实验四 用二分法求解方程实验

1. 实验名称

用二分法求解方程

1. 实验内容

# 二分法是求方程近似解的一种简单直观的方法，设函数f(x)在[a,b]上连续，且f(a)\*f(b)<0,则表明f(x)在[a,b]上至少有一个零点，然后通过二分区间，缩小区间范围，当小到一定的精确度的时候，这个x就是我们所求的近似根了．本实验为用二分法求解x^3-3\*x-1=0在区间[0,1]上的一个根

1. 实验代码

#include <stdio.h>

#include <math.h>

const double eps = 1e-6; //定义我们计算的精度

double a,b,c,d; //假定我们输入的函数是一元三次方程组，a\*x\*x\*x+b\*x\*x+c\*x+d=0

double f(double x) //定义我们的函数

{

return a\*x\*x\*x+b\*x\*x+c\*x+d;

}

int main()

{

double m,n;//求根区间[m,n]

double i,j,sum;

printf("请输入一元三次方程组的系数：a,b,c,d:");

scanf("%lf%lf%lf%lf",&a,&b,&c,&d);

printf("\n请输入求根区间[m,n]:");

scanf("%lf%lf",&m,&n);

if(f(m)\*f(n)<0)

{

while(fabs(m-n)>eps)

{

i=(m+n)/2.0;

sum=f(i);

printf("[%lf %lf]\n",m,n);

if(fabs(sum)<eps)

{

break;

//printf("\n该方程组的近似根为:x2\*=%lf\n",i);

//return 1;

}

else if(f(i)\*f(m)<0)

{

n=i; //修正区间，将[m,n]换成[m,i],这里的i是中点

}

else if(f(i)\*f(n)<0)

{

m=i;//修正区间，将[m,n]换成[i,n],这里的i是中点

}

}

}

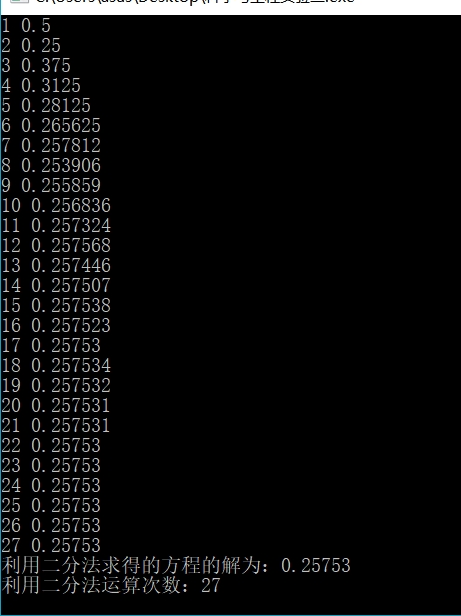
else ; //如有其他求根方法，可以加上

printf("%lf %lf",m,n);

j=(m+n)/2;

printf("\n该方程组的近似根为:x\*=%lf\n",j);

1. 实验结果截图



实验五 牛顿法

1. 实验名称

牛顿法

1. 实验内容

# 牛顿法其实是一种线性方法，其基本思想是将非线性方程逐步归结为某种线性方程来求解。本实验为用牛顿法求解x^3-3\*x-1=0在区间[0,1]上的一个根

1. 实验代码

#include <iostream>

#include<math.h>

using namespace std;

static int k=0;

static int count=1;

double f(double x)

{

return (x\*x\*x-x-1);

}

int main()

{

double x0,x1,x2;

double e=0.00001;

cout<<"请输入两个初值X0，X1"<<endl;

cin>>x0>>x1;

x2=x1-f(x1)\*(x1-x0)/(f(x1)-f(x0));

while(x2-x1>e||x1-x2>e)

{

x0=x1;

x1=x2;

x2=x1-f(x1)\*(x1-x0)/(f(x1)-f(x0));

count+=1;

}

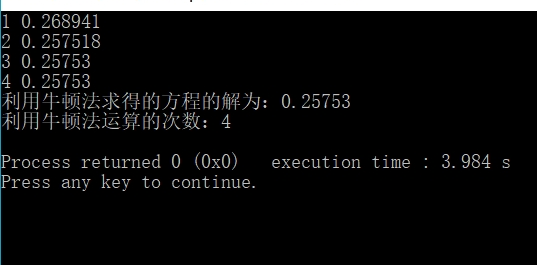
cout<<"方程的根为："<<x2<<endl;

cout<<"迭代的次数为："<<count<<endl;

return 0;

}

1. 实验结果截图



实验六 弦截法

1. 实验名称

弦截法

1. 实验内容

# 用弦截法求解x^3-3\*x-1=0在区间[0,1]上的一个根

1. 实验代码

#include <iostream>

using namespace std;

static int count=1;

double f(double x)

{

return (x\*x\*x-x-1);

}

int main()

{

double x0,x1,x2;

double e=0.00001;

cout<<"请输入两个初值X0，X1"<<endl;

cin>>x0>>x1;

x2=x1-f(x1)\*(x1-x0)/(f(x1)-f(x0));

while(x2-x1>e||x1-x2>e)

{

x0=x1;

x1=x2;

x2=x1-f(x1)\*(x1-x0)/(f(x1)-f(x0));

count+=1;

}

cout<<"方程的根为："<<x2<<endl;

cout<<"迭代的次数为："<<count<<endl;

return 0;

}

1. 实验结果截图

