数値・数式ハイブリッドシステムと常微分方程式†

野田 松太郎†† 岩下 英俊††

微分方程式を計算機で解く問題は厳密な解析解の存在が明らかないくつかの場合に数式処理システムが適用されるが，それ以外の大半は FORTRAN による数値計算によるのみである．数式処理と数値計算の結合によるアルゴリズムの開発と処理システムの作成が期待されているが，いまだ十分に実現されていない．本論ではパソコン上で稼働する小型で移植性に富む数式・数値混合のハイブリッド・システムの作成とその微分方程式求解アルゴリズムへの応用について述べる．システムは PROLOG で記述され，通常の数式を入力とし，出力は見やすい型の数式であったり，FORTRAN プログラムであったり，数値結果であったりする．数式処理と数値計算の結合が容易に成されることが示される．具体的なアルゴリズムとして，数値計算の分野で確立されてきた Taylor 級数法が考察される．現在，この数値計算アルゴリズムでは形式解を得る過程で微分演算を避け複雑な再帰的演算を行っているが，本論の場合は数式微分の採用が容易である．方程式の複雑化と共に数式微分を採用した方法の優位性が示される．さらに，ハイブリッド・システムに導入された Taylor 級数法は代表的な数値解法である Runge-Kutta 法と比較され，入力の容易さと精度面からハイブリッド処理での Taylor 級数法が優れていることが実例を通して述べられる．数式的に得られた形式解はさらに 3 項解析により特異点の位置の探索にも用いられる．

1. 緒 言

最近，数式処理システムへの関心が急速に深まってきており，その利用範囲も単に数学研究・高エネルギー物理学にとどまらず，化学・工学・教育等の分野に広がってきている．しかし，対応するシステムは，REDUCE・MACSYMA 等の利用が比較的容易になりつつあることは事実としても，いまだ十分に多様な利用者の要求を満足してはいない．特に，数値計算を主として行っている計算機利用者から見ると，数式処理システムへの数値計算機能の付加あるいは数値計算プログラムと数式処理の有効な結合等のハイブリッド計算の重要性が強く指摘されている[1]．実際の工学的分野等に出現する大半の微分方程式では厳密な解析解が得られるものは皆無であるといえるため，必然的に数値計算により近似解を得ることになる．計算は FORTRAN 等の科学計算用言語を用いて行われるが，入力の容易さ，出力の見やすさ等が要求される．さらに，数式処理システムのより一層の普及等の面からみてもハイブリッド化あるいはシステムの小型化とその移植性等が要求されている[2),3]．もちろんいろいろな分野の利用者が容易にプログラムし得るような記述言語の使用が望ましいことはいうまでもない．

以上に対し各方面からの接近がなされ上記の REDUCE, MACSYMA 以外に SMP, MAPLE や小型システムを目指した muMATH 等が開発されている．しかし，これらも数値計算との結合に関してはいまだ十分とは言えないように思われる．そこで，我我は特に数値計算との結合を重視した小型数式処理システムを PROLOG を記述言語として開発してきた[1]．このような処理系をパソコン上で稼働させることにより，単に数式処理の普及や数値計算の支援システムに利用するばかりでなく，アルゴリズムの再開発や数学研究，数学教育等上記各分野へのより有効な利用も考えられる．たとえば，各種微分方程式の求解アルゴリズムを考えても，一般に用いられている多くの数値計算アルゴリズムは数値微分の困難を極力避けることに重点を置いている．一方，数式処理の視点からの接近では近似計算などの実用的側面には触れられず，正確に解くことのできる微分方程式（主に常微分方程式）のみが考慮の対象とされている．理想的なハイブリッド処理を用いると，数式微分の遂行と近似計算の可能性の結合により優れた計算法が確立されるであろうことは想像にかたくない．

そこで本論では，常微分方程式に対するハイブリッド解法とそのシステム化について述べる．解法としては伝統的な Taylor 級数法を取り上げ，そのハイブリッド処理の可能性とパソコンへの組み込みについて考える．Taylor 級数法は常微分方程式に対する基本的でかつ有力な解法であるが，数値計算アルゴリズムにするためには数値微分の困難を避けるため計算過程で複雑な記号処理が必要となる．これに対し pre-

† Solving Ordinary Differential Equations on a Hybrid Computation System by MATU-TAROW NODA and HIDETOSHI IWASITA (Department of Electronics Engineering, Ehime University).

†† 愛媛大学工学部電子工学科

compiler の開発が精力的に行われているが[5)-7)]，使用計算機が特定の大型機に限定され移植性に欠けるため，広く用いられてはいない．また，この pre-compiler による結果は数値計算に限られ Taylor 級数を数式の形式で出力できないため，解の振舞いに対する考察には不十分である．この解法をハイブリッド処理システムに導入すると，方程式の入力から級数の計算までを記号的に処理するため，数式のままの出力，その振舞いの解析，あるいは数値結果を得るための FORTRAN プログラムの生成などが自由に行える．ここで構成されたシステムはパソコン上で稼動する小型のもので，かつ移植性に富んでいる．以下記述の一貫性のため，ここで必要な数式処理機能の概略を2章で，Taylor 級数法とその拡張についてを3章で述べる．4章，5章は各々ハイブリッド処理システムの構成といくつかの結果にあてられる．ここでは，数値計算結果との比較もなされ，ハイブリッド処理システム上での Taylor 級数法の良さがみられる．3項解析による特異点の位置の探索に関してもハイブリッド計算が有効になされる．以上の結果が6章にまとめられる．

2. 数式処理の必要性

ハイブリッド処理システムは数式処理部分と数値計算部分とで構成されるが，まず数式処理部分に必要な機能についてまとめる．数式処理システムには**図1**のような段階が必要である．すなわち，通常用いるのと同様な形式での**数式入力**，その内部形式（なんらかのリスト形式）への**変換**，**処理**の段階での様々な記号計算，式の**簡素化**，出力のための**逆変換**あるいは必要な場合の**数値代入**，**出力**等の各段階である．効率的システムのためにはこれら各段階と，対象となるルーチンの作成が必要である．**処理**では多項式演算，数式微分，行列演算，極限操作，Taylor 展開その他諸機能の付加，**簡素化**ではシステム内のデータベースをいかに活用するか等が重要である．**出力段階**では通常の形式に近い数式の2次元出力が望まれる．数値計算部分は豊富な蓄積のある FORTRAN によるのが一般的である．このため，ハイブリッド処理システムでは，**図2**に示すように FORTRAN プログラムの出力やその実行が円滑に遂行されることが要求される．多項式の簡素化を考えても，単なる数式処理の場合と，horner 法的計算が有利な数値計算とでは異なった方式が必要である．出力形式も当然，数値計算部分では

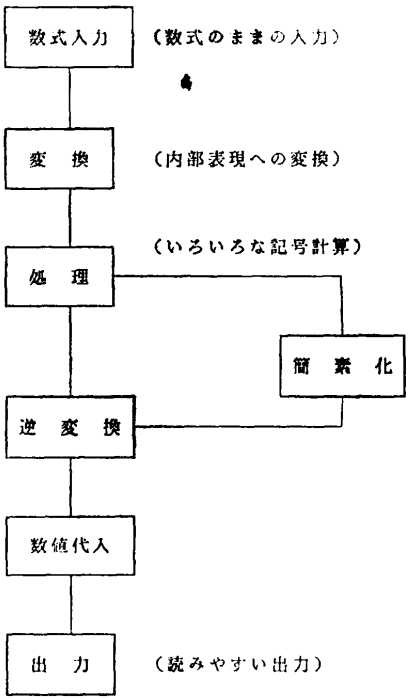

図1 数式処理システム
Fig. 1 The symbolic manipulation system.

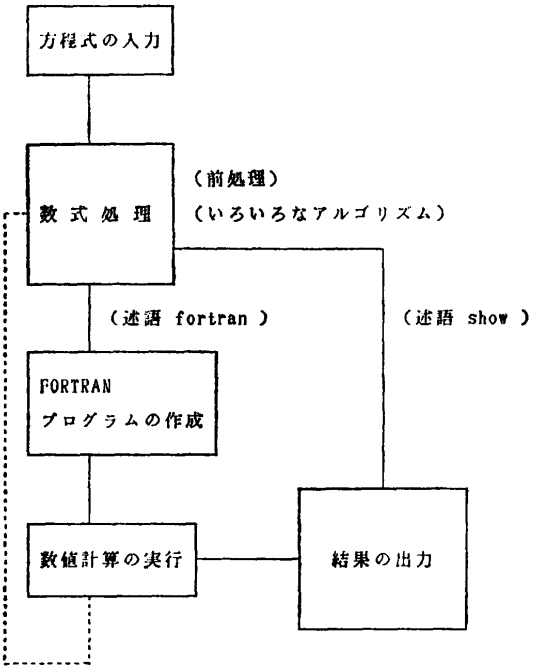

図2 数値・数式ハイブリッド・システムの流れ図
Fig. 2 The flow of hybrid computation.

実行が円滑になされるような FORTRAN プログラムの出力が要求される．本論で述べるシステムでは，これらに対して show（数式の2次元出力），fortran

(FORTRAN プログラムの作成と出力) 等の述語を組み込んでいる.

システムを作成するための記述言語としては多くの場合 LISP が採用されているが, 数式入力, 変換, 簡素化等の過程では主に数式パターンの照合操作が用いられるため PROLOG がむしろ有力であるように思われる. 特に, 数式処理部分の記述言語が LISP あるいは PROLOG であり, 数値計算部分が通常の FORTRAN であるハイブリッド処理システムの場合には, 両者間のデータ形式の相違が大きな障壁になる. これが必要性にもかかわらず有効なハイブリッド処理システムが確立されていない最大の要因である. これに対し, 記述言語そのものをCで作成し, かつCの諸機能を常に用い得るようにすれば, データ形式間の変換も容易になり有効なハイブリッド処理システムを作成することが可能になる[4]. 本論で述べるシステムは, パソコン用 OS の MS-DOS 上で稼働するCにより記述された市販の PROLOG (A. D. A. PROLOG) により記述した. リスト形式のデータと数値データの間の変換は組み込み述語により容易に行える.

3. Taylor 級数法とその拡張

2章でも述べたように, 常微分方程式に対する近似計算の必要性はますます増しており, 数値計算アルゴリズムの研究・開発が大いに進んでいる. これらの方法の原点は関数 (微分方程式の解) の Taylor 級数展開にあるといえる. その概略は以下のとおりである.

今, 与えられた常微分方程式の系

$$dy_i/dx=f_i(x, y_1, \cdots, y_n),\quad y_i(x_0)=y_{i0},\ i=1, \cdots, n \qquad (1)$$

の解 y_i を点 $x=x_0$ に関して Taylor 展開する.

$$y_i(x)=y_i(x_0)+y_i'(x_0)(x-x_0)+\frac{y_i''(x_0)}{2}(x-x_0)^2+\cdots+\frac{y^{(n)}{}_i(x_0)}{n!}(x-x_0)^n+\cdots \qquad (2)$$

初期条件より,

$$y_i(x_0)=y_{i0},\ y_i'(x_0)=f_i(x_0, y_{10}, \cdots, y_{n0}),$$
$$y_i''(x_0)=df_i(x_0, y_{10}, \cdots, y_{n0})/dx, \cdots$$

ここで, $h=x-x_0$ として h に関するべき級数解 (形式解) を得る. 初期値を具体的な数値で与えると容易に x_0 に近い点 x での解を数値的に得ることができる. 第 n 項で打ち切る時の誤差は残余項

$$\frac{y_i^{(n+1)}(\xi)}{(n+1)!}h^{n+1} \text{ ただし } 0<\xi<h$$

より評価することが理論的に可能である. 高階微分方程式の場合は1階連立系に変形する. さらに, 十分な次数までの展開係数が求まれば微分方程式の原点に最も近い特異点の位置 (収束半径) の探索も**3項解析**, **6項解析**等の方法で行うこともできる[6].

この解法は基本的なものであるが汎用的な数値解法にするには1) 高次導関数の計算を数値計算でプログラム化することは困難である, 2) Taylor 級数の収束が速くない, 3) 残余項の計算にはある区間に対する導関数が必要となる, 等の問題点があり, 現在では数値解法としては Runge-Kutta 法等に比し適当なものとはされていない.

これに対し高次導関数を微分演算を用いずに求めることにより Taylor 級数法の数値解法としての汎用化を目指したものに Moore による**再帰的 Taylor 級数法**がある[8]. この方法では, まず**基本関数** (二項演算 (+, −, ∗, /), 単項演算 (符号, 積分記号), べき乗, 対数, 指数, 三角関数等) に着目しその Taylor 展開の各項を再帰的に求める. 次に常微分方程式 (あるいはその系) の右辺を再帰関係を用いて書くことによりべき級数解の各項の係数を得るものである. さらに Barton らは『基本関数』として, 二項・単項演算子のみを採用し, 方程式の右辺にべき乗 $z=y^q$ (q: 実数) 等の非有理型を含む場合には $z'=qzy'/y$ のように変形し, 方程式の元数を増やして対処するような方法で再帰的 Taylor 級数法をシステム化している[5),6]. すなわち方程式を1階連立有理型に変形する前処理をほどこしている. 例えば, 三角関数を含む微分方程式

$$dy/dx=\sin(y) \qquad (3)$$

に対し $y_1=y$, $y_2=\sin(y)$, $y_3=\cos(y)$ と元数を増やし,

$$y_1'=y_2,\ y_2'=y_3y_2,\ y_3'=-y_2^2$$

のように変換するような前処理を行う. この過程を大規模な pre-compiler を用いて数値計算法とすることにより, 計算速度などでも他の数値計算法に劣ることが無く実用に耐え, かつ stiff な微分方程式系への適用も可能となっている[7),9]. さらに, Taylor 級数展開の各項の展開係数を利用して特異点の位置の探索など解の挙動に関する情報も得られる. しかし, 再帰関係の計算などの過程で複雑な数式変形が必要となり, FORTRAN 等の数値計算言語では処理系が大規模で複雑になる. さらに, 形式解を数値的に各 x ごとに数値でしか求めないため初期値の変化等に対しては計算を最初から遂行しなおす必要があり, また解の挙動に

ついての情報を得るにも多くの計算量が必要となる．

4. ハイブリッド処理と Taylor 級数法

3章で述べた Taylor 級数法の持つ数式変形の困難に対して数式処理システムを導入し効率化をはかることは容易に想像できる．図2に示したようなハイブリッド処理を考えるとこの点は容易になる．以下にパソコン上で稼働する我々の小型ハイブリッド処理システムへの Taylor 級数法の組み込みについて述べる．

4.1 アルゴリズムについて

Taylor 級数法の実現に対しては，a）**基本関数**の取り方，b）再帰関係についての問題点等を考慮する必要がある．3章で述べたように，現在大型の数値計算を意図して開発されている方法では，高速にかつ精度よく数値解を求めることを目的とするので**基本関数**は，計算機の組み込み関数を使用できるように多くの種類が選択されている．しかし，数式処理システムでは簡素化処理の遂行のため**基本関数**の増加は対応する関数の処理ルーチン（三角関数間の関係等）の増加を必要とし，関数に対するデータベースが大量なものになる．これはメモリ容量の増加や処理時間の急激な増大をまねきかつ簡素化処理が複雑になるため小型システムには適切ではない．したがって，前処理として**基本関数**に二項演算子および単項演算子（符号，積分記号）のみを採用し，方程式を1階連立有理型に変形する．

形式解を得る過程で，再帰的 Taylor 級数法では数式微分の代用として再帰関係を定義したが，ハイブリッド・システムではこの部分を通常の数式微分の処理に置き換えることも可能である．システム内に組み込まれた数式処理用の述語 diff による数式微分を利用した場合と再帰関係を利用した場合との演算量（右辺の複雑度）に対する速度比較を**図3**に示した．常微分方程式 $y'=y+\cdots+y$ の右辺を敢えて n 回の加算として計算する．この時，数式微分を用いる方法（diff）と数値計算で用いられる再帰関係のみを用いる方法（recursive）とについて右辺の加算回数（n）に対する処理時間の変化が示されている．order は Taylor 級数法で求める展開の次数を表す．両者を比較すると，右辺が簡単なもの（n：小の場合）に対しては後者が

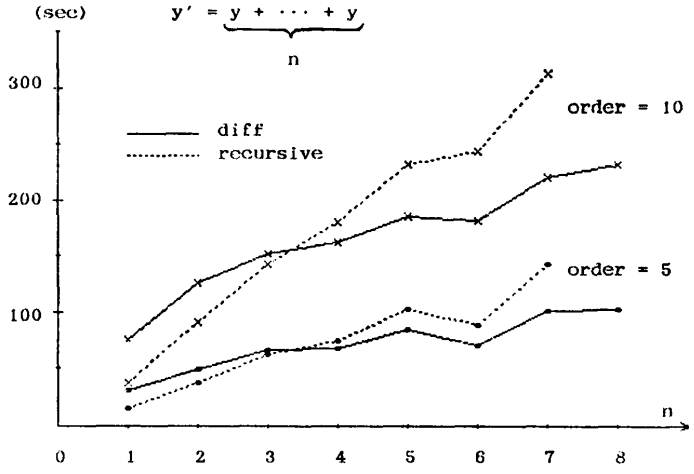

図3 $y'=y+\cdots+y$ に対する形式解を得るためのアルゴリズムの処理速度の比較（n：演算数）

Fig. 3 Execution times to obtain formal solutions by recursive and symbolic differentiation algorithms for $y'=y+\cdots+y$ where n stands for terms (y) in r. h. s.

優位を示すが，右辺の複雑化と共にプログラムの出力に至る数式処理部分の速度は数式微分を用いる方が再帰関係を用いる方法より優れている．右辺に，より複雑な演算が入るとこの差はさらに顕著になる．またプログラムの容易さは比較以前の問題である．そこで，以下では与えられた常微分方程式に対し，右辺を一階連立有理型に変形する前処理の後，数式微分を行う方法を採用する．数値計算法として提唱されている手法はハイブリッド・システムでは変更される方が有利である．

4.2 データ構造

ハイブリッド処理システムにおける重要な問題点は数式処理等に適した記号処理言語と数値計算に対する技術計算との間のデータ構造の相違にある．すでに2章で述べたように，システムの記述言語そのものをCで作成しておくとこの問題の解決は容易になる．ここで採用している PROLOG でも C の関数を直接用いることが可能である．次に，システム内で用いるデータ構造について述べる．その基本はある種の有理式の正準表現であり，多項式の各項をその係数と指数より次のように表す．

$$\text{const}\cdot\alpha^{P}\beta^{Q}\cdots\gamma^{R}->(\text{const},\ P,Q,\cdots,R) \quad (4)$$

const は定数であり，整数，実数の両者が可能である．またここでは(5)を拡張して効率化をはかるため有理式型のデータ構造を考える．すなわち，

式:=分子/分母 ->
((分子のリスト),(分母のリスト)) (5)

とする．分子（あるいは分母）のリストは(4)のリストの和である．ここで導入したデータ構造は，分子・分母の最大公約数を求めることによって約分操作を容易にするばかりでなく，有理関数近似などにも用いることが可能になる．

4.3 FORTRAN プログラムの出力

ハイブリッド処理システムの特色の一つに簡単な式計算の他に数式処理結果の数値計算プログラム（主にFORTRAN）としての出力あるいは他のプログラムとの結合，そのシステム内での計算および数値結果の数式処理での再利用等がある．ここで述べるシステムでは FORTRAN プログラムの作成のために次のような述語 fortran を定義する．Taylor 展開により求められた各展開係数はまず PROLOG の組み込み述語 assert によりデータベースに登録される．これらはまず，horner 法を用いてプログラムでの演算回数を少なくするよう変形される．前処理の結果設定された方程式の元数等の値を与え，FORTRAN サブプログラムが作成される．さらにあらかじめシステム内に用意された主プログラムと結合される．具体的に微分方程式(3)に対し，初期条件を $x=x0$, $y(x0)=y10$ とした場合の主プログラムとシステムで作成されたサブプログラムとが**図 4**(a)，(b)に示されている．図4(b)の下線部のみが方程式ごとに異なる．この部分以外は Taylor 級数法実行のためにあらかじめシステムに組み込まれている．なお，Taylor 展開の次数は5次までとしている．また，前処理より

$y20=\sin(y10)$, $y30=\cos(y10)$

である．この FORTRAN プログラムは MS-DOS の EXEC 命令により PROLOG から OS にもどることなく翻訳，結合，実行がなされる．したがって，この数値結果を再度システム内で数式処理に用いることも

```
      REAL*8 X0,X,XI,XD,WIDTH,Y(20),A(20)
      INTEGER NOFY,NOFA,I,J,K,N,NP,M
C
      CALL INFORM(NOFY,NOFA)
C
      WRITE(*,*) 'Initial Conditions ... '
      READ(*,*) XI,(Y(I),I=1,NOFY)
      IF(NOFA.EQ.0) GOTO 10
      WRITE(*,*) 'Constant Symbols ... '
      READ(*,*) (A(I),I=1,NOFA)
   10 WRITE(*,*) 'Destination Point   '
      READ(*,*) XD
      WIDTH=XD-XI
      WRITE(*,*) 'Number of Divisions  '
      READ(*,*) N
      WRITE(*,*) 'Print-out Steps  '
      READ(*,*) NP
C
      X0=XI
      M=N/NP
      K=0
      WRITE(*,*) X0,(Y(I),I=1,NOFY)
      DO 20 J=1,N
        X=XI+WIDTH*J/N
        CALL EVAL(X,X0,Y,A)
        K=K+1
        IF(K.GE.M) THEN
          K=0
          WRITE(*,*) X0,(Y(I),I=1,NOFY)
        ENDIF
   20 CONTINUE
      STOP
      END
```

(a)

```
      SUBROUTINE INFORM(NOFY0,NOFA)
      NOFY0=1
      NOFA =0
      RETURN
      END
C
C
      SUBROUTINE EVAL(X1,X0,Y,A)
      REAL*8 X1,X0,Y(20),A(20),H,W
      REAL*8 Y10,Y20,Y30
      Y10=Y(1)
      Y20=SIN(Y10)
      Y30=COS(Y10)
      H=X1-X0
      W=(5*Y20**5-18*Y20**3*Y30**2+Y20*Y30**4)/120.D0
      W=W*H-(5*Y20**3*Y30-Y20*Y30**3)/24.D0
      W=W*H+(Y20**3-Y20*Y30**2)/6.D0
      W=W*H+Y20*Y30/2.D0
      W=W*H+Y20
      W=W*H+Y10
      Y(1)=W
      X0=X1
      RETURN
      END
```

(b)

図 4 生成された FORTRAN プログラム．(a) システム内に組み込まれた主プログラム，(b) $y'=\sin(y)$ に対する副プログラム（作成：下線部）

Fig. 4 The FORTRAN program produced by the system (a) main program in the system, (b) subprogram produced by the system (underline) for $y'=\sin(y)$.

```
                                        3         2
                       y20 y30       2   y20  - y20 y30        3
y1 = y10 + y20 (x - x0) + ------- (x - x0)  - --------------- (x - x0)
                          2                      6


     3          3               5        3    2        4
 5 y20  y30 - y20 y30        4  5 y20  - 18 y20  y30  + y20 y30
- -------------------- (x - x0)  + -------------------------------- (
         24                                 120



     5
x - x0)
```

図 5　形式解の2次元出力
Fig. 5　Well formed outputs of the Taylor series.

可能である．

4.4　形式解の出力

形式解を出力する場合も assert によりデータベースに登録された展開係数にもとづくのは FORTRAN 出力の場合と同様であるが，この場合には出力が視覚的に十分理解しうるようになされる必要がある．このために述語 show を定義する．4.2 節で述べたデータ構造の特色を用いて上の例(3)に対し，**図 5** のように画面あるいはプリンタに2次元出力をする．ここではその一部分のみ（展開の次数を5次まで）を示したが，このような出力の方法は MACSYMA 等の大型機用の数式処理システムに見られるのみである．

4.5　特異点の位置の探索

3章で述べたように，Taylor 級数法のようにべき級数で表された形式解を求め得る手法の持つ特長の一つに，他の数値計算法と違って解の挙動などの情報を得ることができることがある．Corliss らも **3 項解析**等により，原点に最も近い特異点の位置（収束半径）を得ている[6]．しかしそこでは数値計算を基本としているため Taylor 展開係数が数値としてしか求められない．よって，**3 項解析**等のための数値計算の計算量が大きくなる．これに対し，我々のハイブリッド・システムによる場合は形式解が数式で求められており，かつ **3 項解析**等のための FORTRAN プログラムの出力も容易である．以上の操作に対するシステムへの入力などが**図 6** に示される．下線部が利用者の入力を表す．

5. ハイブリッド処理による Taylor 級数法

Taylor 級数法を上に述べたハイブリッド処理システムで実行したいくつかの結果をまとめる．

次の2つの常微分方程式の初期値問題を考える．

$$y'=10(y-x^2),\ y(0)=0.02 \qquad (6)$$

$$y'=y-2x/y,\ y(0)=1 \qquad (7)$$

Barton らによれば，12 次までの展開係数を取った計算ながら(6)は Taylor 級数法が他の数値計算法より優れており，(7)は解曲線が複雑な場合である[5]．我々の場合，(6)，(7)に対する Taylor 級数法による数値解の厳密解との相対誤差が各々対応する Runge-Kutta 法（4次の古典的 Runge-Kutta 法）によるものと共に**図 7**(a)，(b)に示されている．Taylor 級数法での展開次数は計算時間とメモリを考え6次までとしたが十分な精度で解が求まることがわかる．数値計算は倍精度で行った．

以上の例から分割幅 h をある程度小さくすると，ハイブリッド・システムに組み込んだ Taylor 級数法は数値解法としても通常の Runge-Kutta 法と比較しても良い性能を示すことがわかる．特に，(6)のように Taylor 級数が切れる（解析解が存在する）場合など

```
root/user/?-system.

How many equations ? 1       ( 方程式の数 )

1> y'=y+sin(y)       ( dy/dx = y + sin(y))

independent variable = x

Wait....

root/user/?-expand.          ( Taylor 展開 )

order = 5                       ( 5 次まで )

root/user/?-show.             ( 2次元出力 )

 . . .

root/user/?-fortran.        ( FORTRAN 実行 )

  ( FORTRAN コンパイラのメッセージ )

  ( 結果の出力 )

root/user/?-exec('type evalsub.f77').

  ( FORTRAN サブプログラムの表示 )

 . . .
```

図 6　システムの動作例（下線部は利用者の入力）
Fig. 6　The user interface of hybrid computation system. User inputs are shown with underlines.

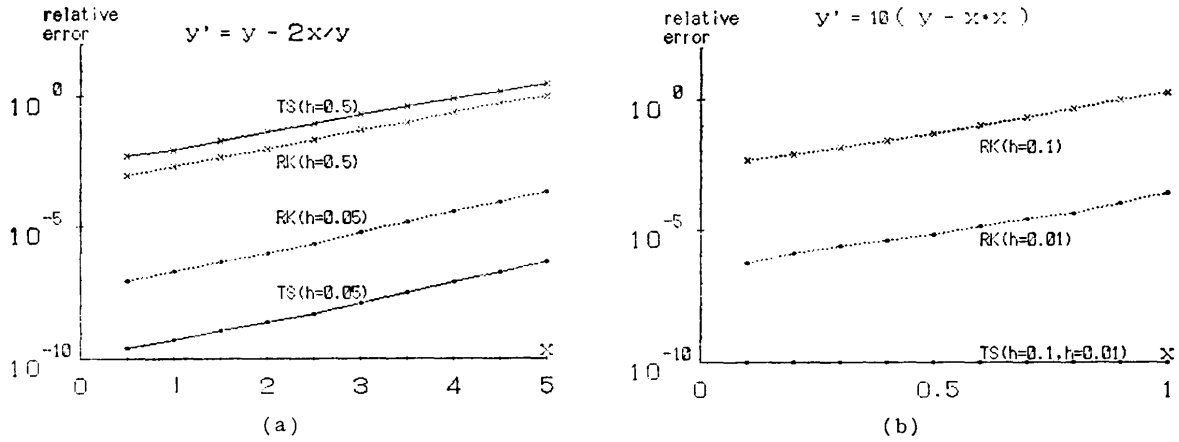

図 7 Taylor 級数法 (TS) と Runge-Kutta 法 (RK) による解の精度比較
h は分割幅を，縦軸は真の解との相対誤差を表す．方程式は (a) $y'=y-2x/y$, (b) $y'=10(y-x^2)$

Fig. 7 Relative errors for the Taylor series method (TS) and the Runge-Kutta method (RK), where h stands for step sizes. ODEs are (a) $y'=y-2x/y$, (b) $y'=10(y-x^2)$.

は Taylor 級数法による良さは明確である．また，高次の Runge-Kutta 法や分割幅を自動的に求めるようにした Runge-Kutta-Fehlberg 法等は計算量が莫大なものになり，手近なパソコン等での計算法としては適当とはいえないが，ここで述べた Taylor 級数法ではより高次の展開係数の計算も困難ではないので十分に精度の良い解を求め得ることも可能である．さらに，形式解が数式で求められているのでいろいろに初期値を変化させたり，パラメータを含んだ微分方程式の数値解を得る場合にここに述べた方法はより有力になり得ることを付け加えておく．

上の場合，比較は単に数値結果のみについて行ったがハイブリッド処理システム上の Taylor 級数法では 3 章，4 章でも述べたように形式解やその漸近挙動等に対する情報を得ることができる．特異点の位置の探索に関して著名な Painlevé-1 型方程式

$$y''=6y^2+x \qquad (8)$$
$$y(0)=1,\ y'(0)=0$$

を取り上げる．原点 ($x=0$) に最も近い極は 1.206 および -1.250 に現れる[6]．これに対し，3 項解析を行う．データベースに登録されている展開係数から容易に 3 項解析に対するプログラムが作成される．$x=0$ から正方向の特異点を求める場合 (1.206 を得る場合) について図 8 に示した．今，次数 5, 6, 7 の 3 項を考えると，$x=0.3$ では特異点までの距離 1.8513 が，$x=0.4$ では 1.2755 が得られ x を 1.2 に近づけると 1.206… に収束していることがわかる．より高次の展開係数を用いると特異点の位置を求めるまでの計算回数は減少する．いろいろの次数 (order=(9, 10, 11), (13, 14, 15)) での結果がやはり図 8 にある．$x=0$ から負方向への探索により同様に特異点 -1.250 を得ることは容易である．なお，これらの計算はすべて 4 章で述べたパソコン FM 16β 上でなされていることを付記する．

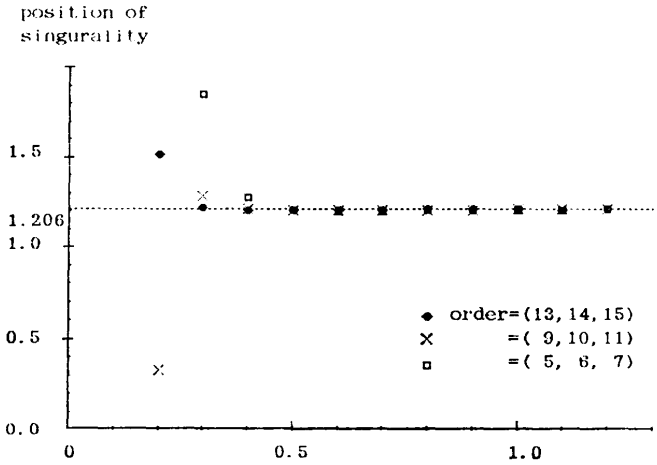

図 8 $y''=6y^2+x$, $y(0)=1$, $y'(0)=0$ に対する特異点探索 (3 項解析)

Fig. 8 The position of singularity for $y''=6y^2+x$, $y(0)=1$, $y'(0)=0$ by 3-term analysis.

6. ま と め

以上のように，本論では PROLOG を記述言語とした小型のハイブリッド処理システムの設計とその上に

導入した常微分方程式の解法について述べた．

ここで作成したハイブリッド処理システムは数値計算との結合を目指したもので，数式の入力，簡素化，いろいろな記号計算はもとより FORTRAN による数値計算プログラムの出力やその実行をシステム内で行う．また，数式の出力も従来は大型の数式処理システムでしかなされなかったような機能が付加されている．

常微分方程式の解法としては，数値計算の分野ではその記号処理の複雑さにより，一般化していなかった Taylor 級数法を考えた．この方法をハイブリッド処理システムに導入することにより形式解を得ることはもちろん，その FORTRAN プログラムを発生させ，数値計算を行うことができる．得られた数値解に対する考察により以下の結論を得る．ハイブリッド処理システム上で考えられた常微分方程式に対する Taylor 級数法は，数値解法として一般的な Runge-Kutta 法と十分に比較し得るものである．特に初期値やパラメータの変化に対応する場合にはより強力になる．Taylor 級数法により数式的に得られた形式解を利用して3項解析のための FORTRAN プログラムを作成し，特異点の位置の探索も容易に行える．

数値計算のみ，あるいは数式処理のみではあまり効率的とは思われなかった解法が数値・数式のハイブリッド処理を考えることにより非常に意味あるものとなることがわかる．この種のアルゴリズムの再発見をより広い分野の問題に対して行うことにより，ハイブリッド処理システムの強化・拡充のみならず数式処理，数値計算の両分野の新しい進展の方向が示されるものと思われる．

参 考 文 献

1) 三井斌友：数式処理と数値処理との界面，情報処理，Vol. 27, No. 4, pp. 422-430 (1986).
2) 元吉文男，佐々木建昭：数式処理の歴史と将来の展望，情報処理，Vol. 27, No. 4, pp. 362-370 (1986).
3) 対馬勝英：小型数式処理システムとその応用，情報処理，Vol. 27, No. 4, pp. 379-387 (1986).
4) 野田松太郎，戒能芳弘：数値計算支援を目的とした小型数式処理システムの設計，数式処理通信，Vol. 3, No. 1, pp. 54-58 (1985).
5) Barton, D., Willers, I. M. and Zahar, R. V. M.: Taylor Series Method for Ordinary Differential Equations, in Rice, J. R. (ed.), *Mathematical Software*, Academic Press, New York, pp. 369-389 (1971).
6) Corliss, G. and Chang, Y. F.: Solving Ordinary Differential Equations Using Taylor Series, *ACM TOMS*, Vol. 8, No. 2, pp. 114-144 (1982).
7) Kedem, G.: Automatic Differentiation of Computer Programs, *ACM TOMS*, Vol. 6, No. 2, pp. 150-165 (1980).
8) Moore, R. E.: *Methods and Applications of Interval Analysis*, SIAM, Philadelphia (1979).
9) Barton, D.: On Taylor Series and Stiff Equations, *ACM TOMS*, Vol. 6, No. 3, pp. 280-294 (1980).

(昭和61年10月23日受付)
(昭和62年4月15日採録)

野田松太郎（正会員）

昭和44年大阪市立大学理学研究科博士課程修了．理学博士．日本学術振興会奨励研究員を経て愛媛大学工学部電子工学科に勤務．現在，同助教授．数値計算と数式処理の結合と「AI 化」およびそれに基づく新しいアルゴリズムの開発に強い関心を持っている．ACM，ソフトウェア科学会，人工知能学会，日本物理学会各会員．

岩下 英俊（学生会員）

昭和38年生．昭和61年愛媛大学工学部電子工学科卒業．同年同大学工学研究科（修士課程）入学．現在に至る．小型数式処理システムの開発研究に従事．電子情報通信学会会員．