

Taylor展開法による常微分方程式の数値解法

平山 弘

機械システム工学科

Numerical Solving for Ordinary differential equations by Taylor series

Hiroshi HIRAYAMA

Abstract

The arithmetic operations and functions of Taylor series can be defined by C++ language. The functions which consist of arithmetic operations, pre-defined functions and conditional statements can be expanded in Taylor series. Using this methods, the solution of an ordinary differential equation can be expanded in Taylor series and expanded up to arbitrary order, so the calculation formula of arbitrary order can be used instead of Runge-Kutta formula. Taylor series can be used for the evaluations of the errors and the optimal step size within given error allowance easily. In addition, we can transform Taylor series into Padé series, which give arbitrary order, high precision and A-stable formula for solving ordinary differential equation numerically.

Key Words: Taylor series, C++ language, power series, A-Stable, ordinary differential equation

1. はじめに

関数をTaylor展開することは、自動微分と呼ばれる方法¹⁾と数学的には同じものである。この自動微分を利用した常微分方程式の数値解法については、伊理、小野、戸田⁶⁾⁹⁾によって、論じられている。この論文では、自動微分を利用して、微分係数が得られるので、これを利用するとRunge-Kutta法を改良することができることを論じている。しかしながら、この方法は、まだ一般化していないプリ・プロセッサの存在を仮定している。また、高次の公式を得るには、通常Runge-Kuttaの公式¹³⁾と同様にかなり難しいように思われる。

常微分方程式の解を、任意の次数までTaylor展開できることは、数学的には、常微分方程式の級数展開による解法と同じであり、その可能性については、久保田⁷⁾などで述べられており、Corliss and Chang¹⁾によって、任意の次数のTaylor展開式を利用した常微分方程式を解くためのパッケージ・ソフトウェアが発表されている。Corliss and Changのソフトウェアは、上記の自動微分と同じように

プリ・プロセッサを利用している。パッケージ・ソフトウェアであるので、簡単な常微分方程式を解くだけならば、使い易いと思われるが、大きなプログラムで、このプログラムを参照することは、かなり難しいと思われる。

本論文では、プログラムでよく使われる演算子(+, -, *, / など) を、被演算の型が異なる場合、別の意味を与えることができるC++言語²⁾の機能(オペレータ・オーバーロード、operator overload)を使い、有限項で打ち切ったTaylor級数間の四則演算、Taylor級数の関数演算を定義する。この機能を使うと、プログラムの形で与えられた任意の関数をTaylor展開することができる。常微分方程式の初期値問題の解を任意次数のTaylor展開の形で、容易に得られる。得られたTaylor級数解を数値計算に利用すれば、任意の次数の数値解法が得られる。さらに、このTaylor展開をある次数のPadé展開式に変形すると、任意次数のA安定な数値計算方法を与えることを示すことができる。

ここで、提案する計算法は、現在最もよく使われる言語の一つであり、入手が容易なC++言語を使っている。このような計算方法は、数値計算でよく使われるFORTRANの新しい規格であるFORTRAN 90でも可能である。

2. Taylor級数

この節では、関数をTaylor展開するための基本的な考え方を説明し、その計算方法について簡単に論じる。詳しくは、Rall¹¹⁾、Henrici⁴⁾や平山⁵⁾などに述べられている。

2.1 Taylor級数の四則演算

Taylor級数の四則計算のプログラムは、以下のように簡単に作ることができる。平行移動によって、展開位置を移動することができるので一般性を失うことなく、原点で展開した式だけを扱うことができる。この級数を次のように定義する。

$$f(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \dots \quad (2.1)$$

$$g(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \dots \quad (2.2)$$

$$h(x) = h_0 + h_1 x + h_2 x^2 + h_3 x^3 + \dots \quad (2.3)$$

このとき、四則演算は、以下のように定義できる。これらの公式は簡単なものであるがまとめて、記載されている文献があまりないので以下に記載する。ここで、 m は、演算の対象となっているTaylor級数の次数である。

$$(1) \text{ 和差 } h(x) = f(x) \pm g(x)$$

係数は次の式によって計算することができる。

$$h_n = f_n \pm g_n \quad (n = 0, \dots, m) \quad (2.4)$$

$$(2) \text{ 乗算 } h(x) = f(x)g(x)$$

係数は次の式によって計算することができる。

$$h_n = \sum_{k=0}^n f_k g_{n-k} \quad (n = 0, \dots, m) \quad (2.5)$$

$$(3) \text{ 除算 } h(x) = \frac{f(x)}{g(x)}$$

係数は次の式によって計算することができる。式からわかるように、 $g_0 = 0$ のときは、計算することはできない。ただし、 $f_0 = g_0 = 0$ の場合は、分子と分母を x で割る操作を行う。この操作で、 $g_0 \neq 0$ になれば、以下の式で除算を行うことができる。

$$h_n = \frac{1}{g_0} \left(f_n - \sum_{k=0}^{n-1} h_k g_{n-k} \right) \quad (n = 0, \dots, m) \quad (2.6)$$

この公式は、 $g(x)h(x) = f(x)$ とおいて、(2.1)、(2.2)、(2.3)の式を代入して、展開し、各次数の係数が等しいと置いて得られる。

2.2 Taylor級数の関数

他の多くの初等関数でも同様に係数間の漸化式を導くことができる。ここでは、Taylor級数の指数関数の計算方法を示す。この方法は後に述べる常微分方程式の解のTaylor展開の単純な例にもなっている。

$$h(x) = e^{f(x)} \text{ とおくと、}$$

$$\frac{dh}{dx} = h \frac{df}{dx} \quad (2.9)$$

である。(2.8)に(2.1)、(2.3)を代入して、両辺の係数を比較することによって、次のような関係が得られる。

$$h_0 = e^{f_0}, h_n = \frac{1}{n} \sum_{k=1}^n k h_{n-k} f_k \quad (n \geq 1) \quad (2.10)$$

このように簡単に計算することができる。対数関数や三角関数の場合も同様な公式を得ることができる。

式(2.10)のように、Taylor展開式の指数関数の計算には、指数関数の計算は、1回しか入らないので、Taylor展開の指数関数の計算量は、通常の四則演算とそれほど大きな違いにはならない。このことは、対数関数や三角関数などでも同様である。

3. 常微分方程式の解のTaylor展開

この節では、Taylor展開の方法を利用して、常微分方程式の解を任意の次数までTaylor展開する方法を示す。Taylor展開された解は、任意次数の一段公式として、Runge-Kutta公式の代わりとして利用できる。また、解の許容誤差を与えたとき、その範囲で最良の刻み幅を決めることもできる。

3.1 解のTaylor展開

次のように $\frac{dy}{dx}$ について解かれた形になっている

常微分方程式について考える。

$$\frac{dy}{dx} = f(x, y) \quad (3.1)$$

ここで、 y および f は、一般にベクトルである。解 y は以下のように、 x について、Taylor展開できるものとする。

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots \quad (3.2)$$

このとき、(3.1)の左辺は

$$\frac{dy}{dx} = a_1 + 2a_2x + 3a_3x^2 + \dots \quad (3.3)$$

となる。 a_0 は初期条件から決定される。 a_1 は (3.1) の右辺に、 $y = a_0$ として代入し、 x の0次まで右辺を計算する。この計算結果と (3.1) の左辺である (3.3) とを比較することによって、 a_1 が決定される。次に、

$$y = a_0 + a_1x \quad (3.4)$$

として、(3.1) の右辺を x の1次まで計算し、(3.3) と比較すれば、 a_2 が決定される。

一般に、 n 次の係数を決定するのに、 $n-1$ 次のTaylor展開式を利用して、(3.1) の右辺を計算し、 $n-1$ 次までのTaylor展開を得る。その式は (3.3) に等しいので、各次数の係数を比較することによって、 n 次の係数を決定することができる。

この操作を何回も繰り返せば、任意の次数まで解を計算することができる。解析的には、これで微分方程式の解が得られたことになる。

この計算は、Picardの逐次計算法¹²⁾

$$y_0 = a_0 \quad (3.5)$$

$$y_n = a_0 + \int_0^x f(x, y_{n-1}) dx \quad (3.6)$$

と計算を繰り返し計算する方法とほぼ一致する。ここで、「ほぼ一致する」の意味は、本方法では、(3.6) の右辺の積分は、 x の $n-1$ 次までしか正しく計算しないが、Picardの方法では、厳密な計算を行う点異なる。

また、Bessel関数 $y = J_0(x)$ が満たす方程式

$$\begin{cases} \frac{dy}{dx} = z \\ \frac{dz}{dx} = -\frac{z}{x} - y \end{cases} \quad (3.7)$$

$x=0$ のとき $y=1, z=0$ も容易に解くことができる。この方程式は、従来のRunge-Kutta法を使うと、 $x=0$ を代入した右辺の値が必要となるが、ゼロでの割り算が発生し、計算ができなくなり、破綻する。本方法では、求まる解の次数は低くなるが問題なく計算できる。

3.2 誤差評価

(3.2) のように得られた n 次のTaylor展開式は、数値計算法の公式と見ると、 n 次のRunge-Kutta公式に相当するものが得られたことになる。 n は任意であるから、任意の次数のRunge-Kuttaの公式に相当するものが得られたことになる。

常微分方程式の解がTaylor級数で得られるので、

誤差評価もFehlbergの公式⁸⁾ に代表される埋め込み型公式と類似な方法で行うことができる。 n 次の展開式が得られた場合、 n 次まで計算した値と $n-1$ 次までの値とを比較することによって、誤差を推定することができる。誤差評価に使われる $n-1$ 次のTaylor展開式は、 n 次式の一部であるから、計算は、 n 次の項の計算だけで済むので効率的である。

このように、誤差評価が容易にできるので、要求精度にあわせて、従来のRunge-Kutta法と同様な方法で、刻み幅を調節することもできる。これまでのRunge-Kutta法では、不可能であった、計算の次数の調整も可能となる。

4 Padé展開

(3.2) のように、常微分方程式の解は、簡単にTaylor展開の形で得られる。このTaylor展開式を、有理関数展開であるPadé展開することによって、安定な計算法が得られる。

4.1 Padé展開の計算

Taylor展開は、容易にPadé展開することができる。Padé展開とは、以下のようにTaylor展開式を、有理関数に変形したものである。

$$a_0 + a_1x + a_2x^2 + \dots = \frac{p_0 + p_1x + \dots + p_Mx^M}{1 + q_1x + \dots + q_Lx^L} \quad (4.1)$$

(4.1) 式の両辺に右辺の分母を掛け、 $M+L$ 次の係数まで一致するように、有理関数の係数を決定することによって得られる。この条件は、次の様な式で表される。

$$a_l + \sum_{k=1}^m a_{l-k} q_k = p_l \quad (l = 0, \dots, M) \quad (4.2)$$

ただし、 m は、 l が L 以下ならば、 $m=l$ とし、 l が L を越えるならば、 $m=L$ とする。(4.2) と同じ関係式であるが、 l が M を越えている場合、次の関係式が得られる。

$$a_l + \sum_{k=1}^L a_{l-k} q_k = 0 \quad (l = M+1, \dots, M+L) \quad (4.3)$$

(4.3) の連立方程式を解き、有理式の分母の係数 (q_1, q_2, \dots, q_L) を決定し、その係数を (4.2) 式に代入して、分子の係数 (p_0, p_1, \dots, p_M) を求めることができる。

このとき、解かなければならない一次方程式 (4.3) の係数は、Toeplitz行列と呼ばれる特別な行列になる。この場合、一次方程式は、通常の一次方程式が未知数の数を n としたとき、 n^3 に比例する

のに対し、 n^2 に比例する高速な計算法¹⁰⁾が知られている。本プログラムでは、この方法によって、計算を行っている。ただし、 n が2、3のように、かなり小さい場合、Gaussの消去法が効率的なので、Gaussの消去法を使う。

Padé展開は、一般に、同じ次数のTaylor展開式より高精度で、収束の良い式を与えることができる場合が多いので、常微分方程式の解のTaylor展開式をPadé展開することは、効率的な常微分方程式の計算法を与えると期待できる。

4.2 A安定性

ここでは、Taylor展開式を、Padé展開することによって、高精度で効率的な計算法が得られるだけでなく、A安定な計算法になることを述べる。

テスト方程式

$$\frac{dy}{dx} = \lambda y \quad (4.4)$$

($x=0$ のとき、 $y=y_0$ とする。 λ は一般に複素定数である。)

を解くことを考える。この方程式の解のTaylor展開は、良く知られているように、

$$y = y_0 e^{\lambda x} = y_0 \sum_{k=0}^{\infty} \frac{(\lambda x)^k}{k!} \quad (4.5)$$

となる。A安定とは、 $\text{Re } \lambda x < 0$ のとき、(4.5)の右辺の級数部分の絶対値が1より小さくなることを言うから、無限級数(4.5)式を使う計算法は、A安定である。実際の計算では、有限項で打ち切った式を使うので、Taylor展開式を使う限りA安定な計算法は存在しない。多くの項をとれば、安定な領域は広くなり、A安定な式に近づくので、次数を上げることによって、より安定性のよい公式を作れる。Corliss等は、30次などのかなり高次の公式を使うことによって、安定化を計っている。

(4.5)の指数関数 $e^{\lambda x}$ のPadé展開については、良く研究されており、次のような結果が得られている。

定理： 指数関数 $e^{\lambda x}$ を(4.1)のように、分子 M 次、分母 L 次式にPadé展開したとき、 $M \leq L \leq M+2$ のとき、A安定である。

証明：E. Hairer and G. Wanner³⁾のIV章(60頁)を参照のこと。

したがって、分母の次数が分子の次数と同じか1または2次高い式に変形すれば、A安定な公式となる。常微分方程式の初期値問題に対する任意次数のA安定な解法が得られたことになる。従来のA安定な計算法は、非線型の方程式を解く必要

である¹⁴⁾のに対し、本方法では、連立一次方程式を解く程度の簡単な計算できる。このため、計算方法としても、かなり簡単であり、計算量も少ない。

また、本方法は、任意の次数まで計算できるので、問題にあった次数で計算することができる。これは、従来のA安定な計算法にはない特徴である。Padé展開したとき、分母の次数を分子より高い次数にすれば、L安定な公式ともなる。

5 数値例

この節では、Taylor展開法によって具体的に例題を解いた数値例を紹介する。最初の例題は、非常に簡単な例で、どのようにして解くかを示して説明する。以下の数値計算には、Borland C++ 4.5Jを使った。NEC PC-9821AP21を使用し、64ビットの倍精度浮動小数点で計算した。

5.1 簡単な例

簡単な例として、田中等¹³⁾によってテストされている、以下の方程式を解く。

$$\text{方程式：} \quad \frac{dy}{dx} = -\frac{1}{3}x^2 y^2 \quad (5.1)$$

$$\text{初期条件：} \quad y(2) = 1 \quad (5.2)$$

$$\text{理論解：} \quad y(x) = \frac{9}{x^3 + 1}$$

この方程式を0.5および0.1刻みで100ステップの計算を行う。この方程式は、stiffな問題ではないので、A安定な解法を使わない方が効率が良いこともある。

本方法で解くとき、準備しなければならないプログラム(関数)は、(5.1)の右辺に相当する関数である。これは次のようなものである。

```
1: void func( const taylor& x,
2:   const taylor& y, taylor& dy )
3: {
4:   dy = -1.0/3.0*x*x*y*y ;
5: }
```

ここで、 x 、 y 、 dy は、それぞれ(5.1)の x 、 y 、

$\frac{dy}{dx}$ を意味する。この関数を見ればわかるように、

Taylorをdoubleに変更すれば、従来の解法と同じ関数を準備するだけであることがわかる。この関数を利用して、(5.1)の解を、 $x=2$ でTaylor展開する

と、以下のような級数が得られる。

$$y = 1 - 1.33333 * x + 1.11111 * x^2 - 0.703704 * x^3 \\ + 0.345679 * x^4 + 0.115226 * x^5 + 0.00137174 * x^6 \\ + 0.13171 * x^7$$

この解は、7次まで計算したときの計算機からの出力である。現在使用しているTaylor級数を表現するデータの中には、展開位置の座標が含まれていないので、出力は常に原点における展開として出力される。この出力の場合、 x は、 $x-2$ を意味する。この式を分母3次、分子4次のPadé展開[3/4] (以後、分母M次、分子N次のPadé展開を[N/M]と記すことにする。)に変換すると、次の展開式が得られる。

$$\text{分子: } 1 - 9.63969e+14 * x + 6.75 * x^2 - 1.625 * x^3 \\ \text{分母: } 1 - 9.63969e+14 * x - 1.28529e+15 * x^2 \\ - 6.42646e+14 * x^3 - 1.07108e+14 * x^4$$

この出力の中の x も、 $x-2$ を意味する。これを使って、 $x = 2.5$ における y の値を計算すると $y = 0.5393258$ を得る。 $x = 2.5$ の時の y の値を初期値にして、 $x = 2.5$ におけるTaylor展開を行い、さらにPadé展開を行う。この式から、 $x = 3.0$ の時の y を計算する。この操作を何回も繰り返すことによって、微分方程式を解くことができる。

本方法の精度を調べるために、(5.1)の問題を刻み幅 0.5、0.1 で7次の公式[3/4]を使い100ステップ計算した。その相対誤差を田中等¹³⁾で引用されている同じ次数の既存の5公式と比較した。その結果をTable 1.に示す。同じ次数の場合、最良化された既存の公式よりは、良い結果は得られなかったが、本方法では計算の次数を簡単に上げることができるので、ある程度次数を上げた公式[9/11] (分子9次、分母11次) を使えば、従来の公式より、かなり良い結果が得られることがわかる。

この方程式の理論解は、Padé展開式を使えば厳密に表現できるのが、計算途中のTaylor展開式が厳密に表現できないため、Padé展開式も解析解と完全に一致することは無かった。このため、このテスト問題でも、数値例としての一般的な性質が出たと思われる。

5.2 弱stiffな問題

ここで導いた公式を使って、弱stiffな常微分方程式を解き、A安定およびA安定でない陰的Runge-Kutta法との比較を行う。比較のため、問題は、田中等¹⁴⁾によって、いろいろテストされている例題を使う。

$$\text{方程式: } \frac{dy}{dx} = 100(\sin x - y) \quad (5.3)$$

$$\text{初期条件: } y(0) = 0 \quad (5.4)$$

$$\text{理論解: } y = \frac{\sin x - 0.01 \cos x + 0.01 e^{-100x}}{1.001}$$

この問題を分子2次、分母2次のPadé展開法[2/2]で解いた結果をTable 2.に示す。このときのステップサイズ h は、0.15である。前の例題と同様に、100ステップ計算し、最初のステップと最後のステップの誤差および最大誤差を示した。分子9次、分母11次のPadé展開法[9/11]による結果もTable 2.に示す。この2つの方法以外の計算法による結果は、田中等¹⁴⁾から引用したもので、2段数陰的Runge-Kutta法による結果である。Table 2.に示した方法は、Jain2を除いて、すべてA安定な計算法である。

この結果を見ると、陰的Runge-Kutta方法と同じ次数のPadé展開法[2/2]は、あまり良い結果とはいえない。しかしながら、この方法はA安定な計算法なので、A安定でないJain2の方法のように発散することはない。前の例題で使ったPadé展開法[9/11]は、他のA安定な陰的Runge-Kutta法と同じ程度の誤差で、前の例題のように、高精度の結果は得られない。このため、次数をさらに上げて[15/15]程度の計算も行ってみたが、ほぼ同じような結果であった。これは、Padé展開を構成している分子、分母の展開式の収束が非常に遅く、かなり高次までの項まで計算する必要があるためであると考えられる。

6 まとめ

1 変数の関数を、数値的にTaylor展開する方法を述べた。この方法は、C++言語を使うと、その関数の数値計算用プログラムの宣言部分の変更程度で、容易にTaylor展開できる。

これを利用すれば、常微分方程式の解を任意の次数まで、Taylor展開できる。解がTaylor展開であるため、その扱いは非常に容易であり、計算結果の誤差評価、許容誤差を与えたときの最良の刻み幅の決定などに利用できる。さらに、このTaylor展開をPadé展開することによって、常微分方程式の初期値問題を解く、任意次数のA安定な公式を得ることができる。

本論文では、他の方法との計算効率については、述べなかった。これは、Taylor展開の計算が、問題によって、大きく計算時間が異なるためである。また、この高次の計算法は、高次のRunge-Kutta公式と同様に、1ステップの計算時間が大きくなっ

ても、計算ステップが大きくとれる可能性があるため、全体として計算効率が良い場合も考えられるためである。このような詳細な議論は、今後の課題である。

参考文献

- [1] Corliss G. and Chang Y. F., Solving Ordinary Differential Equations Using Taylor Series, ACM Trans. Math. Soft., 8(1982), 114-144
- [2] Ellis M. A. and Stroustrup B., The Annotated C++ Reference Manual, Addison-Wesley, 1990
- [3] Hairer E., Wanner G., Solving Ordinary Differential Equations II, Springer-Verlag, 1991
- [4] Henrici, P., Applied Computational Complex Analysis, Vol. 1, John Wiley & Sons, New York, Chap. 1, 1974
- [5] 平山弘, C++言語によるべき型特異点をもつ関数の数値積分, 日本応用数理学会論文誌, 5(1995), 257-266
- [6] 伊理, 小野, 戸田, 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, 情報処理学会論文誌, 26(1986), 389-396
- [7] 久保田光一, コンピューティングの玉手箱 続・自動微分 Taylor展開, bit, 22(1991), 860-861
- [8] 三井斌友, 数値解析入門, 朝倉書店, 1985
- [9] 小野, 戸田, 伊理, 微分係数を用いた埋め込み型Runge-Kutta系2段公式について, 情報処理学会論文誌, 28(1987), 807-814
- [10] Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., Numerical Recipes in C, Cambridge University Press, 1988, Cambridge (邦訳: 丹慶、奥村、佐藤、小林訳、技術評論社、1993)
- [11] Rall, L. B., Automatic Differentiation-Technique and Applications, Lecture Notes in Computer Science, Vol.120, Springer-Verlag, Berlin-Heidelberg-New York(1981)
- [12] 佐野理, キーポイント微分方程式, 岩波書店, 東京, 1993
- [13] 田中, 高山, 安定性のよい9段数7次陽的Runge-Kutta法について, 情報処理学会論文誌, 34(1993), 52-61
- [14] 田中, 高山, 2段数陰的Runge-Kutta法について, 情報処理学会論文誌, 36(1995), 226-234

Table 1. Relative errors in numerical solutions of $y' = -1/3x^2y^2$, $y(2) = 1$ h:step-size

h	Rel. error	[3/4]	[9/11]	Shanks	C&V7	L2	KN5	KN1
0.5	First-step	3.75d-03	2.35d-08	7.31d-04	9.74d-03	1.79d-06	9.63d-07	5.42d-06
	Last-step	4.44d-05	4.01d-12	1.41d-07	1.60d-06	1.08d-09	8.72d-10	1.49d-09
	Maximum	4.44d-03	1.52d-06	7.31d-04	9.74d-03	2.82d-06	2.06d-06	5.42d-06
0.1	First-step	9.74d-06	6.32d-16	3.02d-10	1.99d-09	5.35d-12	3.28d-12	1.28d-11
	Last-step	5.78d-06	6.67d-16	1.41d-11	8.51d-11	9.24d-13	8.06d-13	9.78d-13
	Maximum	3.57d-05	6.73d-12	7.46d-10	4.65d-09	3.32d-11	2.78d-11	4.14d-11

Table 2. Absolute errors in numerical solutions of $y' = 100(\sin x - y)$, $y(0) = 0$ h:step-size

h	Abs. Error	[2/2]	[9/11]	Butcher	Radau IIA	Jain2	Norsett 2
0.15	First-step	4.13d-02	1.34d-04	5.56d-03	8.44d-04	4.60d-02	5.45d-03
	Last-step	3.13d-02	5.34d-04	7.09d-05	1.29d-05	4.37d+65	1.19d-03
	Maximum	2.15d-01	2.23d-03	4.56d-03	8.44d-04	4.37d+65	5.45d-03