

このようにしてFig. 3のように流れが求まると、次にこの流れの中での粒子の運動をEq.(3)を用いて求めることになる。Eq.(3)は2元2次の微分方程式に書き替えられるので前項で求めた流れの値を代入して、RKM法により積分して粒子軌跡を求める方法をとっている。

以上のようにして粒子の軌跡がわかると、捕集体の無限前方のどの領域に入った粒子が捕集体により捕集されるかを知ることができるので、これを種々の力の場、条件によりどのように変化するかを調べると理論的な捕集の機構を明らかに出来ることになる。

RKM: Runge Kutta Merson

測定値から未知パラメーターを決定する方法(II)

—変数探索法—

理学部・化学 須原正彦・村田重男

今回は、非線形最適化の方法の内、我々が用いている方法の概要を説明したが、今回は、個々の場合の具体的な例として「①直接探索法」の内、最も簡単な最小化問題、すなわち一変数探索のプログラムを紹介することにする。多変数関数の最小化問題に用いられている手法は、独立変数が構成する空間のそれぞれの方向に沿って順次探索するので、それぞれの線形探索は一変数探索と等しいため、一変数探索は最小化問題の基本となる。

一変数探索の方法は二つに分類される。

- (1)最小点が存在する区間を求める方法 (2)最小点に近接した点を求める方法

(1)の方法を適用する場合、最小点の存在がわかっている区間が示され、関数はこの区間内で唯一の積値をもっていることを仮定する。最小点の位置が要求される精度で求まるまで、最初に示された区間を種々な方法で短縮していく。(2)の方法を用いるには、最小点に近接した初期点をまず決める。多数の点に当てはまる低次の多項式を定め、つぎに当てはめた関数の最小点を求めると云う手順を、要求されている精度で最小点が求まるまで繰り返す。

我々は(1)に対しては黄金分割による探索法、(2)に対しては二次補間法による探索法を選んだ。

(1) 黄金分割による探索法

探索の各ステップごとに、最小点を含む区間の幅を一定の比率で減らしていく。

求めようとする最小点は、区間 (x_1, x_2) 内に存在し、 $x_1 < x_3 < x_4 < x_2$ を満たすように2点 x_3, x_4 をこの区間内から選ぶ。 i 回目の反復において

$$x_3^{(i)} = \frac{\tau-1}{\tau}(x_2^{(i)} - x_1^{(i)}) + x_1^{(i)}, \quad x_4^{(i)} = \frac{1}{\tau}(x_2^{(i)} - x_1^{(i)}) + x_1^{(i)}$$

一方、探索においては $x_4^{(i)} - x_1^{(i)} = x_2^{(i)} - x_3^{(i)} = (x_2^{(i)} - x_1^{(i)})/\tau$ を満たさねばならぬ。 i 回目の反復の結果、最小点を含む区間は $(x_1^{(i)}, x_4^{(i)})$ であったと仮定すると、

$$x_1^{(i+1)} = x_1^{(i)}, \quad x_2^{(i+1)} = x_4^{(i)}, \quad x_4^{(i+1)} = x_3^{(i)}, \quad \text{となる。}$$

$(i+1)$ 回目ののち最小点を含む区間が $(x_1^{(i+1)}, x_4^{(i+1)}), (x_3^{(i+1)}, x_2^{(i+1)})$ にかかわらず

$$(x_4^{(i+1)} - x_1^{(i+1)}) + \frac{1}{\tau}(x_2^{(i)} - x_1^{(i)}) = (x_2^{(i)} - x_1^{(i)}) \text{ となり } (i-1) \text{ 回目の反復で求めた}$$

区間の長さは i 回目と $(i+1)$ 回目での長さの和に等しくなる。従って、 $(i-1)$ 回目の反復後の区間は“黄金分割”にしたがって分けられていることになる。

区間短縮係数を一定とするには $\tau = (1 + \sqrt{5}) / 2 \approx 0.618$ となる。

このアルゴリズムは

- (i) $f(x_4^{(i)}) > f(x_3^{(i)})$ のとき $x_2^{(i+1)} = x_4^{(i)}, x_4^{(i+1)} = x_3^{(i)},$
 $x_3^{(i+1)} = x_1^{(i)} + (1 - \tau)(x_2^{(i+1)} - x_1^{(i)})$
- (ii) $f(x_4^{(i)}) \leq f(x_3^{(i)})$ のとき $x_1^{(i+1)} = x_3^{(i)}, x_3^{(i+1)} = x_4^{(i)},$
 $x_4^{(i+1)} = x_2^{(i)} - (1 - \tau)(x_2^{(i)} - x_1^{(i+1)})$

であり、 n 回だけ関数値を評価すれば、区間幅は $(x_2^{(1)} - x_1^{(1)})\tau^{n-1}$ となる。具体的なプログラムシートを図 1 に示す。(プログラム名 GLSCT)。

(2) 一変数探索のための 2 次補間法の応用

上に述べた方法は実行が容易で、収束の割合が保証されている長所をもつが、精度をあげるには収束の速度はかなりおそくなる。(2)の方法は、関数はその最小点の近傍では、2 次の補間多項式で十分近似できるという仮定——最適化問題において最も多く用いられ、十分満足できるものであることが確かめられており、より高度の探索方法の発展に重要な key point となっている仮定——を用いている。

我々の用いたアルゴリズムは

- (i) $f(x^{(0)})$ と $f(x^{(0)} + h)$ を求める。
- (ii) $f(x^{(0)} + h) \leq f(x^{(0)})$ であれば、 $x^{(0)} + 2h, x^{(0)} + 4h, \dots, x^{(0)} + 2^i h$ と求めていき、 $f(x^{(0)} + 2^{(i-1)} h) < f(x^{(0)} + 2^i h)$ となるまでつづける。
- (iii) $f(x^{(0)} + h) > f(x^{(0)})$ であれば、 $x^{(0)} - h, x^{(0)} - 2h, \dots, x^{(0)} - 2^i h$ と求めていき、 $f(x^{(0)} - 2^{(i-1)} h) < f(x^{(0)} - 2^i h)$ になるまでつづける。
- (iv) こうして得られた最後の 3 点が囲みを形成する。
- (v) 最小値を推定する前に、さらに最後の区間を 2 等分し、もとの囲みとこの新しい点とから等距離 (L) にある 3 点より構成される新しい囲みをつくる。求まった点を $x_1, x_2, x_3 (x_1 < x_2 < x_3)$ とすると最小点は

$$x_4 = x_2 - \frac{L}{2} \frac{f(x_3) - f(x_1)}{f(x_3) - 2f(x_2) + f(x_1)}$$

- (vi) この点をつぎの新しい囲みをつくるために用いる。新しい囲みの点を等距離にするためのもう一つの点で関数値を評価する。

以上の手続きを繰り返す。プログラムは図 2 に示す。(プログラム名 MINHOK) いずれのプログラムも充分用いるにたるものであることが、種々のテストで確かめられた。GLSCT では COMMON / MGIL / A, B, SPA により探索の区間 (A, B) および収束条件 (許容区間誤差 SPA) を与えることにより最小値の存在する区間 (AA, BB) が求まると同時に印字される。一方 MINHOK では同じ COMMON を用い最小点の近傍と考えられる点の値 (初期値 A), 探索の間隔 (B) および収束条件 (許容誤差 SPA) を与えれば、最小値が望む精度で得られ印字される。

```

ISN      *      I      SOURCE STATEMENT / ERROR MESSAGE                                *ID. SEQ.
1          SUBROUTINE GLSCT                                                              GLS00010
2          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                          GLS00020
3          COMMON /MUL/A,B,SPA                                                            GLS00030
4          600 FORMAT(1H0,T0,'GLSCT STARTING DATA'//11,'INITIAL(A)= 'T15,E13.6,
1J0,'INITIAL(B)= 'T45,E13.6,T00,'FINISH SPACE='T75,E13.6)
5          602 FORMAT(1H0,1UX,'MINIMUM WA 'E13.6,' TO 'E13.6,' NU AIUA NI ARU--UGLS00060
100NBUI,KAZUHO NI YURU')
6          604 FORMAT(1H0,' SUBROUTINE GLSCT ERK NO TAME NI THUSHISHIMASU') GLS00080
7          700 WRITE(6,600,END=710,ERR=710)A,B,SPA
8          1AU=(15NHT(5.0DG)-1.0)/7.0
9          710 A=C
10         X1=A+(1.0-TAU)*(0-A)
11         X2=B-(1.0-TAU)*(B-A)
12         AA=A
13         BB=B
14         200 IF (F(X2)-F(X1))100,100,120
15         100 AA=X1
16         X1=X2
17         X2=BB-(1.0-TAU)*(BB-AA)
18         N=N+1
19         GO TO 150
20         120 BB=X2
21         X2=X1
22         X1=AA+(1.0-TAU)*(BB-AA)
23         N=N+1
24         150 IF (MOD(N,5).EQ.4) IF (SPA=BB+AA)200,210,210
25         GO TO 200
26         210 WRITE(6,602,END=720,ERR=800)AA,BB
27         GO TO 720
28         800 WRITE(6,604,ERR=720)
29         720 RETURN
30         END
    
```

*** OPTIONS IN EFFECT ***

FORTRAN2 LIST,NOMAP,NOASTER,NODEBUG,NOSEQ,NOHEW,SYSLN,DECK,NOSTACK,EBCDIC,
 THAP=00CE,OPT=1,WORK=20,ARRAY=256,TABLE=10

FT997 END OF COMPIATION (GLSCT)

```

ISN      *      I      SOURCE STATEMENT / ERROR MESSAGE                                *ID. SEQ.
1          SUBROUTINE MINHOX                                                              MIN00010
2          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                          MIN00020
3          COMMON /MUL/XX0,HC,C                                                         MIN00030
4          600 FORMAT(1H0//110,'INITIAL VALUE OF X'T30,E13.6//T10,'SPACE OF TANS'
1A'T30,E13.6//T10,'TE[SHI JOKEN PARAMETER'T35,E13.6)
5          602 FORMAT(1H0,1UX,'MINIMUM WA 'E13.6,' DE AKU--NIJHOKANHO NO OUYU')
6          604 FORMAT(1H0,'SUBROUTINE MINHOX ERK NO TAME NI THUSHISHIMASU')
7          700 WRITE(6,600,END=710,ERR=800)XX0,H0,C
8          X0=XX0
9          H=H0
10         710 N=1
11         IF (F(X0)-F(XU+H))100,110,110
12         N=N+1
13         X1=XU+H*2.0**N
14         X2=XU+H*2.0**(N+1)
15         IF (F(X1)-F(X2))200,110,110
16         200 XU=XU+H*2.0**(N-1)
17         XL=(X1+X2)/2.0
18         AL=H*2.0**(N-1)
19         IF (F(X1)-F(XA))310,310,300
20         300 XJ=X2
21         X2=XA
22         GO TO 400
23         310 X3=XA
24         X2=X1
25         X1=XJ
26         GO TO 400
27         100 IF (F(X0)-F(XU+H))210,210,220
28         210 A1=XU+H
29         X2=XU
30         XJ=XU+H
31         GO TO 400
32         220 N=N+1
33         A1=XU+H*2.0**N
34         X2=XU+H*2.0**(N+1)
35         IF (F(X1)-F(X2))230,230,220
36         230 XU=XU+H*2.0**(N-1)
37         XL=(X1+X2)/2.0
38         AL=H*2.0**(N-1)
39         IF (F(X1)-F(XA))330,330,320
40         320 XJ=X1
41         X1=X2
42         X2=XA
43         GO TO 400
44         330 X2=X1
45         X1=XX
46         X3=XU
47         400 X0=X2-XL/2.0*(F(X3)-F(X1))/(F(X3)-2.0*(X2)+F(X1))
48         H=DABS(X0-X2)
49         IF (H=C)720,720,710
50         720 WRITE(6,602,END=730,ERR=800)XU
51         GO TO 730
52         800 WRITE(6,604,ERR=730)
53         730 RETURN
54         END
    
```