第二章 非线性方程求根

2.2 利用阻尼牛顿法求方程的根

解题思路

本题中设置 $\lambda_0 = 1, epsilon = 0.000001$

在`newton.m`文件中实现了阻尼/无阻尼牛顿算法，主要算法代码参照书上伪代码实现如下：

```
%参数
%    f:要求解的函数
%    f1:f的一阶导函数
%    x:初始值x0
%    no_lambda:取0时使用阻尼牛顿算法，取1时使用牛顿算法
function [ x ] = newton(f,f1,x,no_lambda)
delta = 0.000001; %epsilon
xprev = 0;
i = 0;
%主迭代过程
while abs(f(x)) > delta || abs(x-xprev) > delta %约束条件
    xprev = x;
    s = f(x)./f1(x);
    x = x - s;
    i = i + 1;
    if(no_lambda == 0)
        lambda = 1;
        while abs(f(x)) >= abs(f(xprev))
            x = xprev - lambda*s;
            lambda = lambda/2;
        end
        fprintf('i=%d:lambda=%f\t',i,lambda)
    else
        fprintf('i=%d,x=%f\t',i,x)
    end
end
fprintf('\nfinished with i = %d\n',i)
end
```

在`f1.m`,`f1_diff.m`,`f2.m`,`f2_diff.m`四个文件中分别存放了（1）中函数，（1）中函数的一阶导，（2）中函数，（2）中函数的一阶导。`main.m`是控制计算流程的Main进程，`main.m`如下：

```
fprintf('*************阻尼牛顿法计算f1零点********\n');
```

```
x1=newton(@f1,@f1_diff,0.6,0);
x1_t=fzero('f1',0.6);
fprintf('x1=%f,x1_t=%f \n',x1,x1_t)

fprintf('*************阻尼牛顿法计算f2零点********\n');
x2=newton(@f2,@f2_diff,1.35,0);
x2_t=fzero('f2',1.35);
fprintf('x2=%f,x2_t=%f \n',x2,x2_t)

fprintf('*************牛顿法计算f1零点********\n');
x1=newton(@f1,@f1_diff,0.6,1);
x1_t=fzero('f1',0.6);
fprintf('x1=%f,x1_t=%f \n',x1,x1_t)
fprintf('*************牛顿法计算f2零点********\n');
x2=newton(@f2,@f2_diff,1.35,1);
x2_t=fzero('f2',1.35);
fprintf('x2=%f,x2_t=%f \n',x2,x2_t)
```

实验结果

```
*************阻尼牛顿法计算f1零点********
i=1:lambda=0.015625,x=1.140625  i=2:lambda=1.000000,x=1.366814
i=3:lambda=1.000000,x=1.326280  i=4:lambda=1.000000,x=1.324720
i=5:lambda=1.000000,x=1.324718  i=6:lambda=1.000000,x=1.324718
finished with i = 6
x1=1.324718,x1_t=1.324718
*************阻尼牛顿法计算f2零点********
i=1:lambda=0.062500,x=2.496959  i=2:lambda=1.000000,x=2.271976
i=3:lambda=1.000000,x=2.236902  i=4:lambda=1.000000,x=2.236068
i=5:lambda=1.000000,x=2.236068
finished with i = 5
x2=2.236068,x2_t=2.236068
*************牛顿法计算f1零点********
i=1,x=17.900000 i=2,x=11.946802 i=3,x=7.985520  i=4,x=5.356909  i=5,x=3.624996
i=6,x=2.505589  i=7,x=1.820129  i=8,x=1.461044  i=9,x=1.339323
i=10,x=1.324913 i=11,x=1.324718 i=12,x=1.324718
finished with i = 12
x1=1.324718,x1_t=1.324718
*************牛顿法计算f2零点********
i=1,x=10.525668 i=2,x=7.124287  i=3,x=4.910781  i=4,x=3.516911  i=5,x=2.709743
i=6,x=2.336940  i=7,x=2.242244  i=8,x=2.236093  i=9,x=2.236068
i=10,x=2.236068
finished with i = 10
x2=2.236068,x2_t=2.236068
```

可以看出我实现的算法结果和`fzero`函数结果一致，且阻尼牛顿算法迭代步骤比牛顿算法更少，收敛更快。

实验结论

阻尼因子λ在第一步迭代时对收敛速度的增益较大，阻尼牛顿算法较牛顿算法而言收敛速度有很大增加，能够在第一步就向接近正确解的方向收敛。

2.3 fzerotox求解第一类零阶贝塞尔曲线零点

实现思路

直接按照教材2.6实现`fzerotx.m`函数，然后在入口文件`main.m`中按$[i,i+3.14],i=0,1*3.14,2*3.14...9*3.14$的区间调用该函数求出前十个零点，然后绘图即可。

`main.m`实现如下：

```
zero_point = [];
for i=0:3.14:3.14*9
   zero_point = [zero_point,fzerotx(@(x)besselj(0,x),[i,i+3.14])];
end
zero_point
x = 1:0.001:35;
plot(x,besselj(0,x),zero_point,zeros(1,10),'*r')
grid on
```

实验结果

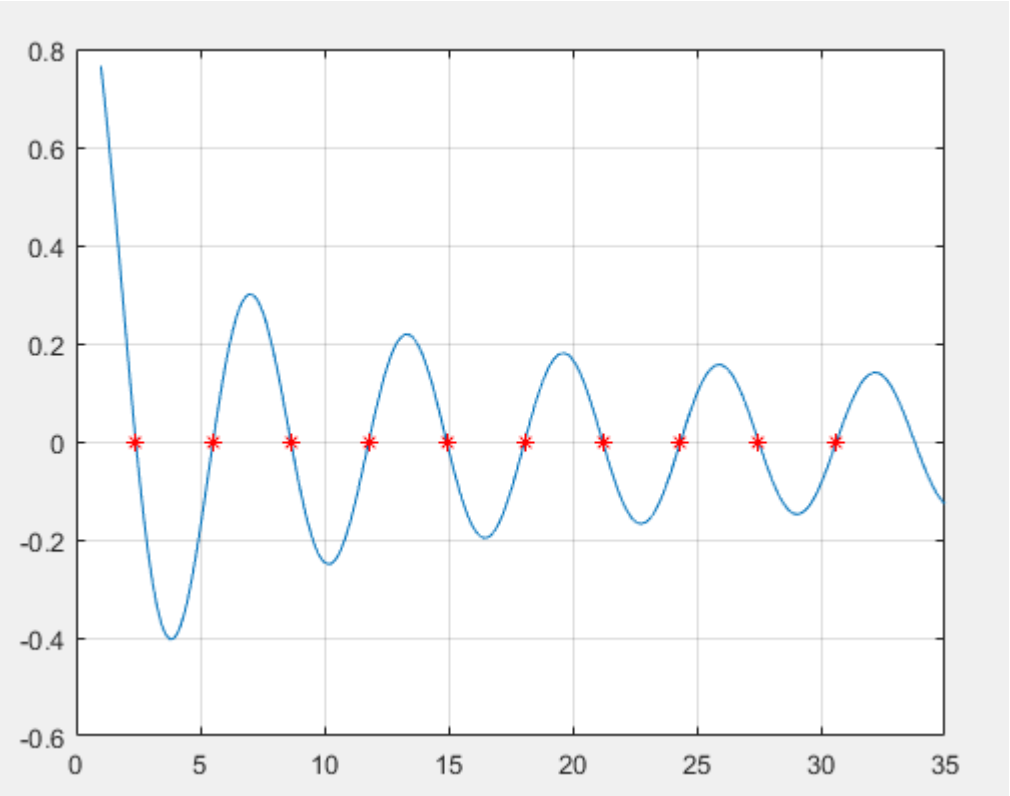
零点值：

```
>> main

zero_point =

    2.4048    5.5201    8.6537   11.7915   14.9309   18.0711   21.2116   24.3525   27.4935   30.6346
```

函数图像：



实验结论

fzerotx算法可以求出小区间，从而得到零点值。最终结果和实际函数图像契合。

在本题中，我学会了matlab中的for循环语句和在图像中画点，对fzerotx算法有了更深入的了解。