研究, 教育に従事。昭和40年松永賞受賞。著書「Network Flow, Transportation and Scheduling」など。



小野 令美 (正会員)

1932年生。1954年お茶の水女子大学理学部数学科卒業。工学博士。東京都立農芸高等学校に勤務。数値計算に興味をもっている。著書「入門数値計算」(共著, オーム社)。応用統計学会会員。



戸田 英雄 (正会員)

1927年生。1950年東京大学工学部計測工学科卒業。1958年東京大学工学部大学院(旧制)了。工学博士。1958年7月電気試験所物理部応用数学課勤務。1981年6月より千葉大学工学部教授現在に至る。計算数学, 計算統計学に興味をもっている。著書「入門数値計算」(共著, オーム社)など。応用統計学会会員。