3X-2

小型ハイブリッド処理システムによる常微分方程式の解法

岩下英俊、 野田松太郎
(愛媛大学工学部)

はじめに

数式処理システムへの数値計算機能の付加或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」の確立は現代の最重要の課題の一つである。
我々はこれを、より使い易い環境のもとで実現するためパソコン上でも稼働する PROLOG で記述された小型ハイブリッド処理システムの開発を行っている。[1]このような処理系を用いることにより、単に数値計算の支援システムとするばかりでなく、アルゴリズムの再開発や数学研究、数学教育の分野への利用等も考えられる。各種微分方程式の解法を考えても、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事に重点を置かれ、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ていない。理想的な「ハイブリッド」処理を用いると、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。本講演では、常微分方程式に対する近似解法である Taylor 級数法とその拡張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述べる。

常微分方程式に対する Taylor 級数法

実際の工学的分野に出現する大半の微分方程式では厳密な解析解を持つ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。しかし、解の漸近的振舞いを知る要求には数値計算のみでは不十分であり、ハイブリッド処理システムあるいは摂動法的な解法の自動化が要求される。このために、常微分方程式に対する基本的でかつ有力な解法であるとされながら計算過程で複雑な文字処理などを要するため、十分に汎用的になっていない Taylor 級数法について我々のシステムでの可能性について調べる。

今、与えられた常微分方程式の系を

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

とする。この時、形式解を得ようとすると $x = x_0$ に関して Taylor 展開を考えるのが通常である。これを数値計算のアルゴリズムとして確立したものが Moore により定式化された再帰的 Taylor 級数法である。[2]まず、常微分方程式系の右辺を『基本関数』(二項演算(+,−,•,/)、単項演算(符号、積分記号)、べき乗、log、exp、sin、cos 等)のみの有理型で書く。$f_i(x, y_i(x))$ は多くの補助変数 T_s により書き直される。なお Barton 等は『基本関数』として、二項•単項演算子のみを取り、べき乗 $z = y^q$ (q:実数)のような非有理型の場合には $z' = qzy'/y$ のように変形し、方程式の元数を増やして対処する方法で再帰的 Taylor 級数法をシステム化している。ここでは、三角関数等を含む場合にも同様にする。[3]。

現在、大規模な pre-compiler により数値計算法として発表されているものは Moore のように、多くの『基本関数』に対する再帰関係を定義している。[4]これ等は計算速度などでも他の数値計算法方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である。[5]しかし、補助変数の導入や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る。このため、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 ごとに求めるため時間的な無駄をしかつ解の挙動に付い

Taylor series method for ODE in Hybrid system
Hidetoshi IWASHITA and Matu-Tarow NODA
Ehime University

ての情報を得る事もできない。

ハイブリッド処理での Taylor 級数法（１）

再帰的 Taylor 級数法の良さをいかし、かつ欠点を補足するには pre-compiler 部を数式処理化するハイブリッド処理の活用が当然考えられる。特にこれを小型システムにも適用しようとすると、数式処理段階での生成されたべき級数の簡素化等でのコスト、三角関数のデータベース構築の煩雑さ等の考慮から、『基本関数』を増やさず前処理を遂行する事が重要である。本論では T_s に対する再帰関係を二項演算子、単項演算子のみについて定める。

$$T_s = T_l \pm T_m \longrightarrow T_s^{(j)} = T_l^{(j)} \pm T_m^{(j)}$$

$$T_s = T_l \cdot T_m \longrightarrow T_s^{(j)} = \sum_{k=0}^{j} T_l^{(k)} T_m^{(j-k)}$$

$$T_s = T_l / T_m \longrightarrow T_s^{(j)} = \frac{1}{T_m^{(0)}} \left[T_l^{(j)} - \sum_{k=1}^{j} T_m^{(k)} T_s^{(j-k)} \right]$$

$$T_s = \int T_m \longrightarrow T_s^{(j)} = T_m^{(j-1)} / j$$

パソコン用ＯＳの MS-DOS 上で基本の PROLOG 処理系には標準的な DEC10 形式の A.D.A.PROLOG(VMA) を用いた小型システムで図のような式の入力により自動的に Taylor 級数展開の実行、その簡素化あるいは FORTRAN プログラムの出力がなされる。これらの過程の大半は単なる数式照合操作であり、システムの記述言語の PROLOG に最適である。図のような場合には、MS-DOS の EXEC 命令により生成された FORTRAN のサブプログラムのコンパイルとシステムに組み込まれている主プログラムとの結合や実行がなされる。

ハイブリッド処理での Taylor 級数法（２）

上の手法の中、数式処理では敢えて再帰関係を構成しなくとも微分演算のみで形式解を構成しうることは明らかである。右辺か非有理型であるときの方程式の元数を増やすことによる有理化の手法のみを Barton から借用することにより、ハイブリッド処理での Taylor 級数法をより効率的に構成しうる。システムへの入力、途中結果の簡素化、結果の蓄積、プログラム出力などは上に述べたものと同一であり、再帰関係の計算を微分演算用のシステム内述語 diff で代用した点のみが違う。両者を比較すると、プログラムの出力に至る数式処理部分の速度では後者のように右辺の有理化の前処理＋数式微分の手法の方が数値計算で用いられる再帰関係のみを用いる方法より僅かながら優れていることがわかる。

むすび

現在、開発している小型のハイブリッド処理システムの常微分方程式の解法への適用について述べた。 Taylor 級数法が効率的にインプリメントされることがわかる。数式的に求められ、蓄積された結果により解の漸近的振舞いなどの数式的評価も可能になる。

参考文献

1)野田、戒能：数式処理通信,3(1985)

2)R.E.Moore:"Methods and Application of Interval Analysis",SIAM(1979)

3)D.Barton,I.M.Willers and R.V.M.Zahar; In "Mathematical Software" ed. J.Rice, Academic Press(1971)

4)G.Kedem;ACM Trans.Math.Soft,6(1980) G.Corliss and Y.F.Chang;ACM Trans.Math.Soft,8(1982)

5)D.Barton;ACM Trans.Math.Soft,6,(1980)

図　システムへの入力

```
root/user/?-system.
How many equations ? 1      ( 方程式の数 )
1> d(d(y))=y+sin(2*y)       ( d²y/dx²=・・・ )
independent variable = x
Wait....
root/user/?-expand.         ( Taylor 展開 )
order = 10                  ( 10 次まで )
root/user/?-fortran.        ( FORTRAN 実行 )
  ( FORTRAN コンパイラのメッセージ )
  ( リンカーのメッセージ )
  ( 結果の出力 )
root/user/?-exec('type evalsub.f77').
  ( FORTRAN サブプログラムの表示 )
```