システム数理IV 第3回

牧野淳一郎

1998 年 4 月 27 日

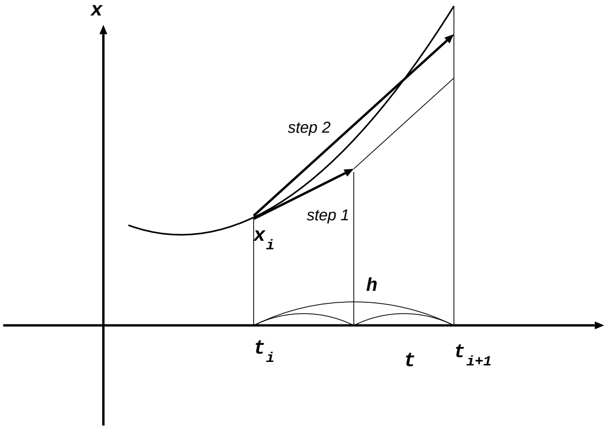
1 Runge-Kutta method

資料の番号と講義の進み方に若干ずれが生じているが、資料のほうは先に進めていくことにする。今回は、Runge-Kutta 法についてその考え方を説明していこう。

2 次の Runge-Kutta 法とは、以下のようなものであった。

$$\begin{aligned} k_1 &= \frac{h}{2} f(x_i, t_i) \\ x_{i+1} &= x_i + hf(x_i + k_1, t_i + h/2) \end{aligned} \tag{1}$$

これについては、図的な説明というものが可能である。元の点からまずオイラー法と同様に接線を引く。が、これを次の時刻まで延ばすのではなく、ステップの半分のところで止める。で、ここでもう一回微分方程式の右辺を評価する。ここでの導関数の値を使って、もとのところ (t_i, x_i) から直線を引くわけである。



実は、2 次精度を実現する RK 型の公式というのはこれ一つではない。以下のようなものも可能である。

$$\begin{aligned} k_1 &= hf(x_i, t_i) \\ k_2 &= hf(x_i + k_1, t_{i+1}) \\ x_{i+1} &= x_i + \frac{k_1 + k_2}{2} \end{aligned} \tag{2}$$

さらに、これらの公式を含む一般的な公式を与えることもできる。

今日は、まず、RK 法というものの一般的な定義を与え、それからどのようにして上の公式が得られるかを考える。また、そこからさらに高次（高精度）な公式の導出につなげていこう。

2 RK 法の一般形

最近は、一般形というと以下の形に書かれることが多い。

$$\begin{aligned} x_{n+1} &= x_n + h\sum_{i=1}^{s} b_i k_i \\ k_i &= f(x_n + h\sum_{j=1}^{s} a_{ij}k_j, t_n + c_i h) \end{aligned} \tag{3}$$

自然数 s を段数（number of stages）という。a_{ij}, b_i, c_i はパラメータであるが、a と c は普通

$$c_i = \sum_{j=1}^{s} a_{ij} \tag{4}$$

となるようにとる。これは、一般にそうでないような公式は不可能ではないがあまりいいことがない（精度がよくならない）からである。

と、こう、式に書いてしまうとすぐにはわからないが、例えば $s = 2$ の場合に書き下してみると

$$\begin{aligned} x_{n+1} &= x_n + h(k_1 b_1 + k_2 b_2) \\ k_1 &= f(x_n + h(a_{11}k_1 + a_{12}k_2), t_n + c_1 h) \\ k_2 &= f(x_n + h(a_{21}k_1 + a_{22}k_2), t_n + c_2 h) \end{aligned} \tag{5}$$

と、まあ、こんな感じになる。こちらを良く見ればすぐにわかるように、

1. a_{ij} $(j \geq i)$ がすべて 0 ならば、k_1から順に計算していくことができる。つまり、「陽的」公式になっている。
2. a_{ij} $(j > i)$ が 0 のときは、各 k_i についての式に k_i だけが入ってくる。これを半陰的 (semi-implicit) 公式という。この場合には、まず k_1 についての方程式をといて、次に k_2 についてのを解いて、、、と順番に計算出来る。
3. 上のような制約が全くない時は、「陰的」公式ということになる。このときは、すべての k_i に対する（一般には非線形な）方程式を一度に解く必要がある。

前の後退オイラー法の場合と同様、なぜ陰的公式といった面倒なものをつくるかというのはよくわからないかもしれない。これについてはあとで説明するとして、まず、陽的公式の例について述べる。

3 古典的 Runge-Kutta 法

ルンゲ・クッタ法のなかでもっとも広く使われているのが「古典的」といわれる公式である。書き下すと、

$$x_{n+1} = x_n + h(k_1/6 + k_2/3 + k_3/3 + k_4/6)$$

$$\begin{aligned} k_1 &= f(x_n, t_n) \\ k_2 &= f(x_n + hk_1/2, t_n + h/2) \\ k_3 &= f(x_n + hk_2/2, t_n + h/2) \\ k_4 &= f(x_n + hk_3, t_n + h) \end{aligned} \tag{6}$$

というものである。これは、いろいろ良い性質をもつ。例えば

1. a_{ij} が $i-j=1$ 以外すべて 0 なので、右辺の計算が楽である。
2. 次数が 4 次であり、4 段陽的公式で到達可能な最高次数を達成している
3. 係数が簡単な有理数なので、プログラムしやすい。また丸め誤差を小さくできる。

この公式が 4 次であることを示すのは、それほど簡単ではない。腕力に自信があるひとは挑戦してみて欲しい。

4 陽公式の到達可能次数

理論的、あるいは実用上重要なのは、段数を決めた時に、次数がいくつまで可能かということである。が、これは一般の場合には未解決であり、以下の結果しかわかっていない

s	1	2	3	4	5	6	7	8	9	10	$s \geq 10$
p	1	2	3	4	4	5	6	6	7	7	$p \leq s-2$

ある段数で、到達可能な最高次数を達成する公式というのは一通りではない。2 段 2 次の場合でも、自由パラメータが一つあって「無限個」の公式があり得るわけである。段数が増えれば自由パラメータの数も増える。

それではどういう公式を使うべきかというのは、精力的な研究が行なわれてきた。問題にもよるが、例えば

1. 局所離散化誤差の係数が小さい
2. 係数の数字が簡単な有理数になる
3. 係数に 0 が多い

というようなことはそれぞれ実用的には重要である。そのような条件を満たす公式が多数提案されている。

Dormand と Prince は、最近、段数 / 次数のさまざまな組合せについて、局所離散化誤差を非常に小さくした公式を求めた。

5 刻み幅調節と埋め込み型公式

さて、高次の公式を使うにしても、実際に問題を解こうという上ではいろいろ考えないといけないことがある。もっとも重要なことは、刻み幅 h をどうやって決めるかということである。

前にやった大域誤差解析では、ある範囲で導関数（高次の公式なら一般に高次の導関数）が最大値を持てば、その最大値と刻み幅の関数で数値解の誤差が押えられるという話をした。が、実際問題としては、解いていくにつれて導関数の値が大きく変わるといったことは普通に起きる。このため

に、刻み幅が一定のままで計算していくというのは非常に無駄なことが多い。

無駄を減らすためには、局所誤差が小さい（解が滑らかな）ところは刻みを大きく、逆に解が急にかわるところでは刻みを小さくしてやればよい。このような方法を可変刻み (variable step)、あるいは適応刻み (adaptive step) という。

例えば Runge-Kutta 法の場合、x_n から x_{n+1} にいくのに前に計算した情報とかなにかを使うわけではないので、刻み幅は任意にとることができる。問題は、どうやって刻み幅を決めるかである。

これにはいろいろな考え方があるが、通常使われるのは、局所離散化誤差を推定して、それがある値以下になるようにするという方法である。

このための一つの考え方は、以下のようなものである。

1. まず、ある幅 h で積分する。
2. 次に、その同じ幅を、$h/2$ ずつ 2 回に分けて積分する。
3. すると、2 回に分けて積分したほうがより正確な答になっていると期待できるので、二つの解の差が誤差の推定値（大きめではあるが）となっている。

誤差の推定値が求まったら、普通は次のステップを調節する。局所離散化誤差の次数がわかっているから、それに応じてステップサイズを調整すればよい。次数プログラムによっては、現在のステップでの誤差が予定した値よりも大きければステップを小さくして計算し直すようになっているものもある。

実装（プログラムを作ること）が容易であることもあって、この方法はわりと広く使われている。が、かなり無駄が多い方法であるというのも確かである。というのは、単に誤差の推定のためだけに、50% の無駄な計算をしているからである。

もうちょっとうまい方法はないかということで、以下のようなことを考えた人がいる。

一般に、RK 型の公式では、最終的な値は

$$x_{n+1} = x_n + \sum_{i=1}^{s} b_i k_i \tag{7}$$

の形になっている。ここで、k_i は全部そのまま使って、b_i を別の b'_i に置き換えた

$$x' = x_n + \sum_{i=1}^{s} b'_i k_i \tag{8}$$

で、局所離散化誤差が元の公式よりも大きいが、むやみと大きくはない（例えば、一次次数が低い）ものがあれば、元の公式との差を誤差の推定に使うことができる。

この形の公式のことを埋め込み型 embedded 公式という。最初に提案した人の名前をとって Runge-Kutta-Fehlberg 公式ということも多い。これもいろいろ提案されているが、上の Dormand-Prince 公式はすべて埋め込み型であり、これが最近はもっとも広く使われている。精度が高いものが必要な時は、10 段 8 次の公式が使われる。これらは

`http://www.unige.ch/math/folks/hairer/software.html`

から入手できる (Fotran と C がある)。

6　陰的公式

陰的公式については、以下の強力な結果が知られている：

s 段陰的公式の到達可能次数は $2s$ である。

これのため、いくらでも次数の高い公式をつくることができる。さらに、これは単なる存在証明ではなく、実際に $2s$ 次の公式が構築されている。これは実はユニークに決まり、「陰的ガウス公式」と呼ばれるものがそれである。以下、陰的ガウス公式と、それに関係したいくつかの公式についてその導出の原理と計算法を簡単に説明しよう。

6.1　ガウス型数値積分

ガウス型公式について述べる前に、まずそれと非常に関係の深い数値積分公式について説明しておく。

数値積分とは、一変数であれば

$$y = f(x) \tag{9}$$

と、微分方程式ではなく単に関数として与えられたものに対して、ある固定区間 (t_s, t_e) での定積分を数値的に求めることである。一般に、解析的に積分ができないようなものを計算しないといけないことはよくあるので、そういうものが必要になることは多い。

数値積分法として、すでに皆さんが知っているであろうものは、たとえば

- 台形公式
- シンプソン公式
- ロンバーグ加速

といったものであろう（知らない人は適当な数値計算の教科書で勉強したほうがいいかも）。なお、ロンバーグ加速については後でもう少し詳しく説明する。

例えば台形公式は、等間隔に区間を分けたとして、ある部分区間での積分を

$$s_i = \frac{h[f(x_i) + f(x_{i+1})]}{2} \tag{10}$$

と、両端を結んだ台形の面積で近似するということに相当する。これに対して、もっとましな方法というのを理論的に考えられないであろうか？

一つの方法は、区間の中の点をいくつかとって、それらを通る補間多項式を作り、それを積分することである。中の点を等間隔にとったものをニュートン・コーツ型の公式という。

さて、一般に補間するための点を p 個とった時、$p-1$ 次の多項式を持ってくればそのすべての点を通るように出来る。ニュートン・コーツ型の公式はそのような最小次数多項式を使うものである。例えばもとの関数が実際に $p-1$ 次以下の多項式ならば、厳密に正しいものが求まる。

この補間の原理を説明しておく。例えば補間に使う点が $(x_1, ...x_i, ...x_n)$ とあるとすれば、以下のような多項式を考える

$$p_i(t) = \frac{(x - x_1)(x - x_2)...(x - x_{i-1})(x - xi + 1)...(x_i - x_n)}{(x_i - x_1)(x_i - x_2)...(x_i - x_{i-1})(x_i - xi + 1)...(x_i - x_n)} \tag{11}$$

この多項式は、 $x = x_i$ で 1、それ以外のすべての x_jで 0 になる。従って、補間多項式は

$$f_n(x) = \sum p_i(x)y_i \tag{12}$$

とすればいい。

数値積分に使うには、あらかじめ $p_i(t)$ の積分を求めておいて、それと y_iを掛けて和をとるだけである。等間隔でなく点をとることで精度をあげることはできないかというのが、ここでの問題である。

ここで、一般的に、不等間隔に点をとってなんとかするという観点から考えることもできるが、なかなか難しいので答のほうから考えていく。

答は、「ガウス・ルジャンドルの多項式の零点を点にとると、次数を最大にできる」というものである。ガウス・ルジャンドルの多項式は、有限区間で帰納的に定義できる直交関数系である。つまり、最初の項を定数として、k 番目の項は k 次多項式であって、それまでの項すべてと直交するようにとるのである。つまり、関数基底 $(1, x, x^2, ...x^k..)$ にグラム・シュミットの直交化を適用したものがガウス・ルジャンドルの多項式である。

だいぶ疲れたので、今回の資料はここまで。次は、直交多項式をつかった積分公式と、数値積分と陰的ルンゲ・クッタ公式の関係について。