实验7 解非线性方程的算法

# 一. 实验要求

1. 理解平分法求解非线性方程的思想；

2. 理解试位法求解非线性方程的思想；

3. 理解牛顿法求解非线性方程的思想。

# 二. 实验内容

1. 用平分法求方程x3+x-1=0的根，而且绝对误差要小于10-2。

2. 用试位法求方程x3+x-1=0的根，而且绝对误差要小于10-2。

3. 用牛顿法求方程x3+x-1=0的根，而且绝对误差要小于10-2。

# 三. 实验结果

平分法：

/\*

\*20140031 陈小龙 计科2班

\*/

#include <stdio.h>

#include <stdlib.h>

#include <float.h>

#include <math.h>

int Bisection**(**float a**,**float b**,**float eps**,**int N**){**

float x**,**fval**;**

int n**;**

**while(**n **<=** N**){**

x **=** **(**a **+** b**)/**2**;** //计算中间点

fval **=** x**\***x**\***x**+**x**-**1**;** //计算对应y值

float fa**;**

fa **=** a**\***a**\***a**+**a**-**1**;** //计算左边点的y值

**if(**fabs**(**x**-**a**)** **<** fabs**(**eps**)){** //如果满足精度要求，则返回

printf**(**"解的区间为："**);**

printf**(**"(%f,%f)\n"**,**a**,**x**);**

printf**(**"解的结果为：%f"**,**fval**);**

**return** n**;**

**}**

**if(**fval **==** 0.0**){** //如果刚好等于，则返回

printf**(**"解为：%f"**,**x**);**

printf**(**"解的结果为：%f"**,**fval**);**

**return** n**;**

**}**

**if(**fval**\***fa **<** 0.0**)** //如果中间点与左边点异号，则以中间点更新右边的点

b **=** x**;**

**else**

a **=** x**;** //否则更新左边的点

n**++;**

**}**

**return** n**;**

**}**

int main**(){**

float a **=** 0.0**,**b **=** 1.0**,**eps **=** 0.01**,**i**;**

int N**,**n**;**

i **=** **(**b**-**a**)/**eps**;**

//N = log(i)/log(2)+1;

//printf("N=%d",N);

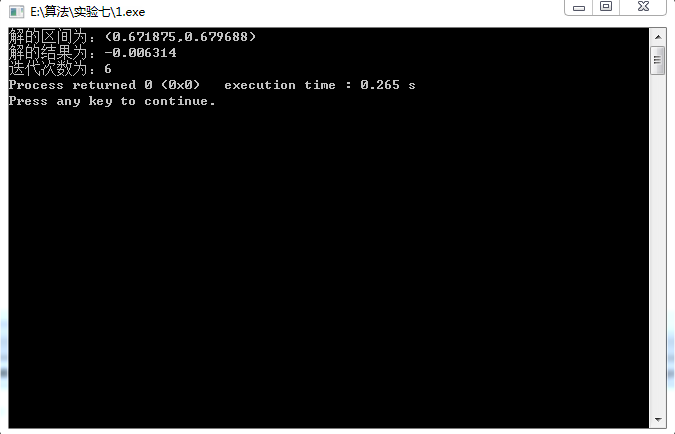
N **=** 50**;**

n **=** Bisection**(**a**,**b**,**eps**,**N**);**

printf**(**"\n迭代次数为：%d"**,**n**);**

**return** 0**;**

**}**



试位法：

/\*

\*20140031 陈小龙 计科2班

\*/

#include <stdio.h>

#include <stdlib.h>

#include <float.h>

#include <math.h>

struct point**{**

float x**;**

float y**;**

**}**a**,**b**;**

int Bisection**(**struct point a**,**struct point b**,**float eps**,**int N**){**

float x**,**fval**;**

int n**;**

**while(**n **<=** N**){**

x **=** **((**b**.**y**-**a**.**y**)\***a**.**x**/(**b**.**x**-**a**.**x**)-**a**.**y**)\*((**b**.**x**-**a**.**x**)/(**b**.**y**-**a**.**y**));** //计算截点的横坐标

fval **=** x**\***x**\***x**+**x**-**1**;** //计算以改横坐标为x值的方程的y值

float fa **=** a**.**x**\***a**.**x**\***a**.**x**+**a**.**x**-**1**;** //计算根的左边的另一点的纵坐标

**if(**fabs**(**x**-**a**.**x**)** **<** fabs**(**eps**)){** //满足精度要求则返回

printf**(**"解的区间为："**);**

printf**(**"(%f,%f)\n"**,**a**.**x**,**b**.**x**);**

printf**(**"解的结果为：%f"**,**fval**);**

**return** n**;**

**}**

**if(**fval **==** 0.0**){**

printf**(**"解的横坐标为：%f"**,**x**);**

printf**(**"解的结果为：%f"**,**fval**);**

**return** n**;**

**}**

**if(**fval**\***fa **<** 0.0**){**

a**.**x **=** x**;** //如果新的点与左边点异号，则用截点更新左边点

a**.**y **=** fval**;**

**}**

**else{**

b**.**x **=** x**;** //否则用截点更新右边点

b**.**y **=** fval**;**

**}**

n**++;**

**}**

**return** n**;**

**}**

int main**(){**

float eps **=** 0.01**;**

a**.**x **=** 0.0**,**a**.**y **=** **-**1.0**,**b**.**x **=** 1.0**,**b**.**y **=** 1.0**;**

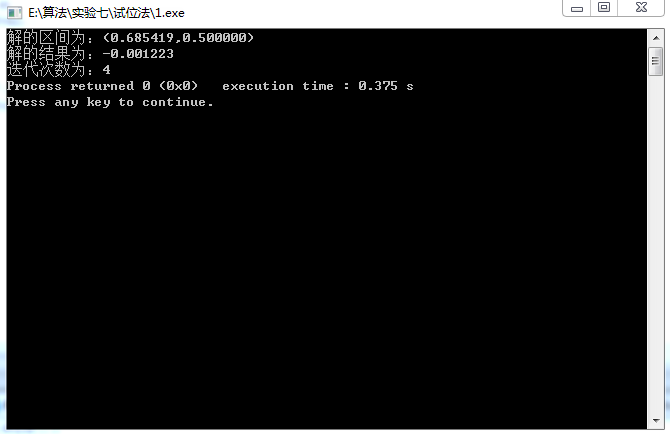
int N**,**n**;**

N **=** 50**;**

n **=** Bisection**(**a**,**b**,**eps**,**N**);**

printf**(**"\n迭代次数为：%d"**,**n**);**

**return** 0**;**

**}**

牛顿法：

/\*

\*20140031 陈小龙 计科2班

\*/

#include <stdio.h>

#include <stdlib.h>

#include <float.h>

#include <math.h>

struct point**{**

float x**;**

float y**;**

**}**a**;**

int niudun**(**struct point a**,**float eps**,**int N**){**

float fval**,**x**;**

int n **=** 1**;**

**while(**n **<=** N**){**

x **=** a**.**x**-**a**.**y**/(**3**\***a**.**x**\***a**.**x**+**1**);** //求截点的横坐标

fval **=** x**\***x**\***x**+**x**-**1**;**

**if(**fabs**(**x**-**a**.**x**)** **<** fabs**(**eps**)** **||** fval **==** 0.0**){**//刚好找到或满足精度要求就停止迭代

printf**(**"解的横坐标为：%f\n"**,**x**);**

printf**(**"解的结果为：%f"**,**fval**);**

**return** n**;**

**}**

**else{**

a**.**y **=** fval**;** //以截点的横坐标新的切点的横坐标，纵坐标

a**.**x **=** x**;**

**}**

n**++;**

**}**

**return** n**;**

**}**

int main**(){**

float eps **=** 0.01**;**

a**.**x **=** 1.0**,**a**.**y **=** 1.0**;**

int N**,**n**;**

N **=** 50**;**

n **=** niudun**(**a**,**eps**,**N**);**

printf**(**"\n迭代次数为：%d"**,**n**);**

**return** 0**;**

**}**

