# 非线性方程求根

2020 年 6 月 17 日

## 0.1 (1)

阻尼牛顿的实现见 dampedNewton.m，既可以选择阻尼也可以选择无阻尼，默认有阻尼，设定的阻尼因子一开始都是 1(需要在 0 到 1 之间)，判定准则是：f(x) 的值不能比 0 大过设定的误差限 (1E-4)，或者两次得到的 x 的差不超过误差限，也设定了最大迭代次数 600，超过迭代次数，则停止并报错。

[11]:
```
syms x
f1=x^3-x-1;
df1=diff(f1,x);
df1_=inline(df1);
f1_=inline(f1);
f2=-x^3+5*x;
f2_=inline(f2);
df2=diff(f2,x);
df2_=inline(df2);
```

### 0.1.1 第 1 个函数

[7]:
```
dampedNewton(f1_,df1_,0.6)
fprintf(" 准确解:%d",fzero(f1_,0.6))
```

```
x=1.140625e+00,lambda:1.562500e-02
x=1.366814e+00,lambda:1
x=1.326280e+00,lambda:1
x=1.324720e+00,lambda:1


ans =
```

```
    1.3247
```

```
准确解:1.324718e+00
```

得到的近似解是 1.324720，fzero 的结果为 1.3247，前 5 位有效数字是一样的

再来考虑不带阻尼的，看看迭代步数有没有区别

[8]:
```
[res,count]=dampedNewton(f1_,df1_,0.6,false)
```

```
res =

    1.3247


count =

    11
```

同样得到 5 位有效数字。但迭代步数更多，前者是 4 次，后者是 11 次，为什么出现这个差异？因为阻尼法保证了 abs(f(x)) 是单减的，这是牛顿法无法保证的，因此它能加速判停条件 abs(f(x))<ep。

### 0.1.2 第 2 个函数

[18]:
```
dampedNewton(f2_,df2_,1.35)
fzero(f2_,1.35)
```

```
x=2.496959e+00,lambda:6.250000e-02
x=2.271976e+00,lambda:1
x=2.236902e+00,lambda:1
x=2.236068e+00,lambda:1

ans =

    2.2361
```

```
ans =

    2.2361
```

同样得到了与 fzero 相同的 5 位有效数字

[10]:
```
[res,count]=dampedNewton(f2_,df2_,1.35)
```

```
x=2.496959e+00,lambda:6.250000e-02
x=2.271976e+00,lambda:1
x=2.236902e+00,lambda:1
x=2.236068e+00,lambda:1

res =

    2.2361



count =

     4
```

相比朴素的牛顿法，迭代次数少了 5 次

[21]:
```
clear dampedNewton
dampedNewton(inline(4*x^4-6*x^2-11/4),inline(diff(4*x^4-6*x^2-11/4)),0.5)
```

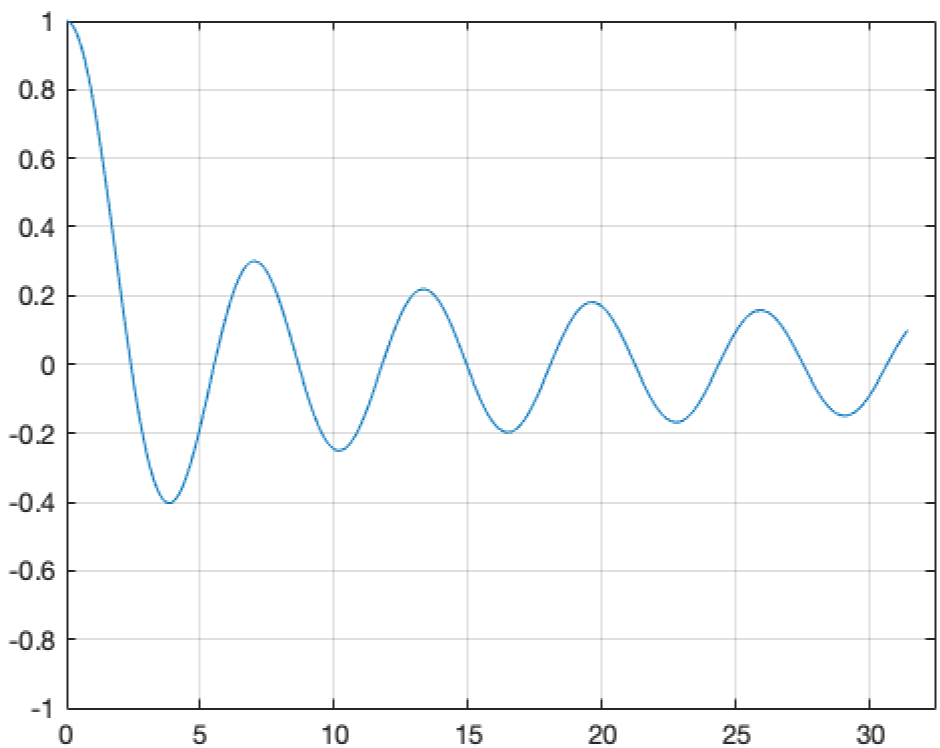
```
Error using dampedNewton (line 26)
不收敛
```

不过上面这个例子说明，阻尼牛顿法加快了收敛速度。但也不能解决牛顿法不收敛的问题，这里的函数是 $4x^4 - 6 * x^2 - 11/4$，起点是 0.5，在 0.5 到-0.5 之间反复摇摆

## 0.2 (2)

由于 fzerotx 需要给定一个区间 `[a,b]`，而且 f(a) 与 f(b) 异号，因此先了解一下 besselj 零点的情况，如图：

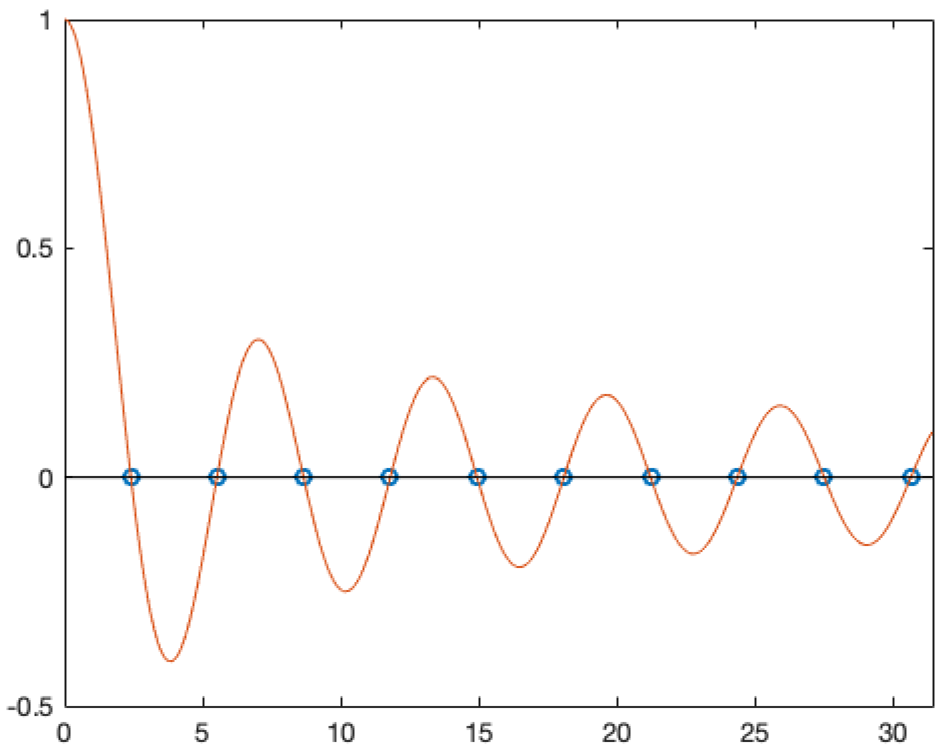
[43]:
```
X = 0:0.1:10*pi;
J = besselj(0,X);
plot(X,J)
axis([0 max(X)+1 -1 1])
grid on
```




搜了搜，知道了 besselj(0,x) 的零点和正弦差不多，而且逐渐具有周期性，因此大概 0 到 10pi 就是一个合理的求根区间，作为 fzerotx 的初始求根区间，如下：

[44]:
```
bessj0=inline('besselj(0,x)');
for n = 1:10
    z(n) = fzerotx(bessj0,[(n-1) n]*pi);
```

```
end
x = 0:pi/50:10*pi;
y = bessj0(x);
plot(z,zeros(1,10),'o',x,y,'-')
line([0 10*pi],[0 0],'color','black')
axis([0 10*pi -0.5 1.0])
```




10 个零点图如上，零点如下

[45]:
```
z
```

```
z =

  Columns 1 through 7
```

```
  2.4048    5.5201    8.6537   11.7915   14.9309   18.0711   21.2116

Columns 8 through 10

 24.3525   27.4935   30.6346
```