Peter the Great St.Petersburg

Polytechnic University

Laboratory work №2

«Root finding for nonlinear equations »

Student: Vedmankin D.V.

3332302/90001

Check: Kurtc V.V.

Saint Petersburg

2021

1. Problem statement

Let f is a real-valued function of one variable.

We use numerical approach ()

**Find x\***: f(x\*) = 0

Segment with only one root, which I have chosen, is [-3; 0].

1. Methods

Stages:

We need to identify a segment with all roots, then choose the one with single root.

Then we improve the precision of the result. ,

i is the number of iterations.

Methods:

1**) The bisection method**.

Algorithm :

while |a-b| > Ɛ

c =

if f(a)\*f(c) < 0

b = c;

else

if f(c)\*f(b) < 0

a = c ;

else break

end

end

Applicability condition:

2) **Chord method:**

Chose the initial value:

Chose fixed point:

Then the sequence

**Convergence**

If assume that

1. are functions of fixed sign in [a, b],

sequence converges and the following inequality holds :

where on

**For example:**

The bisection method

*a=9, b=10*

Convergence condition

*f(a)f(b)<0*

*c=*

*f(*

*next*

*f*

Chords method

Choose the range

*a=9, b=10*

are functions of fixed sign in [a, b],

Chose the initial value:

Chose fixed point:

Convergence condition

*f(a)f(b)<0*

Fixed point

Initial point

Hence Chords method converging

Pick

Then

1. Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | x\* | i | Ɛ |
| The bisection method | 9.001 | 1 |  |
| 9.0016 | 11 |  |
| 9.001599 | 21 |  |
| The chords method | 9.0000498 | 1 |  |
| 9.00004975 | 2 |  |
| 9.00004975 | 2 |  |

Figure 1. Result of chord method

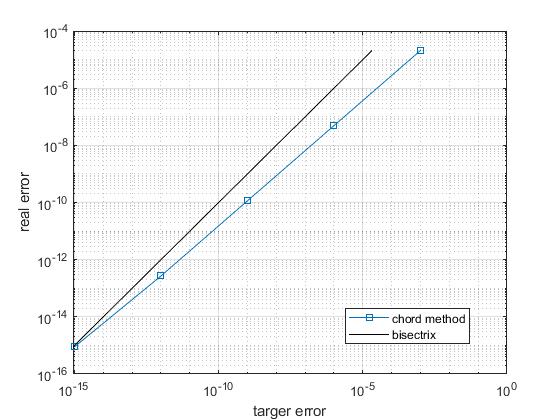
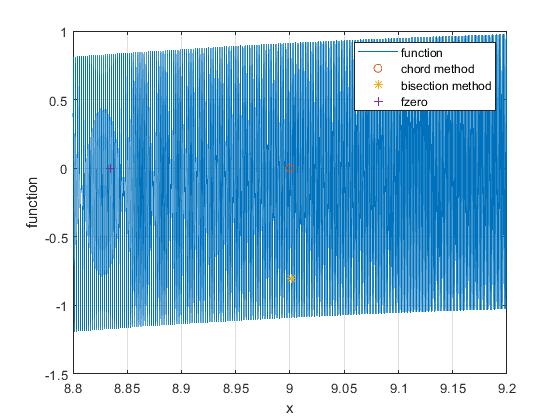


Figure 2. Result of comparison



1. Discussion:

During the experiment we have found the root of equation with two different methods. Both of them can be used successfully, but in this specific experiment the chord method speed of convergence is 10 times higher than the bisection method’s one.

1. Conclusion

The bisection method is less effective than the chord method. With decreasing accuracy Ɛ, the number of iterations of the method increases.

1. Appendix

6.1 Texts of programs

6.1.1 The bisection method

clear;figure('Name','метод половинного деления');

a=9; b=a; eps=10.^-3; %ITER= [];epsY1=[];

f=@(x) sin(cosh(x)+1)-sinh(cos(x)+1);

% syms t; diff(f,t)

%=cosh(cos(x)+1)\*sin(x)+cos(cosh(x)+1)\*sinh(x)

f1=@(x)cosh(cos(x)+1)\*sin(x)+cos(cosh(x)+1)\*sinh(x);%first derivative

f2=@(x)cosh(cos(x)+1)\*cos(x)-sin(cosh(x)+1)...

\*sinh(x)^2-sinh(cos(x)+1)\*sin(x)^2+cos(cosh(x)...%second derivative

+ 1)\*cosh(x);

while sign(f(a).\*f(b)) > 0 || b==10

%|| sign(f1(a).\*f1(b)) > 0 || sign(f2(a).\*f2(b)) > 0

b=b+0.001;

end %applicability #1

disp("b = ");disp(b);

m=1;ITER(1)=1;EPS(1)=abs(a-b);

c=(a+b)/2;

while abs(a-b)>=2\*eps

c=(a+b)/2;

if (f(a)\*f(c)) < 0

b=c;

else

a=c;

end

%plot(m,log10(abs(a-b)/2),'o');

m=m+1;

ITER(m)=m;%#ok<\*SAGROW>

EPS(m)=abs(a-b);

end % bisection method

disp("c = ");disp(c);

disp("fzero ");fzero(f,c)

semilogy(ITER,EPS,'o-');

ylabel('точность вычислений');

xlabel('количество итерации');

6.1.2 The chord method

clear;

a=9; b=a; eps=10^(-5);%iterX1= [];epsY1=[];

f=@(x) sin(cosh(x)+1)-sinh(cos(x)+1);

syms t; diff(f,t)

%=cosh(cos(x)+1)\*sin(x)+cos(cosh(x)+1)\*sinh(x)

f1=@(x)cosh(cos(x)+1)\*sin(x)+cos(cosh(x)+1)\*sinh(x);

f2=@(x)cosh(cos(x)+1)\*cos(x)-sin(cosh(x)+1)\*sinh(x)^...

2-sinh(cos(x)+1)\*sin(x)^2+cos(cosh(x)+ 1)\*cosh(x);

while sign(f(a).\*f(b)) > 0 || b==10 %sign(f1(a).\*f1(b)) > 0 || sign(f2(a).\*f2(b)) > 0

b=b+0.00001;

end %applicability

if f(a)\*f2(a)>0 % fixed point/ initial value

fixp=a; inVal=b;

else

fixp=b; inVal=a;

end

fixf=f(fixp);

disp("b = ");

disp(b);

m=1; hold on;

while 1

x1=inVal-( f(inVal)\*((inVal-fixp)/(f(inVal)-fixf))) ;

ITER1(m)=m;%#ok<\*SAGROW>

EPS1(m)=abs(x1-inVal);

%plot(m,log10(abs(x1-inVal)),'o');

m=m+1;

if abs(x1-inVal)<=eps

break;

end

inVal=x1;

end% chord method

disp("x = ");disp(x1);

disp("fzero ");fzero(f,a)

figure('Name','chord method');

semilogy(ITER1,EPS1,'s-');

xlabel('iterations');

ylabel('epsilon');

hold on;LAB2\_1;

legend('chord method','bisection method')