数学实验作业第十三周

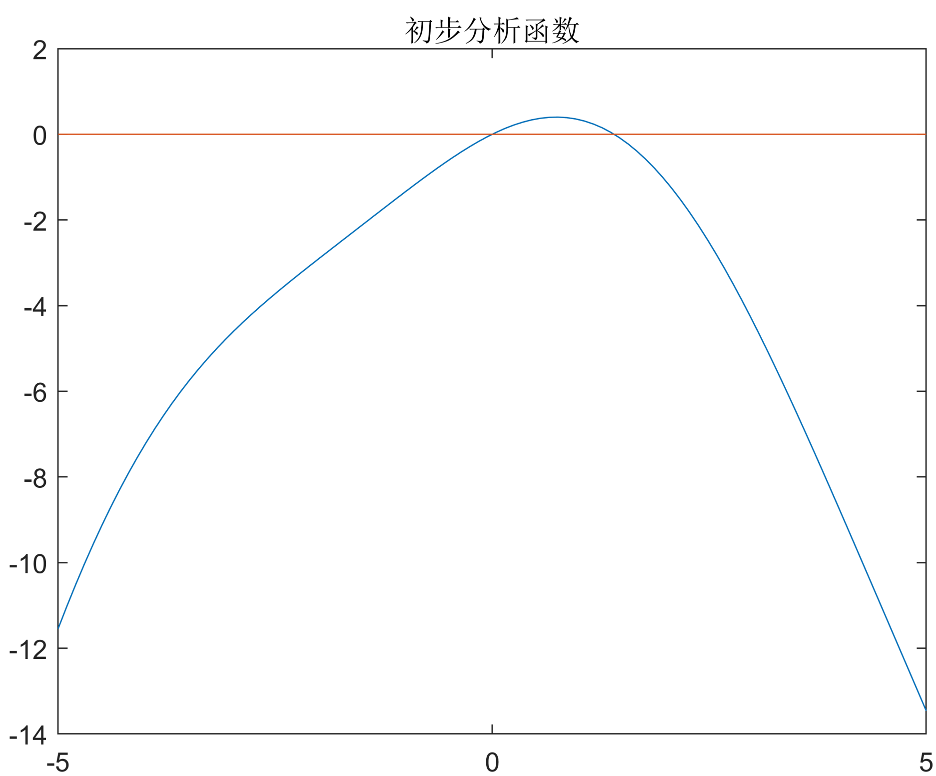
学号：202023092020

姓名：冯健齐

目录

fzero程序......1
分析收敛域......2
fsolve程序......3
分析收敛域......3
迭代法求解......4
牛顿法......5

```
xx=linspace(-5,5,100);
y=sin(xx)-(xx.^2)/2;
plot(xx,y,[-5,5],[0,0]);
title('初步分析函数');
```



有图以及函数性质可知，有两个根，一个在0附近，一个在[1,2]上

fzero程序

```
fun=@(x) sin(x)-(x^2)/2;%输入方程
options=optimset('TolX',1e-10,'Display','off');
for x0=[0 1 2]
    x0
    [x,fv,ef,out]=fzero(fun,x0,options);
    vpa(x,10)
    times=out.iterations
end
```

```
x0 = 0
ans = 0.0
times = 0
x0 = 1
ans = 1.404414824
times = 6
x0 = 2
ans = 1.404414824
times = 5
```

```
%intervaliterations求包含根的区间所采取的迭代次数
%iterations 求零点迭代次数
%funcCount 函数计算次数
%algorithm 'bisection, interpolation'
%message 退出消息
```

可见，根约为0与1.404414824

分析收敛域

```
for x0=[0 1 2 5 10 14 15]
    x0
    [x,fv,ef,out]=fzero(fun,x0);
    times=out.iterations
end
```

```
x0 = 0
times = 0
x0 = 1
times = 6
x0 = 2
times = 5
x0 = 5
times = 13
x0 = 10
times = 9
x0 = 14
times = 6
x0 = 15
正在退出 fzero: 将终止搜索包含符号变化的区间
 因为在搜索期间遇到 NaN 或 Inf 函数值。
(-1.60894e+154 处的函数值为 -Inf。)
请检查函数或使用其他起始值重试。
times = 0
```

```
%由数据可以初步分析看出，0的收敛域(-oo,a)中a在[0,1]之间
%第二个根的收敛域(a,b)中b在[14,15]之间
a=0.7;
```

```
ten=1*10^-5;
x1=0;
while(x1<1)
    a=a+ten;
    x1=fzero(fun,a,options);
end
a
```

```
a = 0.7372
```

```
b=14.7;
x2=1.4;
while(x2<2)
    b=b+ten;
    x2=fzero(fun,b,options);
end
b
```

```
b = 14.7984
```

可见收敛域为（-oo，0.7372）与（0.7372，14.7984）

fsolve程序

```
clear;clc;
fun=@(x) sin(x)-(x^2)/2;%输入方程
options=optimset('tolx',1e-10,'Display','off');
for x0=[0 1 2]
    x0
    [x,fv,ef,out]=fsolve(fun,x0,options);
    x=vpa(x,10)
    times=out.iterations
end
```

```
x0 = 0
x = 0.0
times = 0
x0 = 1
x = 1.404414824
times = 6
x0 = 2
x = 1.404414841
times = 4
```

可见，根约为0与1.404414824

分析收敛域

```
options=optimset('tolx',1e-10,'Display','off');
for x0=[0 1 2 5 10 14 15 50 100 1000 10^5 10^10]
    x0
    [x,fv,ef,out]=fsolve(fun,x0,options);
    x=vpa(x,10)
    times=out.iterations
end
```

```
x0 = 0
x = 0.0
times = 0
x0 = 1
x = 1.404414824
times = 6
x0 = 2
x = 1.404414841
times = 4
x0 = 5
x = 1.404414931
times = 6
x0 = 10
x = 1.404414931
times = 8
x0 = 14
x = 1.404415056
times = 8
x0 = 15
x = 1.404414978
times = 9
x0 = 50
x = 1.404415088
times = 11
x0 = 100
x = 1.404414975
times = 13
x0 = 1000
x = 1.404414975
times = 19
x0 = 100000
x = 1.404414934
times = 31
x0 = 1.0000e+10
x = 243.4432479
times = 49
```

```
%由数据可以初步分析看出，0的收敛域(-oo,a)中a在[0,1]之间
%第二个根的收敛域为(a,+oo)
a=0.7;
ten=1*10^-5;
x1=0;
while(x1<1)
    a=a+ten;
    x1=fsolve(fun,a,options);
end
a
```

```
a = 0.7391
```

由分析可见，收敛域为（-oo,0.7391)与(0.7391,+oo)

迭代法求解

```
clear;clc;
wucha=1e-10;
i=1;
cha=1;
x(1)=1;
while(abs(cha)>wucha)
    x(i+1)=(2*sin(x(i)))^0.5;
    cha=x(i+1)-x(i);
    i=i+1;
end
x=vpa(x',10);
disp(x);
```

$$\begin{pmatrix} 1.0 \\ 1.297282533 \\ 1.387679869 \\ 1.402341597 \\ 1.404168809 \\ 1.404385791 \\ 1.4044114 \\ 1.40441442 \\ 1.404414776 \\ 1.404414818 \\ 1.404414823 \\ 1.404414824 \\ 1.404414824 \end{pmatrix}$$

```
i
```

```
i = 13
```

0的解显而易见，关于另一个解，可见迭代法求出的解与上面一致

牛顿法

```
clear;clc;
f=@(x) sin(x)-(x^2)/2;%输入方程
df=@(x) cos(x)-x;
[x k xv]=new(f,df,1,100,1e-10);
x=vpa(x,10)
```

x = 1.404414824

```
k
```

```
k = 6
```

```
xv
```

```
xv = 1×7
    1.0000    1.7428    1.4641    1.4070    1.4044    1.4044    1.4044
```

可见牛顿法求解也一致，而且相对于迭代法，迭代次数较少

```
function [r, t, xv] = newton(f, df, x0, n, tol)
x(1) = x0;
x(2) = x(1) - f(x(1)) / df(x(1));
k=2;
while abs(x(k) - x(k-1)) > tol*abs(x(k))
    x(k+1) = x(k) - f(x(k)) / df(x(k));
    k = k + 1;
    if(k > n)
        error('Error');
    end
end
r = x(k); % root
if nargout > 1
    t = k - 1;
end
if nargout == 3
    xv = x;
end
end
```