

Møller法と Gill法について

7L-2

幸谷智紀 永坂秀子

石川職業能力開発短期大学校 日本大学理工学部

と、表形式で表わす。この表中に現れない a_{ij} ($i \leq j$) は、すべて 0 として考える。

1 初めに

1951年, Gill は [1] において, 係数の記憶領域を減らし, 合わせて計算過程の丸め誤差を補正する 4 段 4 次陽 Runge-Kutta 法を提案した。この講演では, そのうち補正法のみを取り扱うことにする。

その後, Møller は更に簡易な丸め誤差補正法を考案した。その手法は Gill のアナロジーだが, 計算回数はずっと少なく済む。

この 2 つの丸め誤差補正法を一般の m 段階 Runge-Kutta 法に適用可能にしたものを, それぞれ Gill 法, Møller 法と呼ぶことにしよう。アルゴリズムの詳細は次節で述べる。

ここで, 以降で扱う常微分方程式と, m 段階 Runge-Kutta 法について簡単に述べる。

対象となる 1 階常微分方程式の初期値問題を

$$(1.1) \quad \frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

とする。このとき, (1.1) の $x = \alpha$ における数値解を得るものとする。これを m 段階 Runge-Kutta 法を用いて解くには, 区間 $[x_0, \alpha]$ を刻み幅 $h = (\alpha - x_0)/n$ で n 分割し, Algorithm 1 のように計算すればよい。

Algorithm 1 (m 段階 Runge-Kutta 法)

1. For $i = 0, 1, \dots, n-1$

$$(1) \quad k_1 := f(x_i, y_i)$$

$$(2) \quad k_2 := f(x_i + c_2 h, y_i + h \cdot a_{21} k_1)$$

$$(3) \quad \vdots$$

$$(4) \quad k_m := f(x_i + c_m h, y_i + h \cdot \sum_{j=1}^{m-1} a_{mj} k_j)$$

$$(5) \quad y_{i+1} := y_i + h \cdot \sum_{j=1}^m w_j k_j$$

Algorithm 1 で定数 $c_2, c_3, \dots, c_m, a_{21}, a_{31}, a_{32}, \dots, a_{m,m-1}, w_1, w_2, \dots, w_m$ を陽 Runge-Kutta 法の「係数」と呼び,

$$(1.2) \quad \begin{array}{c|cccccc} & a_{21} & & & & \\ c_2 & a_{31} & a_{32} & & & \\ c_3 & \vdots & \vdots & \ddots & & \\ \vdots & a_{m1} & a_{m2} & \cdots & a_{m,m-1} & \\ c_m & w_1 & w_2 & \cdots & w_{m-1} & w_m \end{array}$$

On Møller and Gill Methods

Tomonori Kouya Hideko Nagasaka

Ishikawa Polytechnic College Department of Science and Technology, Nihon University

2 Gill 法と Møller 法

Algorithm 1 から分かるように, 関数計算を除いた大部分が

$$(2.3) \quad C_i := A_i + h \cdot B_i \quad (i = 0, 1, \dots)$$

という形をしていることがわかる。このうち, $h \cdot B_i$ を「修正量」と呼ぶことにする。

一般に, 刻み幅 h は小さくしないと十分な精度を得ることは難しい。そのため, 修正量は A_i に比べて絶対値が小さくなり, 有限桁の浮動小数点計算では $h \cdot B_i$ の下位桁が積み残されてしまう。この現象を「情報落ち」という。情報落ちで加算されなかった分が, その計算で発生した丸め誤差となる。

この情報落ちを根本から解決するには, 浮動小数点数の桁数を増やす以外にない。しかし, 次のように計算方法を変更することで, 同じ桁数でも情報落ちによる丸め誤差をある程度補正することが可能となる。

Algorithm 2 (丸め誤差補正法)

1. $q := 0$

2. For $i = 0, 1, \dots$

$$(1) \quad t := h \cdot B_i$$

$$(2) \quad s := t - q$$

$$(3) \quad C_i := A_i + s$$

$$(4) \quad r := C_i - A_i$$

$$(5) \quad q := r - s$$

q に情報落ちした量を足しこんでいき, それが修正量に加算されうる大きさになれば, それだけ丸め誤差が補正される仕組みである。

Gill 法と Møller 法は, この Algorithm 2 を Algorithm 1 のどの部分に適用するかでその違いが出る。

Algorithm 3 (Gill 法)

1. $q := 0$
2. For $i = 0, 1, 2, \dots$
 - (a) $k_1 := f(x_i, y_i)$
 - (b) $t := h \cdot a_{21} k_1$
 - (c) $s := t - q$
 - (d) $y_i^{(2)} := y_i + s$
 - (e) $k_2 := f(x_i + c_2 h, y_i^{(2)})$
 - (f) $r := y_i^{(2)} - y_i$
 - (g) $q := r - s$
 - (h) for $l = 3, 4, \dots, m$
 - i. $t := h \cdot \sum_{j=1}^{l-1} (a_{lj} - a_{l-1,j}) k_j$
 - ii. $s := t - q$
 - iii. $y_i^{(l)} := y_i^{(l-1)} + s$
 - iv. $k_l := f(x_i + c_l h, y_i^{(l)})$
 - v. $r := y_i^{(l)} - y_i^{(l-1)}$
 - vi. $q := r - s$
 - (i) $t := h \cdot \sum_{i=1}^m (w_i - a_{mi}) k_i$
 - (j) $s := t - q$
 - (k) $y_{i+1} := y_i^{(m)} + s$
 - (l) $r := y_{i+1} - y_i^{(m)}$
 - (m) $q := r - s$

Algorithm 4 (Møller 法)

1. $q := 0$
2. For $i = 0, 1, 2, \dots$
 - (a) $k_1 := f(x_i, y_i)$
 - (b) $k_2 := f(x_i + c_2 h, y_i + h \cdot a_{21} k_1)$
 - (c) $\vdots$
 - (d) $k_m := f(x_i + c_m h, y_i + h \cdot \sum_{i=1}^{m-1} a_{mi} k_i)$
 - (e) $t := h \cdot \sum_{i=1}^m w_i k_i$
 - (f) $s := t - q$
 - (g) $y_{i+1} := y_i + s$
 - (h) $r := y_{i+1} - y_i$
 - (i) $q := r - s$

Møller 法 (Algorithm 4), Gill 法 (Algorithm 3) の相違点は以下の通りである。

Møller 法 — y_{i+1} の計算で発生する丸め誤差のみを補正する

Gill 法 — 各 k_l ($l = 2, 3, \dots, m$) の $y_i^{(l)}$ と y_{i+1} の

計算全てに補正を行う

特に Gill 法は, $k_1, k_2, \dots, k_m, y_{i+1}$ の計算全てに丸め誤差の補正を行うため, 前段階の計算量に修正量を足しこむという Algorithm 2 を適用するには, 係数 (1.2) を次のように変更する必要がある。

$$(2.4) \quad \begin{array}{c|cccc} c_2 & a_{21} & & & \\ c_3 & a_{31} - a_{21} & & & \\ \vdots & \vdots & & \ddots & \\ c_m & a_{m1} - a_{m-1,1} & \cdots & a_{m,m-1} & \\ \hline & w_1 - a_{m1} & \cdots & w_{m-1} - a_{m,m-1} & w_m \end{array}$$

これは計算回数を増やさないためにも, 事前に求めておくことが望ましい。

3 数値実験

本講演では, 様々な計算機環境でこの 2 つの丸め誤差補正法で計算し, その結果について報告する。テスト問題は以下の通り。

1. $dy/dx = -y, y(0) = 1$
解析解: $y(x) = \exp(-x)$ 積分区間: $[0, 1]$
2. $dy/dx = 100(\sin x - y), y(0) = 0$
解析解: $y(x) = (\sin x - 0.01(\cos x + \exp(-100x)))/1.0001$
積分区間: $[0, 1]$

参考文献

- [1] S. Gill: A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine, Proceedings of the Cambridge Philophical Society, Vol. 47, 1951, pp. 96 - 108.
- [2] 伊理正夫・松谷泰行: Runge-Kutta-Gill 法について, 情報処理, Vol. 8, No. 2, 1967, pp. 103 - 107.
- [3] T. Kouya: The Comparison between the Møller and the Gill Methods for Explicit Runge-Kutta Process, (投稿中).
- [4] O. Møller: Quasi Double-Precision in Floating Point Addision, BIT 5, 1965, pp. 37 - 50.
- [5] O. Møller: Note on Quasi Double-Precision, BIT 5, 1965, pp. 251 - 255.
- [6] 永坂秀子: 計算機と数値解析, 朝倉書店, 1980, pp.215 - 223.
- [7] 永坂秀子・幸谷智紀: Runge-Kutta-Gill 法, Møller 法と Optimizing Compiler について, 第 37 回日本大学理工学部学術講演会予稿集, 1993/11/19.
- [8] 清水辰次郎・大橋常道: Runge-Kutta-Gill タイプの数値解法について I, II, 日本数学会 応用数学分科会講演予稿集, 1975/4, pp. 101 - 111.