超入門・加速法

大阪大学理学部物理学科 3 年 大和 寛尚・学籍番号 04B21077

課題 1 自分のコンピュータに Julia をインストールし，Emacs に `julia-mode` を導入して Julia プログラミング環境を整備しました．また，Julia の基本的な文法を確認しました．

課題 2 Richardson 加速, Aitken 加速について学んだ．これらの加速法は対象の数列が

$$x_n = \alpha + c\gamma^n$$

のような形で書けるという仮定のもとで有効である．Richardson 加速法は，上の式において

$$\alpha \sim \frac{x_{n+1} - \gamma x_n}{1 - \gamma} \equiv y_n$$

と書けることを用いて収束先を予測する．Aitken 加速法は γ が未知の場合にも有効である．γ が

$$\gamma \sim \frac{x_{n+2} - x_{n-1}}{x_{n+1} - x_n}$$

と書けることを利用して y_n の定義式の γ にこれを代入して新たな数列 z_n を作り出す．

$$\alpha \sim x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} \equiv z_n$$

これを用いて収束先を予測することが可能になる．具体的な数列にこの加速法を適用してみる．まず，サンプル数列を次の数列とする．

$$a_n = 2.5 + \frac{1000}{\sqrt{n+1}} \cdot 0.9^n$$

言語 Julia を用いて本レポート末尾にあるように数列を生成できる．以降の実験の実際のプログラムのソースコードはすべてレポート末尾に記載した．また，GitHub からダウンロードすることも可能です．この数列は $n = 60$ で

$$a_{60} \sim 2.7300$$

程度であり，誤差は 1 割ほど．Richardson 加速法を適用すると

$$\alpha \sim 2.4809$$

程度であり，より精度が高い結果が得られる．同じく，Aitken 加速法を適用すると

$$\alpha \sim 2.5027$$

となる．Richardson 加速よりも精度が良い．Aitken 加速法を複数回適用することでさらに良い結果を得ることができる．いま，1 回だけ Aitken 加速法を適用して得られた数列にもう一度 Aitken 加速法を適用して

$$\alpha \sim 2.5000$$

というより精度の良い結果が得られる．

（次ページにソースコードを記す）

（ソースコード）

```
$ cat acceleration.jl
# richardson method
function richardson(gamma, orig, i)
    return (orig[i+1] - gamma * orig[i]) / (1 - gamma)
end

# aitken method
function aitken(orig, i)
    return orig[i] - ((orig[i+1] - orig[i]) ^ 2) /
                     (orig[i+2] - 2 * orig[i+1] + orig[i])
end

# generating sample sequense -- (*)
const a    = 2.5
const c    = 1000.0
const rate = 0.9
const n    = 60
seq = [a + (c / sqrt(i + 1)) * (rate ^ i) for i = 1:n]

# applying richardson method to (*)
ric = [richardson(rate, seq, i) for i = 1:(n-1)]

# applying aitken method to (*)
ait = [aitken(seq, i) for i = 1:(n-2)]

# applying aitken method to (*) twice
cov = [aitken(ait, i) for i = 1:(n-4)]

println(ric, ait, cov)
```

以上のソースコードは GitHub にあります．次でダウンロード可能です．

```
$ wget https://raw.githubusercontent.com \
                        /hironaoy/numerical/main/acceleration.jl
```