

# 数値計算の基礎Ⅱ

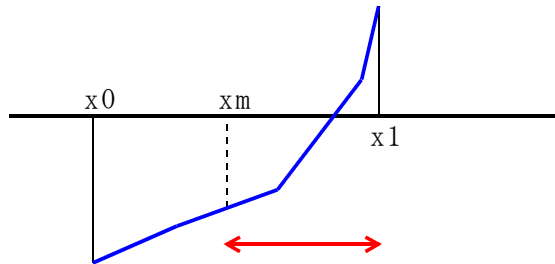
## 0. 目次

1. 2分法
2. はさみうち法
3. 割線法
4. 割線法（2次曲線近似）
5. ニュートン法（接線近似）

## 1. 2分法

区間  $[x_0, x_1]$  にある関数  $f(x)$  の根を求める。

区間  $[x_0, x_1]$  を  $x_m = (x_0 + x_1) / 2$  で 2 等分し、区間  $[x_0, x_m]$ ,  $[x_m, x_1]$  に分割する。 $f(x_m)$  の絶対値が十分小さい値  $\text{eps}$  より小さいとき、 $x_m$  を根とする。そうでないとき、分割された 2 つの区間の内、根の存在する区間に対して、同様の操作を繰り返す。



### ● プログラム (rf111.c)

$f(x) = x^3 - a$  の根を求める。

```

1  /* << rf111.c >> */
2  /* 2分法。*/
3  #include <stdio.h>
4  #include <math.h>
5
6  int main() {
7      int n;          /* 繰り返し回数。*/
8      double a,      /* aの値。*/
9                eps, /* 誤差。*/
10               fm,  /* f(xm)の値。*/
11               x0,  /* 区間の左端。*/
12               x1,  /* 区間の右端。*/
13               xm;  /* 中央値。xm=(x0+x1)/2。*/
14
15     /* データ入力 (例 a=8.0)。*/
16     scanf("%lf",&a);
17     printf("%lf の立方根    ",a);
18
19     /* 初期設定。*/
20     x0 = 0.0; x1 = 3.0;
21     eps = 1.0e-3;
22     printf("初期区間 [%6.2lf,%6.2lf]  eps = %6.2le¥n¥n", x0, x1, eps);
23     printf("      n                近似値");
24     printf("                関数值 ¥n");
25
26     /* 計算過程と途中経過出力。*/
27     n = 0;

```

```

28 while( 1 ) {
29     n++;
30
31     /* 近似値 xm を求める。*/
32     xm = (x0 + x1)/2;
33
34     /* 近似値に対応する関数値を求める。*/
35     fm = xm*xm*xm - a;
36     printf("%5d%24.16lf%24.16lf ¥n", n, xm, fm);
37
38     /* 解として出力。*/
39     if( fabs(fm) < eps ) {
40         printf("根 : %lf ¥n", xm); break;
41     }
42
43     /* 根の存在する区間 [x0, x1] を更新する。*/
44     if( fm > 0 ) {
45         x1 = xm;
46     } else {
47         x0 = xm;
48     }
49 }
50 }

```

## 実行結果

```

% cc rf111.c -lm
% a.out
8
8.000000 の立方根  初期区間 [ 0.00,  3.00]  eps = 1.00e-03

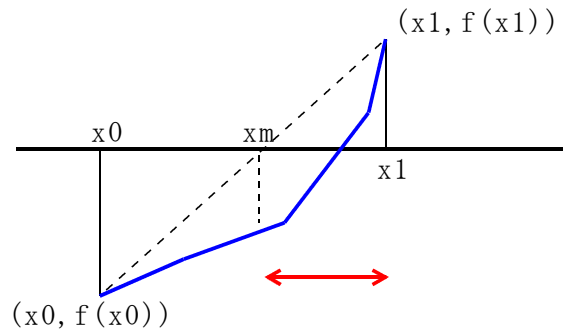
   n              近似値              関数値
  1      1.5000000000000000      -4.6250000000000000
  2      2.2500000000000000       3.3906250000000000
  3      1.8750000000000000      -1.4082031250000000
  4      2.0625000000000000       0.7736816406250000
  5      1.9687500000000000      -0.3691711425781250
  6      2.0156250000000000       0.1889686584472656
  7      1.9921875000000000      -0.0933842658996582
  8      2.0039062500000000       0.0469666123390198
  9      1.9980468750000000      -0.0234146192669868
 10      2.0009765625000000       0.0117244729772210
 11      1.9995117187500000      -0.0058579446049407
 12      2.0002441406250000       0.0029300451424206
 13      1.9998779296875000      -0.0014647543448518
 14      2.0000610351562500       0.0007324442269692
根 : 2.000061

```

## 2. はさみうち法

区間  $[x_0, x_1]$  にある関数  $f(x)$  の根を求める。

点  $(x_0, f(x_0))$  と点  $(x_1, f(x_1))$  を結ぶ直線と、 $x$  軸との交点  $(x_m, 0)$  で区間  $[x_0, x_1]$  を区間  $[x_0, x_m]$ ,  $[x_m, x_1]$  に分割する。  
 $f(x_m)$  の絶対値が十分小さい値  $\text{eps}$  より小さいとき、 $x_m$  を根とする。  
 そうでないとき、分割された2つの区間の内、根の存在する区間に対して、同様の操作を繰り返す。



### ●プログラム (rf121.c)

$f(x) = x^3 - a$  の根を求める。

```

1  /* << rf121.c >> */
2  /* はさみうち法。*/
3  #include <stdio.h>
4  #include <math.h>
5
6  int main() {
7      int n;      /* 繰り返し回数。*/
8      double a,  /* aの値。*/
9              eps, /* 誤差。*/
10             f0, /* f(x0)の値。*/
11             f1, /* f(x1)の値。*/
12             fm, /* f(xm)の値。*/
13             x0, /* 区間の左端。*/
14             x1, /* 区間の右端。*/
15             xm; /* x軸との交点。*/
16
17     /* データ入力 (例 a=3.0)。*/
18     scanf("%lf",&a);
19     printf("%lf の立方根    ",a);
20
21     /* 初期設定。*/
22     x0 = 0.0; x1 = 3.0;
23     eps = 1.0e-3;
24     printf("初期区間 [%6.2lf,%6.2lf]  eps = %6.2le¥n¥n",x0,x1,eps);
25     printf("      n                近似値");
26     printf("                関数値 ¥n");

```

```

27
28  /* 計算過程と途中経過出力。*/
29  n = 0;
30  f0 = x0*x0*x0 - a;
31  f1 = x1*x1*x1 - a;
32  while( 1 ) {
33      n++;
34
35      /* x軸との交点 xm を求める。*/
36      xm = x0 - f0*(x1-x0)/(f1-f0);
37
38      /* xm に対応する関数値を求める。*/
39      fm = xm*xm*xm - a;
40      printf("%5d%24.16lf%24.16lf ¥n", n, xm, fm);
41
42      /* 解として出力。*/
43      if( fabs(fm) < eps ) {
44          printf("根 : %lf ¥n", xm); break;
45      }
46
47      /* 根の存在する区間 [x0, x1] を求める。*/
48      if( fm > 0 ) {
49          x1 = xm; f1 = fm;
50      } else {
51          x0 = xm; f0 = fm;
52      }
53  }
54 }

```

## 実行結果

```

% cc rfl21.c -lm
% a.out
8
8.000000 の立方根  初期区間 [ 0.00,  3.00]  eps = 1.00e-03

   n          近似値          関数値
1    0.8888888888888888    -7.2976680384087791
2    1.4747274529236867    -4.7927316770215107
3    1.7819734703922894    -2.3414689644553830
4    1.9156086630638791    -0.9705656832068890
5    1.9683098752124191    -0.3742877386484231
6    1.9882408770124156    -0.1402814400263273
7    1.9956561821205037    -0.0520126839937651
8    1.9983980770838818    -0.0192076821620288
9    1.9994096045728158    -0.0070826539314401
10   1.9997824569261708    -0.0026102329463118
11   1.9999198486188070    -0.0009617780293674
根 : 1.999920

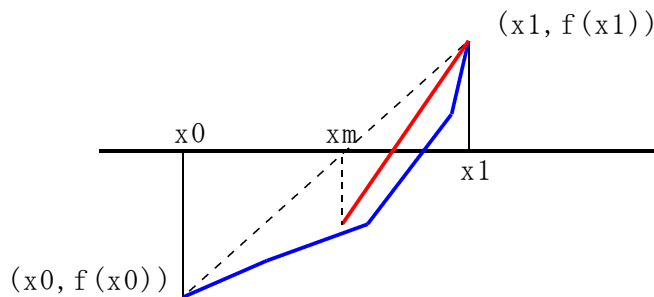
```

### 3. 割線法

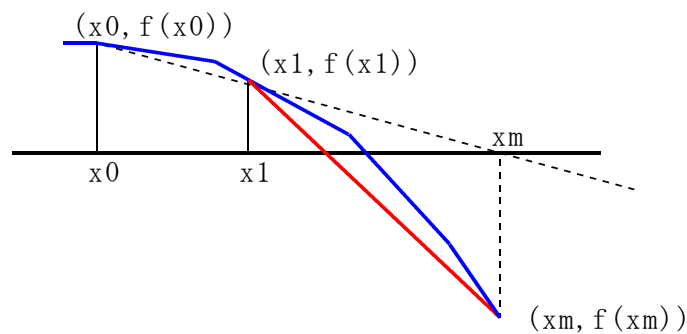
2点  $(x_0, f(x_0))$ ,  $(x_1, f(x_1))$  を結ぶ直線と  $X$  軸との交点  $(x_m, 0)$  を求め、根の近似値とする。

$f(x_m)$  の絶対値が十分小さい値  $\text{eps}$  より小さいとき、 $x_m$  を根とする。

そうでないとき、点  $(x_1, f(x_1))$  と点  $(x_m, f(x_m))$  を 2 点として同様の操作を繰り返す。



区間  $[x_0, x_1]$  に根が存在していない場合でも、自動的に根の存在する区間をさがしていくことができる。



#### ●プログラム (rf131.c)

$f(x) = x^3 - a$  の根を求める。

```

1 /* << rf131.c >> */
2 /* 割線法。*/
3 #include <stdio.h>
4 #include <math.h>
5
6 int main() {
7     int n;        /* 繰り返し回数。*/
8     double a,    /* aの値。*/
9             eps, /* 誤差。*/
10            f0,   /* f(x0)の値。*/
11            f1,   /* f(x1)の値。*/
12            fm,   /* f(xm)の値。*/
13            x0,   /* 区間の左端。*/
14            x1,   /* 区間の右端。*/
15            xm;   /* x軸との交点。*/

```

```
16
17  /* データ入力 (例 a=8.0)。*/
18  scanf("%lf",&a);
19  printf("%lf の立方根 ¥n¥n",a);
20
21  /* 初期設定。*/
22  x0 = 0.0; x1 = 3.0;
23  eps = 1.0e-3;
24  printf("初期区間 [%6.2lf,%6.2lf] eps = %6.2le ¥n¥n",x0,x1,eps);
25  printf("      n              近似値");
26  printf("              関数値 ¥n");
27
28  /* 計算過程と途中経過出力。*/
29  n = 0;
30
31  /* x0,x1の関数値を求める。*/
32  f0=x0*x0*x0 - a;
33  f1=x1*x1*x1 - a;
34  while( 1 ) {
35      n++;
36
37      /* x軸との交点 xm を求める。*/
38      xm = x0 - f0*(x1-x0)/(f1-f0);
39
40      /* xm に対応する関数値を求める。*/
41      fm = xm*xm*xm - a;
42      printf("%5d%24.16lf%24.16lf ¥n",n,xm,fm);
43
44      /* 解として出力。*/
45      if( fabs(fm) < eps ) {
46          printf("根 : %lf ¥n",xm); break;
47      }
48
49      /* 新たな区間 (根が存在しない場合もある) を求める。*/
50      x0 = x1; f0 = f1;
51      x1 = xm; f1 = fm;
52  }
53 }
```

## 実行結果

```
% cc rf131.c -lm
% a.out
8
8.000000 の立方根

初期区間 [ 0.00, 3.00] eps = 1.00e-03

   n           近似値           関数値
  1      0.8888888888888888      -7.2976680384087791
  2      1.4747274529236867      -4.7927316770215107
  3      2.5956210160301723       9.4873436900274264
  4      1.8509258936452322      -1.6588636283051441
  5      1.9617571044256743      -0.4501955634842156
  6      2.0030386777882758       0.0365195628933375
  7      1.9999412087904005      -0.0007054737769598
根 : 1.999941
```



## 4. 割線法（2次曲線近似）

3点  $(x_0, f(x_0))$ ,  $(x_1, f(x_1))$ ,  $(x_2, f(x_2))$  を通る2次曲線とx軸との交点  $(x_m, 0)$  を求め、根の近似値とする。

2次曲線が虚根を持つときは、軸のx座標を根の近似値とする。

$f(x_m)$  の絶対値が十分小さい値  $\text{eps}$  より小さいとき、 $x_m$  を根とする。

そうでないとき、点  $(x_1, f(x_1))$ ,  $(x_2, f(x_2))$ ,  $(x_m, f(x_m))$  を3点として同様の操作を繰り返す。

### ●プログラム (rf141.c)

$f(x) = x^3 - a$  の根を求める。

```

1  /* << rf141.c >> */
2  /* 割線法（2次曲線近似）。*/
3  #include <stdio.h>
4  #include <math.h>
5
6  int main() {
7      int n;          /* 繰り返し回数。*/
8      double a,      /* aの値。*/
9              d,      /* 2次曲線の判別式。d=q*q-4*p*r。*/
10             eps,    /* 誤差。*/
11             f0, f1, f2, /* f(x0), f(x1), f(x2)の値。*/
12             fm,     /* f(xm)の値。*/
13             p, q, r, /* 2次曲線の係数。*/
14             t0, t1,
15             x0, x1, x2, /* x0, x1, x2の値。*/
16             xm,     /* 2次曲線とx軸との交点。*/
17             z0, z1; /* 2次曲線の実根。*/
18
19     /* データ入力（例 a=8.0）。*/
20     scanf("%lf", &a);
21     printf("%lf の立方根    ", a);
22
23     /* 初期設定。*/
24     x0 = 0.0; x1 = 1.0; x2 = 3.0;
25     eps = 1.0e-10;
26     printf("初期値 x0 =%6.2lf x1 =%6.2lf x2 =%6.2lf    ", x0, x1, x2);
27     printf("eps = %6.2le ¥n¥n", eps);
28     printf("      n                近似値");
29     printf("                関数値 ¥n");
30
31     /* 計算過程と途中経過の出力。*/
32     n = 0;
33
34     /* x0, x1, x2 に対応する関数値を求める。*/
35     f0 = x0*x0*x0 - a;
36     f1 = x1*x1*x1 - a;

```

```

37 f2 = x2*x2*x2 - a;
38 while( 1 ) {
39     n++;
40
41     /* 2次曲線 : y=p*x*x+q*x+r を求める。*/
42     t0 = (f1 - f0)/(x1 - x0);
43     t1 = (f2 - f1)/(x2 - x1);
44     p = (t0 - t1)/(x0 - x2);
45     q = t0 - p*(x0 + x1);
46     r = f0 - p*x0*x0 - q*x0;
47
48     /* 2次曲線の根から近似値を求める。*/
49     /* 虚根の場合、2次曲線の軸を近似値とする。*/
50     /* 実根の場合、2根 z0, z1の内、絶対値|x2-z0|, |x2-z1|の */
51     /* 小さい方を新近似値とする。*/
52     d = q*q - 4*p*r;
53     if( d < 0 ) { /* 虚根の場合。*/
54         xm = -q*(2*p);
55     } else { /* 実根の場合。*/
56         if( q > 0 ) {
57             z0 = (-q - sqrt(d))/(2*p); z1 = (r/p)/z0;
58         } else {
59             z0 = (-q + sqrt(d))/(2*p); z1 = (r/p)/z0;
60         }
61         if( fabs(x2-z0) < fabs(x2-z1) ) {
62             xm = z0;
63         } else {
64             xm = z1;
65         }
66     }
67
68     /* xm に対応する関数値を求める。*/
69     fm = xm*xm*xm - a;
70     printf("%5d%24.16lf%24.16lf ¥n", n, xm, fm);
71
72     /* 解として出力。*/
73     if( fabs(fm) < eps ) {
74         printf("根 : %lf ¥n", xm); break;
75     }
76
77     /* 3点を更新する。*/
78     x0 = x1; f0 = f1;
79     x1 = x2; f1 = f2;
80     x2 = xm; f2 = fm;
81 }
82 }

```

実行結果

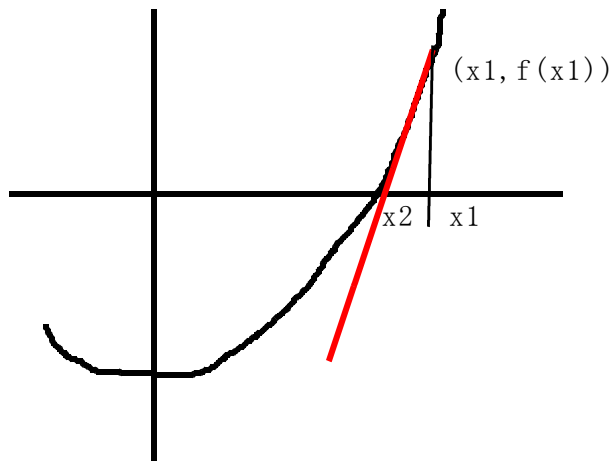
```

% cc rf141.c -lm
% a.out
8
8.000000 の立方根  初期値 x0 =  0.00 x1 =  1.00 x2 =  3.00
eps = 1.00e-10
   n                近似値                関数値
   1      1.8380874888399532      -1.7899008098476195
   2      1.9874747382256139      -0.1493638131879352
   3      1.9998333784254341      -0.0019992923229223
   4      2.0000000281638264       0.0000003379659213
   5      1.9999999999999951      -0.00000000000000586
根 : 2.000000
    
```

## 5. ニュートン法 (接線近似)

根の近似値  $x_1$  から始め、点  $(x_1, f(x_1))$  を通る関数  $f(x)$  の接線と  $x$  軸との交点  $(x_2, 0)$  を求める。  $f(x_2)$  の絶対値が十分小さい値  $\text{eps}$  より小さいとき、  $x_2$  を根とする。 そうでないとき、  $x_2$  を近似値として、同様のことを繰り返す。

関数  $f(x) = x^3 - a = 0$  の根を求める場合。



### ● プログラム (rf151.c)

$f(x) = x^3 - a$  の根を求める。

```

1  /* << rf151.c >> */
2  /* ニュートン法 (接線近似)。*/
3  #include <stdio.h>
4  #include <math.h>
5
6  int main() {
7      int n;          /* 繰り返し回数。*/
8      double a,      /* aの値。*/
9              eps,   /* 誤差。*/
10             fx,    /* f(x2)の値。*/
11             x1, x2; /* x1, x2の値。*/
12
13     /* データ入力 (例 a=8.0)。*/
14     scanf("%lf", &a);
15     printf("%lf の立方根    ", a);
16
17     /* 初期設定。*/
18     x1 = 1.0; eps = 1.0e-10;
19     printf("初期値 x1 = %6.2lf  eps = %6.21e  ¥n¥n", x1, eps);
20     printf("      n                近似値");
21     printf("                関数値 ¥n");
22

```

```

23  /* 計算過程と途中経過の出力。*/
24  n = 0;
25  while( 1 ) {
26      n++;
27
28      /* 近似値 x2 を求める。*/
29      x2 = (2*x1*x1*x1 + a)/(3*x1*x1);
30
31      /* x2 に対応する関数値を求める。*/
32      fx = x2*x2*x2 - a;
33      printf("%5d%24.16lf%24.16lf ¥n", n, x2, fx);
34
35      /* 解として出力。*/
36      if( fabs(fx) < eps ) {
37          printf("根 : %lf ¥n", x2); break;
38      }
39      x1 = x2;
40  }
41  }

```

### 実行結果

```

% cc rf151.c -lm
% a.out
8
8.000000 の立方根  初期値 x1 =  1.00  eps = 1.00e-10

      n              近似値              関数値
1      3.3333333333333335      29.0370370370370416
2      2.4622222222222221      6.9273164554183788
3      2.0813412476715789      1.0163315496105632
4      2.0031374991412871      0.0377090839858456
5      2.0000049116755041      0.0000589402507966
6      2.0000000000120624      0.0000000001447482
7      2.0000000000000000      0.0000000000000000
根 : 2.000000

```