

常微分方程式の計算における丸め誤差の改善<sup>†</sup>平野 泰彦<sup>††</sup>

## Abstract

In the numerical integration of ordinary differential equations, the narrower intervals do not always bring out the more precise solution, because of round-off errors.

Runge-Kutta-Gill method is known to improve round-off errors, but it is often used in unsatisfactory manner, therefore producing unsuccessful results. In this paper, necessary conditions for using Runge-Kutta-Gill method correctly are discussed, and experimental results are given. The same conditions are not so effective for predictor-corrector methods as for Runge-Kutta-Gill method. A more adequate method is proposed in this paper. Numerical examples are also given.

## 1. ま え が き

常微分方程式の数値計算において、刻み幅を小さくすればするほど精度があがるというわけではない。あまりに小さくとり過ぎると、丸め誤差の累積により精度は一般にわるくなる。Runge-Kutta-Gill 法は丸め誤差を補償する数値解法として知られている。しかしながら、そのプログラムは数式そのままではなく、丸め誤差補償のアルゴリズムを必要とする。数値計算用サブルーチンを調べても、正しくプログラミングされているものは少ない。補償される理論的根拠に例題による実験結果を加えて、丸め誤差の補償されるアルゴリズムを明らかにする。さらに、予測子修正子法における丸め誤差の対策について述べる。

## 2. 反復加算における丸め誤差

コンピュータにより、 $X=1.00$  に  $H=0.01$  を 100 回反復加算すると、ちょうど 2.00 にならないで、1.999...となることは経験的によく知られている。これはコンピュータが浮動小数点 2 進形式で演算を行なっていることによる。小数を含む実数を 2 進数に変換すると一般に循環小数になり、そのうち 1 ワードの仮数部に相当する上位けたが格納される。X と H の加算にはまずけたあわせを行ない、 $X > H$  なら H の下位けたが切捨てられる。したがって、これを繰返せば、この

切捨てた部分が累積して誤差が増加し、しかも X の値が大きいほどの切捨てられるけた数は大きくなる。コンピュータでは、一般にこの部分を切捨てているが、なかには丸めているものもあり、誤差が平均的に小さくなる。

いま、反復加算

$$X = X + H \quad (1)$$

の代りに乗算を用いて、

$$X = XO + \text{FLOAT}(I) * H \quad (I=1, 2, \dots) \quad (2)$$

とすると、乗算では先のけたあわせによる丸め誤差の累積は生じない。しかも、単精度の浮動小数点乗算では、倍精度演算またはガードディジットにより精度を救済している。

反復加算を倍精度演算すれば、この誤差が小さくなることは明らかである。しかしながら、単精度演算でもこの切捨てた部分を捨て、丸め誤差を補償することができる。そのプログラムは  $X = X + H$  の代りにつぎのようになる。

$$R = H - Q$$

$$W = X$$

$$X = X + R$$

$$R = X - W$$

$$Q = Q + R - H \quad (3)$$

これを式(3)と対照的な図1を用いて説明する。第1式において、最初は  $Q=0$  とし R に H の値を入れる。第2式では加算するまえの X の値をとっておく。第3式で加算を行ない、H の下位けたが切捨てられる。つぎの第4式で H の加算された上位けた部分が求

<sup>†</sup> Improving Round-off errors in Integration of Ordinary differential equations, by Yasuhiko Hirano (Musashino Electric Communication Laboratory N. T. T.)

<sup>††</sup> 武蔵野電気通信研究所応用プログラム研究室

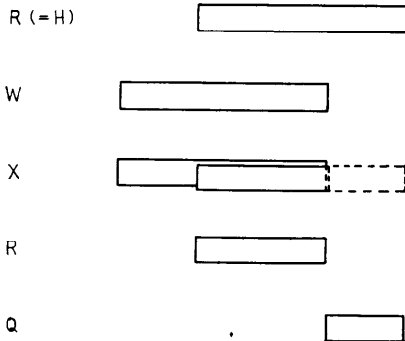


図1 反復加算における丸め誤差の補償手順  
Fig. 1 Procedure of compensating round-off errors in iteration of additions.

まり、さらに第5式のQにHの切捨てられた下位けた部分が逆符号で求められる。これを符号反転して第1式に代入すれば、前回のHの切捨てた部分が次回のHに加算される。それゆえ、このようにして反復加算すれば、丸め誤差が補償されるわけである。

### 3. Runge-Kutta-Gill 法

Runge-Kutta-Gill 法は Gill が固定小数点用に丸め誤差を考慮して開発したもので、浮動小数点用の正しいプログラムとして Thompson<sup>1)</sup>の論文がある。これを引用すると、

```
DO 50 J=1, 4
CALL DERIV
N1=N+1
DO 40 I=1, N1
R=AA(J)*(YD(I)-BB(J)*Q(I))
W=Y(I)
Y(I)=Y(I)+H*R
R=(Y(I)-W)/H
Q(I)=Q(I)+3.*R-CC(J)YD(I)
```

```
40 CONTINUE
50 CONTINUE
```

ここに、係数は  $AA(1)=CC(1)=CC(4)=1, 2$ ,  $AA(2)=CC(2)=(2-\sqrt{2})/2$ ,  $AA(3)=CC(3)=(2+\sqrt{2})/2$ ,  $AA(4)=1/6$ ,  $BB(1)=BB(4)=2$ ,  $BB(2)=BB(3)=1$  である。

これはN元連立微分方程式を刻み幅Hで解くプログラムで、Y(I), I=1, ..., N は従属変数、Y(N+1) は独立変数である。サブルーチン DERIV は Y の導関数 YD(I), I=1, ..., N を計算し、DY(N+1)=1 であ

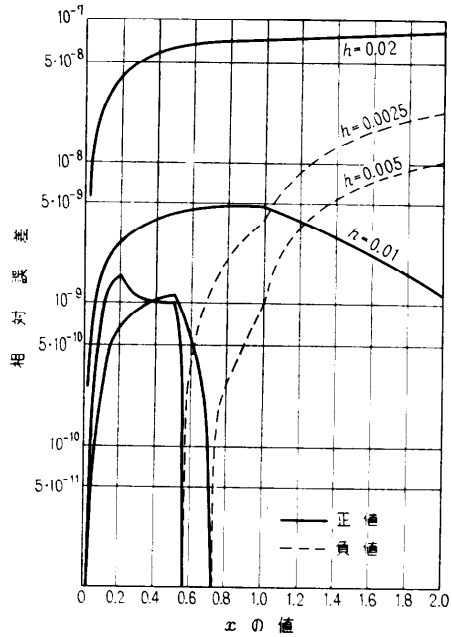


図2 例題1の Runge-Kutta の法による計算結果  
Fig. 2 Computation results of example 1 by Runge-Kutta method.

る。□で囲った部分が必要であることは節2に述べた理由から明らかである。また、独立変数自身をこのプログラムを通すことによりその丸め誤差を補償している。もちろん、節2のアルゴリズムをそのまま組込れてもよい。むしろ、この方が独立変数を明確にして取扱いやすい。これによりXの計算に用いるHと、Yの計算に用いるHの相違による丸め誤差が補償される。要するに、Runge-Kutta-Gill 法は節2に述べたアルゴリズムを巧みに組込んで、丸め誤差を補償しているわけである。

実験例として、丸め誤差の累積のわかりやすい例題1と、やや複雑な問題として例題2をとりあげた。

例題1. つぎの問題は打ち切り誤差の相対値がxの値に関してほぼ一定であるので、刻み幅を小さくすると打ち切り誤差は無視され、丸め誤差の累積を調べるのに都合よい。

$$y' = 3y/(1+x) \quad y(0) = 1. \quad (4)$$

この正解は

$$y = (1+x)^3$$

である。

実験結果として、まず Runge-Kutta の法による計算結果の相対誤差を図2に示す。刻み幅を小さくする

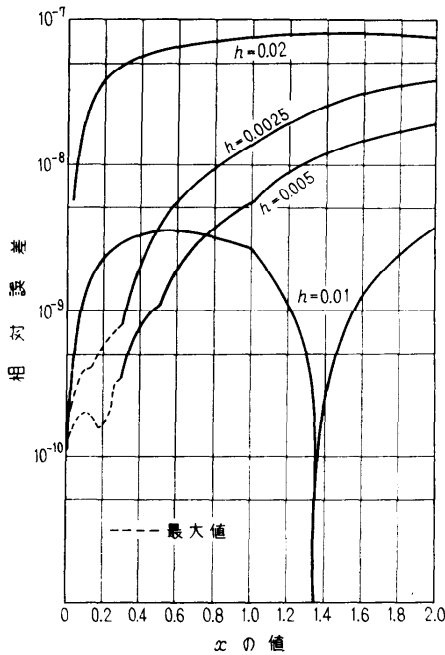


図3 例題1の  $x$  を補償しない Runge-Kutta-Gill 法による計算結果  
 Fig. 3 Computation results of example 1 by Runge-Kutta-Gill method without compensation of independent variable.

と切り誤差は小さくなり、丸め誤差の累積による増加がみられる。図3は  $x$  の丸め補償を行わない Runge-Kutta-Gill 法による計算結果であって、 $x$  が大きくなるとその丸め誤差が大きくなり、これが計算値に影響している。図2および3において、 $x=0.5$  および  $1.0$  における曲線が折れているのは、 $x$  のけたあげによりHのけたが1けた落されるための変化である。図4は  $x$  についても丸め補償を行なった Runge-Kutta-Gill 法による結果であって、丸め誤差の累積による増大はなく、相対誤差は  $5 \times 10^{-10}$  以下におさえられている。

以上により、Runge-Kutta-Gill 法の正しいプログラムでは、独立変数自身についても丸め補償を行なうことが必要である。16進形式演算のマシンでは、丸め誤差の累積は以上とは異なった形状になるが、このときも補償は可能である。これについては参考文献2)を参照されたい。

例題2. 少し複雑な問題としてつぎの2元連立微分方程式を用いた。

$$y_1' = 1 - (y_1 + y_2)/2 - (y_1 - y_2)x/2,$$

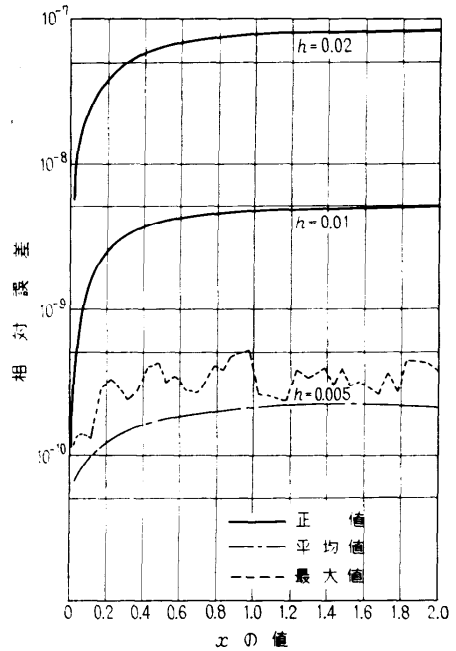


図4 例題1の Runge-Kutta-Gill 法による計算結果  
 Fig. 4 Computation results of example 1 by Runge-Kutta-Gill method.

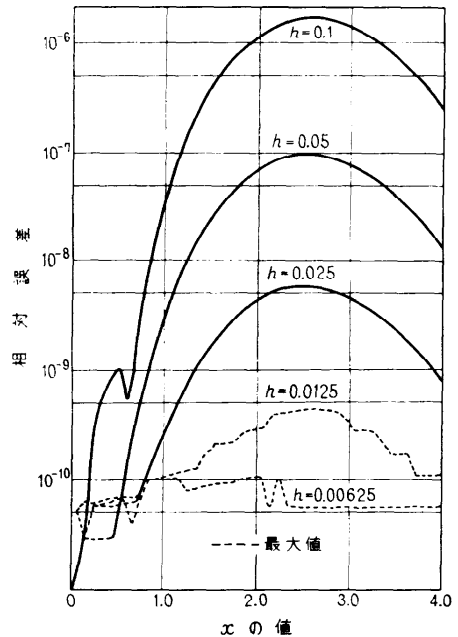


図5 例題2の Runge-Kutta-Gill 法による計算結果  
 Fig. 5 Computation results of example 2 by Runge-Kutta-Gill method.

表 1 各公式の係数表  
Table 1 Coefficients of each formulas

		$a_0$	$a_1$	$a_2$	$a_3$	$b_{-1}$	$b_0$	$b_1$	$b_2$	$b_3$	$e_3$	$N_c$
アダムス法	P	1	0	0	0		55/24	-59/24	37/24	-9/24	251/6	
	C	1	0	0	0	9/24	19/24	-5/24	1/24		-19/6	1
ハーミング法	P	0	0	0	1		8/3	-5/3	8/3	0	112/3	
	C	9/8	0	-1/8	0	3/8	6/8	-3/8	0		-3	1.33
5次ミルン法	P	0	1	0	0		8/3	-5/3	8/3	4/3	116/3	
	C	1/8	7/8	0	0	65/192	243/192	51/192	1/192		-25/16	0.53

$$y_2' = 1 - (y_1 + y_2)/2 + (y_1 - y_2)x/2. \quad (5)$$

$$x=0 \text{ において } y_1=2, y_2=0.$$

この正解は

$$y_1 = 1 + \exp(-x^2/2), y_2 = 1 - \exp(-x^2/2)$$

である。図5に正しい Runge-Kutta-Gill 法による計算結果の  $y_1$  の相対誤差を示す。この打ち切り誤差は  $x$  に関して山形であるが、刻み幅を小さくしても丸め誤差の累積はなく、相対誤差は小さくなっている。図に点線で示す部分は丸めの影響により大きく変動し、最大値を示した。

4. 予測子修正子の誤差の伝ばん

予測子修正子法として、精度の高い5次の公式が広く用いられている。その代表として、Milne 法は振動解を生じやすい不安定性のあることが知られている。5次の修正公式はつぎの一般式から導かれる<sup>3)</sup>。

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} + h(b_{-1} y_{n+1}' + b_0 y_n' + b_1 y_{n-1}' + b_2 y_{n-2}') + e_5 h^5 y^{(5)}/5! \quad (6)$$

$y=1, x, x^2, x^3$ , および  $x^4$  についてこの式は正確であって各係数が求まり、 $y=x^5$  より誤差項が求まる。

$a_1$  と  $a_2$  の値は公式の誤差と安定性を考慮して定められ、その組合せにより各種の公式を得る。予測子修正子法としては、Adams 法と Hamming 法が広く用いられているようである。この2つと、精度がよく連続系シミュレーション CSMP に用いられている5次 Milne 法を検討の対象にした。各公式の係数値を予測子を加えて表1に示す。

各公式の  $a_1$  と  $a_2$  に対する安定性を調べるために、目安として微分方程式  $y' = Ay$  を使い、 $Ah=r$  とおくと、式(6)はつぎの差分方程式で表わされる。

$$(1 - b_{-1}r)y_{n+1} - (a_0 + b_0r)y_n - (a_1 + b_1r)y_{n-1} - (a_2 + b_2r)y_{n-2} = 0 \quad (7)$$

$h \rightarrow 0$  とすると  $r \rightarrow 0$  であって、この特性方程式は次式になる。

$$\rho^3 - a_0 \rho^2 - a_1 \rho - a_2 = 0 \quad (8)$$

この方程式は  $\rho_3=1$  なる根をもち、2次因子はつぎのようになる。ここに、 $a_0=1-a_1-a_2$  である。

$$\rho^2 + (a_1 + a_2)\rho + a_2 = 0 \quad (9)$$

上式の根は寄生根であって、所望の解に対する  $\rho_3=1$  より小さいことが望ましい。

Adams 法では  $\rho_1 = \rho_2 = 0.$

Hamming 法では  $\rho_1 = 0.422, \rho_2 = -0.296.$

5次 Milne 法では  $\rho_1 = 0, \rho_2 = -7/8.$

丸め誤差の伝ばんについては、近似的に導関数の影響を無視して次式により調べる。

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} \quad (10)$$

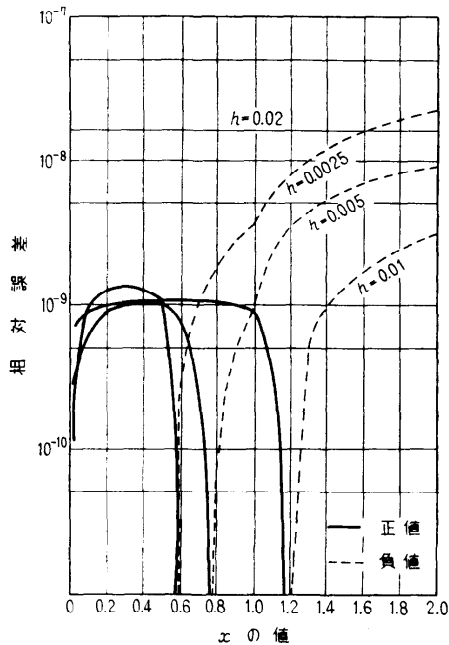


図 6 例題1の丸め補償のない Adams 法による計算結果

Fig. 6 Computation results of example 1 by Adams method without compensation of round-off errors.

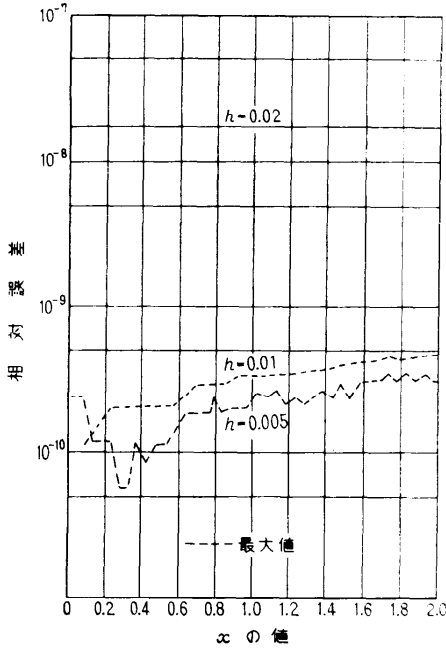


図 7 例題 2 の丸め補償をした Adams 法による計算結果

Fig. 7 Computation results of example 1 by Adams method with compensation of round-off errors.

$\rho_3=1$  であるから、特性方程式の解より

$$y_n = C_1 \rho_1^n + C_2 \rho_2^n + C_3 \quad (11)$$

である。いま、境界条件として  $y_{-2}=y_{-1}=0$ ,  $y_0=\varepsilon$  とすると、

$$\begin{aligned} \frac{C_1}{\rho_1^2} + \frac{C_2}{\rho_2^2} + C_3 &= 0, \\ \frac{C_1}{\rho_1} + \frac{C_2}{\rho_2} + C_3 &= 0, \\ C_1 + C_2 + C_3 &= \varepsilon. \end{aligned} \quad (12)$$

$C_3$  について解くと、

$$C_3 = \frac{\varepsilon}{1 + a_1 + 2a_2}.$$

その公式が  $n \rightarrow \infty$  で安定であるとき、この値は  $\varepsilon$  の長時間演算の誤差伝ばんを表わしている。すなわち、誤差  $\varepsilon$  は次式で増幅され、 $N_c$  は雑音増幅率と呼ばれる。

$$N_c = \frac{1}{1 + a_1 + 2a_2} \quad (14)$$

$|N_c| < 1$  ならば丸め誤差は時間とともに減少する。各公式のこの値が表 1 に示されている。

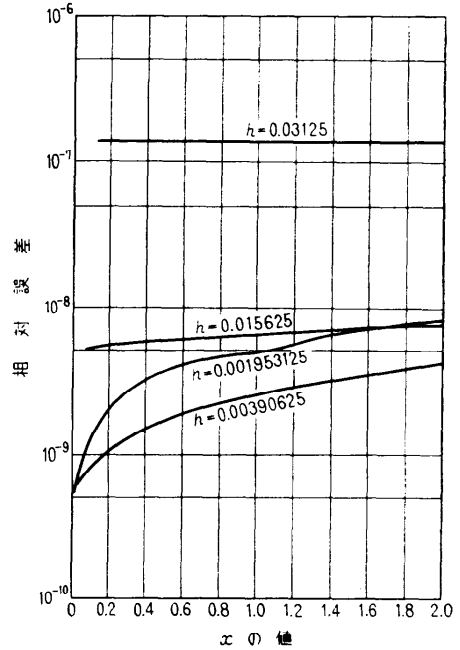


図 8 例題 1 の丸め補償のない Adams 法による計算結果

Fig. 8 Computation results of example 1 by Adams method without compensation of round-off errors on  $h=0.03125, 0.015625, \dots$ , etc.

実際的には、式(12)において  $y_{-2}=0$ ,  $y_{-1}=0$  ではない。  $y_{-2}=\alpha\varepsilon$ ,  $y_{-1}=\beta\varepsilon$  とすると、同様の計算により次式を得る。

$$N_c = \frac{1 + \alpha a_2 + \beta(a_1 + a_2)}{1 + a_1 + 2a_2}. \quad (15)$$

$\alpha=\beta=1$  ならば、 $a_1$  と  $a_2$  に関係なく  $N_c=1$  である。

各公式について、上記の雑音増幅率を節 3 の例題 1 を用いて検証した。出発値の相対誤差は  $\alpha\varepsilon=0.11641 \times 10^{-9}$ ,  $\beta\varepsilon=0.34924 \times 10^{-9}$ ,  $\varepsilon=-0.99993 \times 10^{-5}$  であって、 $y_0$  のみ故意に  $-10^{-5}$  の大きい誤差を与え、刻み幅は打ち切り誤差を無視しうるように小さく選び、 $h=0.0078125$  で計算を続行した。そのとき、解の相対誤差はつぎの値に落ちついた。

Adams 法	$-1.0025 \times 10^{-5}$
Hamming 法	$-1.2775 \times 10^{-5}$
5次 Milne 法	$-0.5404 \times 10^{-5}$

これらの計算値は  $N_c\varepsilon$  であって、表 1 の  $N_c$  の理論値とほぼ一致している。計算に、評価誤差による補

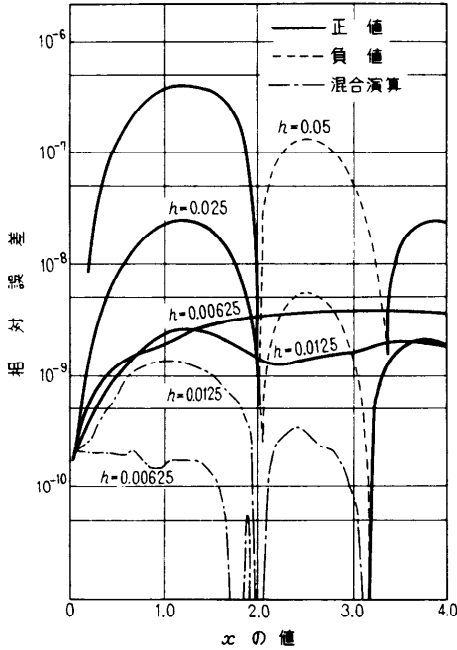


図9 例題2の丸め補償した Adams 法による計算結果と混合演算による計算結果

Fig. 9 Computation results of example 2 by Adams method with compensation of round-off errors, and in mixed mode.

正は行っていない。なお、5次 Milne 法は精度、雑音増幅率とも優れているが、 $\rho_2 = -7/8$  なる根をもち、そのために誤差は1ステップごとの減衰振動をした。

### 5. 丸め誤差改善の実験

前節の丸め誤差の伝ばんの検討結果から、雑音増幅率が常に1である Adams 法では、節2に述べたアルゴリズムをとり入れることにより、丸め誤差の補償ができそうである。例題1による実験結果を以下に述べる。

まず、丸め補償しない Adams 法により計算した結果を図6に示す。刻み幅を0.01以下にとると、丸め誤差と打ち切り誤差が打消しあい、それ以後丸め誤差が累積して相対誤差の増大がみられる。ここに、以下の実験では評価誤差による補正はしていない。独立変数と修正演算について丸め補償を行なうと図7のようになり、丸め誤差が改善されたが、ほんの少し誤差累積の傾向がみられる。なお刻み幅を0.03125, 0.015625...のような2進数で有限小数になる数値に選ぶことは

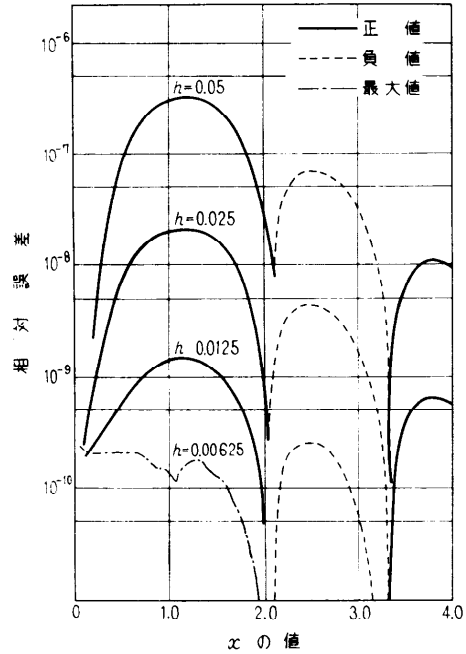


図10 例題2の評価誤差により補正した Adams 法の混合演算による計算結果

Fig. 10 Computation results of example 2 by Adams method in mixed mode.

$x$  の丸め補償をした場合に相当し、その計算結果は図8に示すように多少改善される。

つぎに、少し複雑な例題2を刻み幅を変えて計算した結果を図9に示す。刻み幅を小さくすると誤差は正符号で序々に累積し、改善の度合は Runge-Kutta-Gill 法に較べるとかなり小さい。

一般に、丸め誤差の危険があるときは倍精度演算を行えばよい。しかしながら、導関数の計算に時間を要するときは、 $X$ ,  $H$  および  $Y$  のみを倍精度とする混合演算が考えられる。そのとき導関数の計算は単精度とする。図9の鎖線はこの計算結果であって、かなり改善されている。 $h=0.05, 0.025$  の場合は単精度と変りない。しかも、演算時間とメモリ容量がわずかに増加するだけである。

予測子修正子法の修正反復は精度と演算時間の点から経験的に1~2回がよいといわれている。修正反復を1回にとどめ、評価誤差により予測演算と修正演算の結果を補正したのが、よく知られている Hamming 法である。Adams 法についてもこの形に書くことができる。

$$p_{n+1} = y_n + h(55y_n' - 59y_{n-1}' + 37y_{n-2}' - 9y_{n-3}')/24,$$

$$\begin{aligned} m_{n+1} &= p_{n+1} - 251/270(p_n - c_n), \\ c_{n+1} &= y_n + h(9m_{n+1}' + 19y_n' - 5y_{n-1}' + y_{n-2}')/24, \\ y_{n+1} &= c_{n+1} + 19/270(p_{n+1} - c_{n+1}). \end{aligned} \quad (16)$$

上式より計算すれば、図 10 に示すように誤差がならされて最悪の精度が改善される。

## 7. む す び

Runge-Kutta-Gill 法における丸め誤差改善の根拠は、独立変数と従属変数とについて、加算で切捨てられたけたを拾って補償するからである。予測子修正子法においては、このように補償しても十分効果が現われず、混合演算を行なうのがよいという結論に達した。しかも、評価誤差により補正して精度を向上することができる。常微分方程式の数値計算において、刻み幅の選択は誤差と演算時間を考慮して適当にとらねばならないが、実際問題として計算してみてもわかることが多い。その上、丸め誤差を含むと判断を誤る恐れがある。評価誤差により自動刻み幅制御を行なうと

き、単精度演算では往々にして丸め誤差のために正規に動作しないことがある。混合演算による丸め誤差の対策は Runge-Kutta 法、予測子修正子法ともに適用できる。なお、以上の実験では 1 語 48 ビットのコンピュータを用いたが、ビット数の少ないコンピュータでは丸め誤差の影響は一層大きく現われる。

## 参 考 文 献

- 1) R. J. Thompson: Improving Round-off in Runge-Kutta Computation with Gill's Method, ACM, Vol. 13, No. 12, Dec. 1970. (情報処理 Vol. 12, No. 6 に紹介.)
- 2) 平野泰彦: ルンゲ・クッタ法の丸め誤差の累積とその改善の実験例, 情報処理学会, 46 年度講演予稿集.
- 3) R. W. Hamming: Numerical Methods for Scientists and Engineers, p. 176, McG-Hill, 1962.

(昭和 47 年 6 月 10 日受付)

(昭和 47 年 8 月 1 日再受付)