計算物理学:No.3

1　効率の良いプログラミングについて

正しく動くプログラムを作るには、一気にプログラム全体を書き上げるのではなく、比較的小さな (半ば独立した) 単位に分けて、各単位が正しく動くことを確かめた上で次のステップに進むと良い。このようにすると、問題を発見し、修正し易くなる。

このためにはサブルーチンや副プログラムを使うと有効であるので、復習しておくこと。プログラムはしばしば再利用するので、そのことを意識した書き方をしておくと良い (さらに、ファイルのコピー法、ディレクトリーの使い方に馴染んでおくと効率的に作業できる)。各サブルーチン間で変数だけでなく配列の受渡しをすることができる。

2　収束加速法：Aitken 法

n 個の数列に対する Aitken 加速のサブルーチンを使って、次のような数列を加速してみる。

```
c     MAIN ROUTINE:EXAMPLE
c
      implicit real*8 (a-h,o-z)
      dimension a(20),b(20),c(20)
      integer i,n
c
      n=10
      do i=1,n
         b(i)=0.0d0
         c(i)=0.0d0
      end do
      do i=1,n
         a(i)= 0.7 + 0.5*(0.9)**i + 0.2*(0.38)**i
      end do
      call Aitken(a,b,n)
      call Aitken(b,c,n-2)
      do i=1,n
```

```
         write(*,1000) i,a(i),b(i),c(i)
      end do
 1000 format(i4,3e18.10)
      end
```

これは、2 段の Aitken を加速した例である (入出力の format 文の書式も調べること)。

3 交代数列と Richardson 加速

混合型の収束列

$$a_n = a + \lambda^n (c_1 n^{-\alpha_1} + c_2 n^{-\alpha_2} + \cdots) \tag{1}$$

で λ が既知の場合、Richardson 加速が実行でき、

$$a_n^{(1)} \approx a + \lambda^n \frac{c_1(-\alpha_1)}{(1-\lambda)} n^{\alpha_1 - 1} \tag{2}$$

となる (これを確かめてみよ) [1] 。従って再び混合型となり、Richardson 加速を繰返し適応できる。

交代数列は混合型の $\lambda = -1$ の場合にあたるので、交代数列に対し $\lambda = -1$ とした Richardson 加速も実行できる。余裕があれば、レポート課題の交代数列にたいして、Richardson 加速を試してみよ。

4 その他の収束加速法

Richardson,Aitken 加速の他の収束加速法には、ϵ-algorithm, Levin 変換等がある [2, 3]。なお、単純な冪乗的振舞いをする数列

$$\lim_{n\to\infty} \frac{a_{n+1} - a}{a_n - a} = 1 \tag{3}$$

(対数収束列) には、一般的な収束加速法がないことが証明されている [4, 5]。

4.1 ϵ-algorithm

ϵ-algorithm の手続きは、以下の通りである。

1) 2 次元配列 ϵ を用意し、その第 0 行に元の配列を入れる。

$$\epsilon_{0,n} = a_n \qquad (n = 1, 2, \cdots) \tag{4}$$

2) 隣の要素との差の逆数を次の行に書く。

$$\epsilon_{1,n} = 1/(\epsilon_{0,n+1} - \epsilon_{0,n}) \qquad (n = 1, 2, \cdots) \tag{5}$$

3) $i = 1, 2, 3, \cdots$ の順に

$$\epsilon_{i+1,n} = \epsilon_{i-1,n+1} + 1/(\epsilon_{i,n+1} - \epsilon_{i,n}) \qquad (n = 1, 2, \cdots) \tag{6}$$

このようにして得られた $\epsilon_{2,n}, \epsilon_{4,n}, \epsilon_{6,n}, \cdots$ が収束加速列である。

ϵ-algorithm は特に

$$a_n = a + \sum_{r=1}^{k} b_r(n)\lambda_r^n \qquad (b_r(n) \text{ は } n \text{ の多項式}) \tag{7}$$

という場合には、$\epsilon_{2K,n}$ $(K = \sum_{r=1}^{k}(b_r(n) \text{ の次数}))$ が正しい収束値 a を与える。

参考文献

[1] 数値計算法の数理、杉山 正顕、室田 一雄 共著、(岩波書店) 第 12 章 4 節

[2] 数値計算の常識、伊理 正夫、藤野 和健 著、(共立出版) 20 章

[3] “Sequence Transformations and Their Applications”, J. Wimp (Academic Press, New York 1981)

[4] 数値計算法の数理、杉山 正顕、室田 一雄 共著、(岩波書店) 第 12 章 6 節

[5] “Sequence Transformations”, J. P. Delahaye, (Springer-Verlag, Berlin, 1988)