二分法

山本昌志*

2006 年 2 月 27 日

1　本日の学習内容

本日は，非線形方程式の解を求める方法のひとつである二分法 (bisection mathod) について学習する．教科書 [1] では，バイナリサーチによる方程式の解き方と書いてある部分である．非線形方程式とは，線形 (直線) でない方程式で，

$$x^2 + x + 1 = 0 \qquad x^5 - 10x^3 + x - 3 = 0 \qquad x\sin(x) + 1 = 0 \qquad \log x + \cos(x) = 1 \tag{1}$$

等である．この中で，どれが解けるだろうか?．世の中にあるほとんどの方程式は解析解を求めることは不可能である．そこで，コンピューターを用いて，極めて精度の良い近似解を計算することになる．一般的な工学の問題では，精度の良い近似解で十分，役に立つ．

非線形方程式の解を求める方法では，ニュートン法の方が実用的であるが，時間が無いので，ここでは述べないことにする．教科書 [1] には，ニュートン法についても書かれているので，興味のある者は自分で学習せよ．また，私の 5 年生の講義でも取り上げているので，それを参考にしても良い．

2　二分法

2.1　考え方

二分法の考え方は単純であるが，非常に強力な方程式の近似解を求める方法である．考え方の基本は，閉区間 $[a, b]$ で連続な関数 $f(x)$ の値が，

$$f(a)f(b) < 0 \tag{2}$$

ならば，$f(\alpha) = 0$ となる α が区間 $[a, b]$ にある．これは，中間値の定理から保証される．こんなことを言わないまでもあたりまえである．ただ，連続な区間で適用できることを忘れてはならない．

実際の数値計算は，$f(a)f(b) < 0$ であるような 2 点 $a, b(a < b)$ から出発する．そして，区間 $[a, b]$ を 2 分する点 $c = (a + b)/2$ に対して，$f(c)$ を計算を行う．$f(c)f(a) < 0$ ならば b を c と置き換え，$f(c)f(a) > 0$ ならば a を c と置き換える．絶えず，区間 $[a, b]$ の間に解があるようにするのである．この操作を繰り返し

*独立行政法人　秋田工業高等専門学校　電気情報工学科

て，区間の幅 $|b-a|$ が与えられた値 ε よりも小さくなったならば，計算を終了する．解へ収束は収束率 1/2 の一次収束という．

後は教科書を用いて説明する．

2.2 プログラム例

教科書の二分法のプログラムをリスト 1 にしめす．前節の二分法での説明とプログラムの変数の対応は，次の通りである．

$a \rightarrow$ `left`

$b \rightarrow$ `riht`

$c \rightarrow$ `mid`

$\varepsilon \rightarrow$ `epsilon`

言うまでもないと思うが，このプログラムは，方程式

$$x^5 - 10x^4 + 25x^3 + 40x^2 + 200x - 500 = 0 \tag{3}$$

の [1, 3] にある近似解の一つを求めている．

リスト 1: 教科書の二分法のプログラム

```
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4
5  /* 解析したい関数 */
6  double func(double x)
7  {
8      return x*x*x*x*x
9          -10.0*x*x*x*x
10         +25.0*x*x*x
11         +40.0*x*x
12         +200.0*x-500.0;
13 }
14
15 /* 2分探索法（バイナリサーチ） */
16 double BinarySearch(void)
17 {
18     double left,mid,right,epsilon;
19
20     /* ↓「答に非常に近い」という範囲を定義する。
21         この値をいろいろと変えることで，答の精度を調節できる。
22         ちなみに，あまり小さくしすぎると情報落ちの関係で
23         答が求まらなくなってしまうので注意。 */
24     epsilon=0.00001;
25
26     /* 「leftとrightの間に確実に解がある」という範囲を指定する */
27     left=1.0;
28     right=3.0;
29
30     /* 範囲をひたすら絞り込む */
31     while(fabs(right-left)>epsilon &&fabs(func(left))>epsilon)
32     {
```

```
33          mid=(left+right)/2.0;
34
35          /* func(left)とfunc(mid)が同符号なら */
36          if(func(left)*func(mid)>=0.0)
37              left=mid;               /* leftの位置をmidに合わせる */
38          else
39              right=mid;              /* rightの位置をmidに合わせる */
40      }
41      return left;
42  }
43
44  int main(void)
45  {
46      double d;
47      d=BinarySearch();
48      printf("方程式の解は%lf, "
49              "そのときのfunc(x)は%lfです。\n",d,func(d));
50      return EXIT_SUCCESS;
51  }
```

3 gnuplotを使ったグラフ表示

グラフ表示を行った後，二分法により方程式の解を計算するプログラムをリスト 2 に示す．このプログラムを実行すると，図 1 のようなグラフが表示された後，近似解が，

```
answer = -1.5479375e+00
```

と表示される

次年度以降，ここで学習した C 言語の知識を他の科目で応用しようとする者は，このプログラムを理解せよ．そして，自分で書いて見よ．

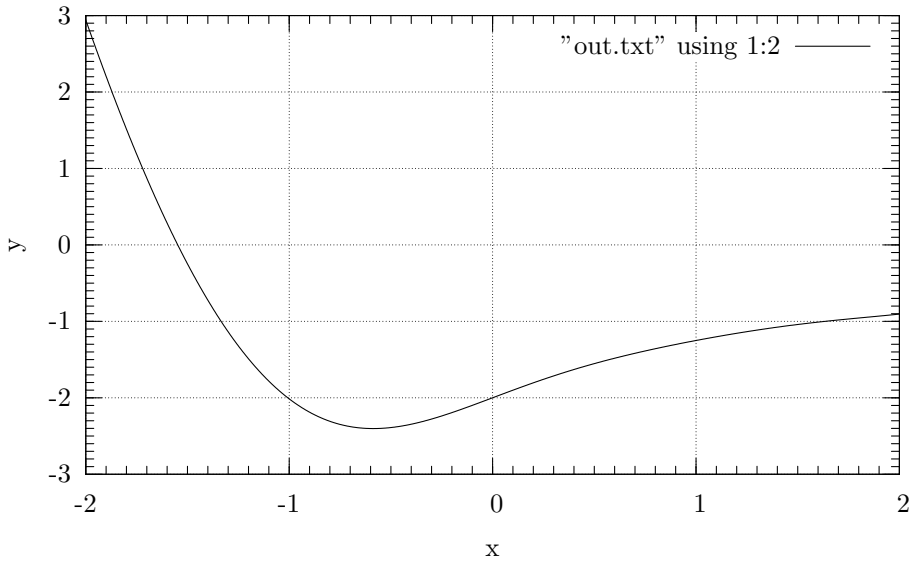

図 1: 方程式 $f(x) = 0$ を解くときの関数 $f(x)$ のグラフ表示.

リスト 2: グラフ表示付き二分法のプログラム

```c
#include <stdio.h>
#include <math.h>
#define EPSILON (1.0e-10)

double function(double x);
double bisection(double (*f)(), double left, double right);
void mk_data(char *a, double (*f)(), double x1, double x2, int n);
void mk_graph(char *f, char *xlb, double x1, double x2,
              char *ylb, double y1, double y2);

/*=============================================================*/
/*    main function                                            */
/*=============================================================*/
int main(void){

  double left, right, ans;

  left=-2.0;
  right=2.0;

  mk_data("out.txt", &function, left, right, 1000);
  mk_graph("out.txt", "x", left, right, "y", -3, 3);

  ans = bisection(&function, left, right);

  printf("answer = %12.7e\n",ans);

  return 0;
}

/*=============================================================*/
/*  to be solved equation.  f(x)=0                             */
/*=============================================================*/
double function(double x){
  double y;

  y=x*cos(x)+sin(x)+exp(-x*x)+x*x-x-3;

  return y;
}

/*=============================================================*/
/*  bisection method f(x)=0                                    */
/*=============================================================*/
double bisection(double (*f)(), double left, double right){
  double temp, mid;

  /*---- left と right が逆の場合 ----------*/

  if(left > right){
    temp = left;
    left = right;
    right = temp;
  }

  //  while(right-left > EPSILON){
  while(right-left > EPSILON){
    mid = (left+right)/2.0;

    if(f(left)*f(mid) >= 0){
```

```
61          left = mid;
62        }else{
63          right = mid;
64        }
65
66      }
67
68       return (left+right)/2.0;
69    }
70
71    /*==========================================================*/
72    /*    make a data file                                      */
73    /*==========================================================*/
74    void mk_data(char *a, double (*f)(double), double x1, double x2, int n){
75      double x, y, dx;
76      int i;
77      FILE *out;
78
79      dx = (x2-x1)/n;
80
81      out = fopen(a, "w");
82
83      for(i=0; i<=n; i++){
84        x = x1+dx*i;
85        y = f(x);
86
87        fprintf(out, "%e\t%e\n", x, y);
88      }
89
90      fclose(out);
91    }
92
93    /*==========================================================*/
94    /*    make a graph                                          */
95    /*==========================================================*/
96    void mk_graph(char *f, char *xlb, double x1, double x2,
97                  char *ylb, double y1, double y2){
98
99      FILE *gp;
100
101     gp = popen("gnuplot -persist","w");
102
103     fprintf(gp, "reset\n");
104
105     /* --------  set x grid ----------*/
106
107     fprintf(gp, "set grid\n");
108
109     /* --------  set x axis ----------*/
110
111     fprintf(gp, "set xtics 1\n");
112     fprintf(gp, "set mxtics 10\n");
113     fprintf(gp, "set xlabel \"%s\"\n", xlb);
114     fprintf(gp, "set nologscale x\n");
115     fprintf(gp, "set xrange[%e:%e]\n", x1, x2);
116
117     /* --------  set y axis ----------*/
118
119     fprintf(gp, "set ytics 1\n");
120     fprintf(gp, "set mytics 10\n");
121     fprintf(gp, "set ylabel \"%s\"\n", ylb);
122     fprintf(gp, "set nologscale y\n");
```

```
123     fprintf(gp, "set yrange[%e:%e]\n", y1, y2);
124
125           /* ------  plat graphs ---------*/
126
127     fprintf(gp, "set terminal x11\n");
128
129     fprintf(gp, "plot \"%s\" using 1:2 with line\n", f);
130
131     fprintf(gp, "set terminal epslatex\n");
132     fprintf(gp, "set output \"function.eps\"\n");
133
134     fprintf(gp, "replot\n");
135
136     pclose(gp);
137 }
```

参考文献

[1] 紀平拓男, 春日伸弥. プログラミングの宝箱　アルゴリズムとデータ構造. ソフトバンクパブリッシング(株), 2004 年.