牛顿迭代法

#include<iostream>

#include<cstdio>

#include<cmath>

#include<cstring>

#include<algorithm>

using namespace std;

const double eps = 1e-6;

double y;

double F(double x)

{

return 8\*x\*x\*x\*x + 7\*x\*x\*x + 2\*x\*x + 3\*x + 6;

}

double F1(double x)//F(x) 求导

{

return 32\*x\*x\*x + 21\*x\*x + 4\*x + 3;

}

double newton(double x)

{

int cnt = 1;

while(fabs(F(x) - y) > eps)

{

x = x - (F(x)-y)/F1(x);//牛顿迭代法

if(++cnt > 30)

return 0;//break;

}

return x;

}

int main()

{

int t;

cin >> t;

while(t--)

{

scanf("%lf",&y);//8\*x^4 + 7\*x^3 + 2\*x^2 + 3\*x + 6 == Y的Y值

bool flag = false;

double x;

for(double i = 0.0;i < 100;i++)

{

x = newton(i);

if(x && x >= 0.0 && x <= 100.0)

{

flag = true;

break;

}

}

if(flag)

printf("%.4f\n",x);

else

printf("No solution!\n");

}

return 0;

}

法二，首先求一下x=0-100的y值，发现是递增的，那么就可以二分了

#include<iostream>

#include<cstdio>

#include<cmath>

#include<cstring>

#include<algorithm>

using namespace std;

const double eps = 1e-6;

double y;

double cal(double x)

{

return 8\*x\*x\*x\*x + 7\*x\*x\*x + 2\*x\*x+3\*x+6;

}

int main()

{

int t;

cin >> t;

while(t--)

{

scanf("%lf",&y);//8\*x^4 + 7\*x^3 + 2\*x^2 + 3\*x + 6 == Y的Y值

double l=0,r=100;

for(int i=0;i<100;i++)

{

double mid=(l+r)/2;

double v=cal(mid);

if(v<y)l=mid;

else r=mid;

}

if(fabs(cal(r)-y)>eps)

printf("No solution!\n");

else

printf("%.4f\n",r);

}

return 0;

}