Study on Solving Nonlinear Equations of Computers

Hu Jia
(Department of Math and Computer Science, Nanchang Normal University Nanchang Jiangxi China 330000)

Keywords: nonlinear equation, bisection method, Newton method, interpolation method, Aitken iteration method, iterative root finding

Abstract. this paper provides the process for finding roots of nonlinear equations, and gives four root finding iterative methods, which are respectively the bisection method, Newton method, interpolation method and Aitken iteration method giving their solving algorithms and using an instance of f(x)=x3-3x+1 to verify the process of the whole solution.

1. Introduction

We call the process of using the calculating devices such as calculators and electronic computers to determine the numerical solutions of mathematical problems as numerical calculation. As the rapid development of the science and technology, there are a lot of complicated numerical calculating problems to be solved not only in the field of agricultural production but also in the field of the top technology of national defense such as the design of mechanical and electrical products, the design of construction engineering projects, weather forecast, development of new sophisticated weapons, and rocket launching, etc. The occurrence of digital electronic computer greatly promotes the progress of numerical calculating method. At present, many complicated numerical calculating problems can be solved through electronic computers. As is known to us, lower-degree polynomial equations lower than quartic equations have analytical solutions, and can be solved according to the analytical solutions. As to ordinary higher-degree polynomial equations, analytical solutions can not be gained in theory. Therefore, we adopt the numerical method to obtain the approximate solution, and the iteration method is most frequently used.

2. Root Finding of Nonlinear Equations

As the basic process for finding roots of nonlinear equations, the iteration method is generally divided into two steps, in which the first step is: to analyze the approximate position of a real root according to the property of equations set and give the initial approximate value of the equation root or the interval of the equation root, and the second step is: to use the iteration method to solve the equation for the exact value of the root on the basis of the approximate value or its interval. The exact solution is the solution that satisfies the given precision. ~~Here, the precision can be taken as an error.~~

We can determine the initial approximate value of the real root of a nonlinear equation or the interval of the root through the graphic method and the step-by-step scanning method. Firstly, the graphic method is to solve the equation according to the geometrical significance of the equation, for example, the real root of the nonlinear equation of f(x) =0 is actually the abscissa of the intersection point between y=f(x) and axis x. We can directly draw the curve of f(x), and then approximately determine the abscissa of the intersection point between curve f(x) and axis x, which is the initial approximate value of the root. We can give the approximate interval of a root in the drawing as well. If it is difficult to draw curve f(x), we can change equation f(x)=0 into equation z1(x)=z2(x), draw curve z1(x) and z2(x) respectively, and then determine the approximate abscissa of the intersection point of the two curves in the drawing or determine the interval of the abscissa of the intersection point.

Secondly, the step-by-step scanning method is that the equation has real roots among interval [a,b]. We select h as the step size, start from the leftmost endpoint a, and solve equations xk=a+k*h, k=0, 1, 2, 3…; their corresponding function value is f(xk). If the function value of two adjacent

point is an opposite sign, f(xk)f(xk+1)<0, which indicates that there must be a real root in interval [xk,xk+1]; if the function value of a point is 0, xk is a real root of the equation. We should carefully select step size h by using step-by-step scanning method. If step size h is too large, it may cause the loss of the real root; if step size h is too small, it increases the calculation complexity. As to nonlinear equations, we recommend to carry out theoretical analysis at first, determine the number of real roots and approximate interval range approximately according to the drawing, and carry out step-by-step scanning with respect to required real roots to determine their intervals.

After the approximate solution or interval of nonlinear equation roots are gained, the iteration method can be used to correct the roots or their range until the precision that we require is satisfied. In an engineering project, we generally find roots under certain constraint conditions. Under such constraint conditions, only one real root within a range needs to be obtained in general. In the following, we will introduce a few methods for finding a real root of a nonlinear equation.

3. Several Methods of Iteration Method

3.1 Bisection Method

The first method is bisection method. If function f(x) is nonlinear, there is a nonlinear equation of f(x)=0, f(x) is continuous within its interval [a,b], and it has a real root, which is f(a)f(b)<0. To obtain a real root that satisfies the precision of ε. The idea of bisection method is to bisect the interval each time so as to narrow the interval range of the root until the given precision is satisfied. Take (a+b)/2 as the interval midpoint x, and obtain f(x) of the midpoint. F(x) can be divided into two conditions. First, if f(x)=0, x is the root of the equation to be solved, the value is (a+b)/2, and the iteration is terminated. Second, if f(x) < 0 or f(x)>0, f(a)f(b)<0, take the point which has opposite sign with x from a,b to form a new interval with x. Because the values of two endpoints of the new interval are opposite signs, the equation must have a real solution within the new interval, in this way, the interval range is narrowed down successfully. The iteration is carried out continuously on the basis of such idea until the interval length is less than ε, and at this time, the iteration loop is finished. Therefore, we can take the midpoint of the interval range as our final result. Its algorithm is as follows:

(1) Take the midpoint of the interval x=a+b/2.
(2) If f(x)=0, x is a real root, and the process is terminated.
(3) If f(a)f(x)<0, make b=x; if f(x)f(b)<0, make a=x;
(4) If |a-b|<ε, the process is terminated, and the final result is a+b/2; otherwise, repeat the execution of the process from step (1). The functional function is as follows:

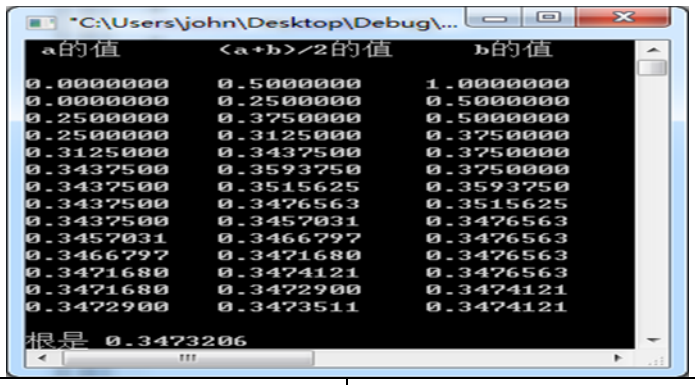
```
/*=================================================================
========
Input parameters: a, b is respectively the left endpoint and right endpoint of the interval.
                  F() is a function, and the function is f(x)=x3-3x+1.
                  eps is precision.
                  max is the maximum numbers of iterations.
Return value: 0 refers to failure of iteration, and 1 refers to success of iteration.

=================================================================
===*/

double disroot(double a,double b,double (*f)(),double eps)
{
double x,y_mid;
double y_a=(*f)(a);
double y_b=(*f)(b);
while(b-a>eps)
{
```

```
  x=(a+b)/2;
  printf("%1.7f      %1.7f        %1.7f\n",a,x,b);
  y_mid=(*f)(x);
  if(y_mid==0)return (x);
  if(y_a*y_mid<0.0)
  {b=x;y_b=y_mid;}
  else
  {a=x;y_a=y_mid;}
  }
  return((a+b)/2);
}
double f(double x)
{
double z;
z=x*x*x-3*x+1;
return z;
  }
```

The running result is as follows:



a 的值	Value of a
<a+b>/2 的值	Value of <a+b>/2
b 的值	Value of b
根是	The root is

3.2 Newton Method

The second method of solution is Newton method. At present, use the Newton method to find a real root that satisfies the precision of ε. If the first derivative and second derivative of f(x) exists within interval range [a,b], and the signs of the first derivative and second derivative of f(x) remain the same within the interval [a,b] and in the meanwhile satisfy f(a)f(b)<0,f(x0) and f''(x0)>0,x0∈[a,b], use the Newton method to obtain a real solution that meets the precision requirement. The nonlinear equation f(x)=0 is known, we can obtain its inverse function f-1(y). If x0 is the initial approximate value of the nonlinear equation solution, y0=f(x0),x0=f—1(y0), and obtain (f-1)(y0)=1/(f'(x0)) according to the knowledge of inverse function derivative. Given that the root of the nonlinear equation f(x)=0 is r, then r=f-1(0); now, carry out f-1(y) for Taylor expansion at y=y0, and obtain f-1(y)=f-1(y0)+(f-1)'(y0)(y-y0)+(f-1)''(y0)(y-y0)2+…; then approximately get the series of the first two items. It is f-1(y)=f-1(y0)+(f-1)'(y0)(y-y0), substitute y=0 into the equation, and obtain the result of f-1(y)=f-1(y0)+(f-1)'(y0)(0-y0), namely r=x0+(0-y0)/f'(x0)= x0-f(x0)/f'(x0); in this way, we obtain an approximate expression of the nonlinear equation solution. Because only the series of the first two items are taken, it is not an accurate solution, which does not

necessarily satisfy our precision. Therefore, we use the following sequences to solve the equation xk+1 =xk-f(xk)/f'(xk),k=0,1,2… for iteration until it satisfies | xk+1- xk |<ε and | f(xk+1)|<ε.

The algorithm is as follows:

(1) Select x_0 that satisfies $f(x_0)f''(x_0)>0$ within the interval range [a,b].

(2) Calculate the value of x, $x=x_0-f(x_0)/f'(x_0)$.

(3) If $|x-x_0|<\varepsilon$ and $|f(x)|<\varepsilon$, the process is terminated, and x is the final result. Otherwise, make $x_0=x$, repeat the execution of the process from step (2).

The functional function is as follows:

```
/*=================================================================
========
    Input parameters: x0 refers to the indicator of iteration initial value and final value.
                      f() is a function, and the function is f(x)=x3-3x+1.
                      fd() is the derivative of f().
                      eps is the precision.
                      max is the maximum numbers of iterations.
    Return value: 0 refers to failure of iteration, and 1 refers to success of iteration.
=================================================================
===*/
    int newdonroot(double *x0,double (*f)(),double (*fd)(),double eps,double max)
      { double x,dis;
      double y0;
      double yd0;
      int num=0;
      do
    { y0=f(*x0);yd0=fd(*x0);
      x=*x0-y0/yd0;
      num++;
      dis=fabs(x-*x0);
      if(dis<fabs(y0))
      dis=fabs(y0);
      printf("%1.7f          %1.7f       %1.7f\n",*x0,x,dis);
      *x0=x;}while(dis>eps&&num<max);
    if(num==max)
    return(0);
    else
    return(1);
    }
```

The running result is as shown in the screenshot:

计算机非线性方程的求解	Solution to nonlinear equations of computers

3.3 Interpolation Method

The third method is interpolation method. It is a transformation of Newton method. As to Newton method, it needs less information and does not need the derivative of the function, and it is more frequently applied in the field of engineering. Given the nonlinear equation f(x)=0,f(x)

satisfies the first derivative and second derivative within the interval range [a,b], and the signs of the first derivative and second derivative within the interval range [a,b] remain the same, f(a)f(b)<0, use interpolation method to obtain a real solution that satisfies the precision requirement. During the interpolation method, substitute the derivative in the equation with difference quotients, namely f(x)-f(a)/x-a or f(x)-f(b)/x-b, and in this time we can obtain the iteration equation xk+1 =xk-(x-a/f (xk)-f(a))×f(xk) or xk+1 =xk-(x-b/f (xk)-f(b))×f(xk). These two equations are difference iteration equations. The following conclusions can be obtained according to the geometrical significance of these two equations: 1. when the signs of the first derivative and second derivative of f(x) are the same, get the initial value of x0=a, the iteration equation uses xk+1 =xk-(x-b/f(xk)-f(b))×f(xk); 2. when the signs of the first derivative and second derivative of f(x) are not the same, get the initial value of x0=b, the iteration equation uses xk+1 =xk-(x-a/f (xk)-f(a))×f(xk).

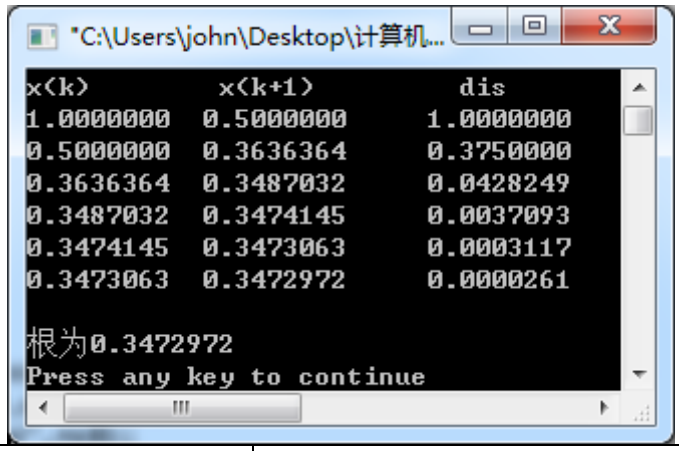
Its algorithm is as follows:

(1) Decide the signs of f'(x) and f"(x) within the interval range [a,b]. If the signs o f f'(x) and f"(x) are the same, execute step (2); if the signs o f f'(x) and f"(x) are not the same, execute step (5).
(2) Make the initial value x_0=a.
(3) Solve the equation x =x_0-(x_0-b/f(x_0)-f(b))×f(x_0).
(4) If |x-x_0|<ε and |f(x)|<ε, the process is terminated, and x is the final result; otherwise, make x_0=x, repeat the execution of the process from step (2).
(5) Make the initial value x_0=b.
(6) Solve the equation x =x_0-(x_0-a/f(x_0)-f(a))×f(x_0).
(7) If |x-x0|<ε and |f(x)|<ε, the process is terminated, and x is the final result; otherwise, make x_0=x, repeat the execution of the process from step (6).

The functional functions are as follows:

```
/*=================================================================
========
    Input parameters: x0 refers to the indicator of iteration initial value and final value.
                      a, b are the interval endpoints that are not selected as the initial values.
                      f() is a function, and the function is f(x)=x3-3x+1.
                      eps is the precision.
                      max is the maximum numbers of iterations.
    Return value: 0 refers to failure of iteration, and 1 refers to success of iteration.
=================================================================
===*/
int interroot(double *x0,double ab,double (*f)(), double eps, double max)
{
double x,dis;
double y0,y_d,y_ab=f(ab);
int num=0;
do
{y0=f(*x0);
  y_d=(*x0-ab)/(y0-y_ab);
  x=*x0-y_d*y0;
  num++;
  dis=fabs(x-*x0);
  if(dis<fabs(y0))
  dis=fabs(y0);
  printf("%1.7f    %1.7f         %1.7f\n",*x0,x,dis);
  *x0=x;
}
while(dis>eps&&num<max);
if(num==max)
return (0);
```

```
else
return (1);}
```



计算机	Computer
根为	The root is

3.4 Aitken Iteration Method

The Aitken iteration method is an improvement of a simple iteration method, because the simple iteration has its limitation. First, whether the iteration sequence obtains the real root of the equation at first; second, the convergence speed is not necessarily objective, and this is the reason why the simple iteration method can not be applicable to all the nonlinear equations. In the Aitken iteration method, given that the real root of f(x)=0 is α, in the simple iteration equation xk+1=φ(xk),k=0,1,2…, the ratio of the iteration error of No. k+1 and the iteration error of No. k is xk+1-α/xk-α=q=φ'(ξ), in which ξ is a point between α and xk. The equation α=(q/(1-q))$\times$(xk+1-xk)+ xk+1 can be solved, but q=φ'(ξ) can not be determined. We get its approximate form to carry on, and the iteration equations are as follows:

x(1)k+1=φ(xk)

q=φ'(xk)

xk+1= x(1)k+(q/(1-q))$\times$ (x(1)k+1-xk)

The above iterations are not very convenient in actual applications. Not only φ(x) must be calculated, but also its derivative should be calculated. In order to make it convenient for us to substitute the derivative with the form of difference quotient, use the simple iteration method to obtain x(1)k+1 at first. The difference quotient of x(1)k+1 and the real root is x(1)k+1-α/xk-α=q1=φ'(ξ_1), and use x(1)k+1 to obtain x(2)k+1. The iteration equation is x(2)k+1=φ(x(1)k+1), and the difference quotient of it and the real root is x(2)k+1-α/x(1)k+1-α=q2=φ'(ξ_2). Given that the difference quotients of these two times are the same, namely q1=q2, thus we obtain:

x(1)k+1-α/xk-α=x(2)k+1-α/x(1)k+1-α, from which α is obtained, and at last we obtain the new Aitken iteration format as follows:

x(1)k+1=φ(xk)

x(2)k+1=φ(x(1)k+1)

xk+1= x(2)k+1- （x(2)k+1- x(1)k+1）2/ (x(2)k+1- 2x(1)k+1+ xk)

Such format is applicable to all the equations, and has no convergence problems, which greatly improves the convergence speed.

Its algorithm is as follows:

(1) Determine the initial iteration value as x_0.

(2) Given $x^{(1)}=\varphi(x_0)$, $x^{(2)}=\varphi(x^{(1)})$.

(3) Solve the equation $x= x^{(2)} - (x^{(2)} - x^{(1)})^2 / (x^{(2)} - 2x^{(1)} + x_0)$.

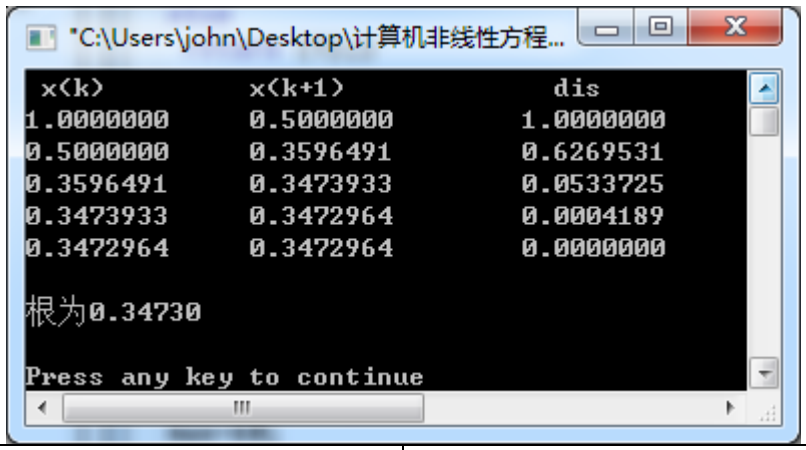
(4) If $| x^{(2)} - x^{(1)} |<\varepsilon$, the process is terminated, and x is the final result; otherwise, make $x_0=x$, repeat the execution of the process from step (2).

The functional functions are as follows:

```
/*==========================================================================
========
Input parameters: x0 is the indicator of the iteration initial value and final value.
                  f() is an iterative function, and the iterative function is f(x)=x^3-3x+1.
                  eps is the precision.
                  max is the maximum numbers of iterations.
Return value: 0 refers to failure of iteration, and 1 refers to success of iteration.
==========================================================================
===*/

int aitkenroot(double *x0,double (*f)(),double eps,int max)
{
  double x,x1,x2,dis;
  int num=0;
  do
  {
  x1=f(*x0);
  x2=f(x1);
  dis=fabs(x2-x1);
  x=x2-dis*dis/(x2-2*x1+*x0);
  num++;
  printf("%1.7f        %1.7f             %1.7f\n",*x0,x,dis);
  *x0=x;
    }while(dis>eps&&num<max);

if(num==max)return(0);
else
  return (1);}
```



计算机非线性方程	Nonlinear equations of computers
根为	The root is

4. Summary

Iteration method is a successive approximation, which uses some fixed equation- so called iteration equation to repeatedly verify the approximate value of the root so as to achieve its precision step by step until the result that satisfies the precision requirement is obtained.

References

[1] *Calculation Method (version two)* by Yi Dayi, Shen Yunbao, Li Youfa Zhejiang University Press in 2002
[2] *Modern Numerical Calculation* by Teaching and Research Section for Computational Mathematics of Tongji University Posts and Telecom Press in 2009
[3] *MATLAB Practical Numerical Analysis* by Zhang Defeng Tsinghua University Press in 2012
[4] *Numerical Analysis (version four)* by Li Qingyang, Wang Nengchao, Yi Dayi Huazhong University of Science and Technology Press in 2015