# 非线性方程求根实验报告

计 76 陈之杨 2017011377

## Ex. 2.2

### 题目大意

编程实现阻尼牛顿法求解方程：

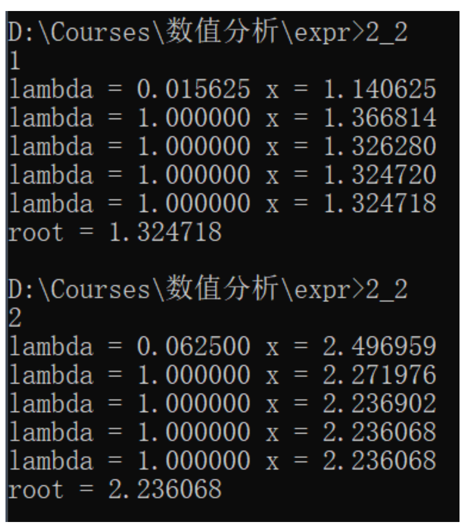
- $x^3 - x - 1 = 0,\ x_0 = 0.6.$
- $-x^3 + 5x = 0,\ x_0 = 1.35.$

### 结果分析

代码见 `ex2_2.cpp`。

初始阻尼因子为 1.0，采用逐次折半法更新。设置误差阈值 $\epsilon = 10^{-8}$。

实验结果如图所示：



```
D:\Courses\数值分析\expr>2_2
1
lambda = 0.015625 x = 1.140625
lambda = 1.000000 x = 1.366814
lambda = 1.000000 x = 1.326280
lambda = 1.000000 x = 1.324720
lambda = 1.000000 x = 1.324718
root = 1.324718

D:\Courses\数值分析\expr>2_2
2
lambda = 0.062500 x = 2.496959
lambda = 1.000000 x = 2.271976
lambda = 1.000000 x = 2.236902
lambda = 1.000000 x = 2.236068
lambda = 1.000000 x = 2.236068
root = 2.236068
```

使用 Wolfram Alpha 验证，两个方程的解分别为 $x = 1.32471796, x = 2.23606798$。运行结果与实际解相符。

观察运行时的迭代过程，第一步的 $\lambda$ 较小，而后的 $\lambda$ 均为 1。阻尼牛顿法通过启发式调整阻尼因子，避免初始值偏离准确解较远时发散或缓慢收敛的现象。事实上，第一个方程的初始值与准确解的相差超过 1，但在第一步迭代后很快接近了准确解。

# Ex. 2.3

### 题目大意

使用给出的 `fzerotx` 程序，求第一类零阶贝塞尔函数的前十个正零点。

### 结果分析

代码见 `zerotx.m` 和 `ex2_3.m`。

首先作出函数图像，观察零点位置。从零开始，对长度为 1 的区间依次调用 `fzerotx` 函数，如果有根，则记录保存。

结果如图所示：