Министерство образования и науки РФ

ФГБОУ ВПО «Тамбовский государственный технический университет»

Кафедра

Отчёт по лабораторной работе № 7

по дисциплине «Вычислительная математика»

«Методы двумерного поиска»

Вариант 17

Выполнил студент гр.

.

Проверил:.

Тамбов 20

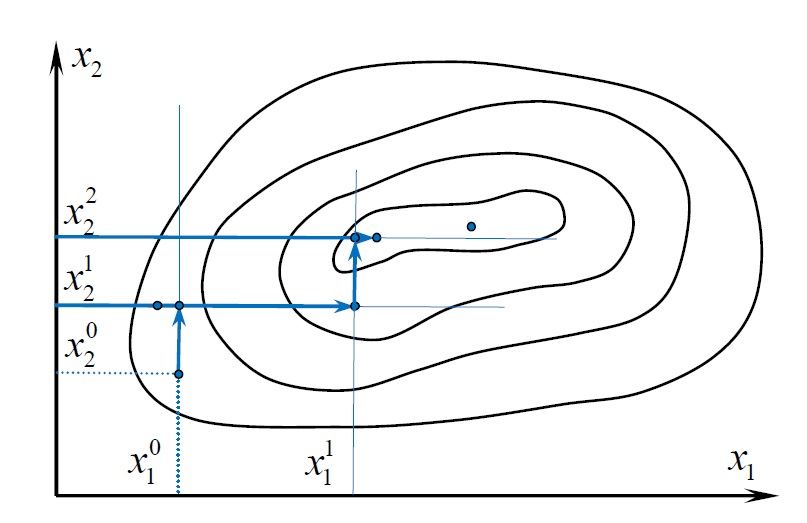
**Задача:**

Составить блок-схему алгоритма и реализовать его в программе для вычислений на ЭВМ для следующих методов одномерного поиска:

1. Метод покоординатного спуска
2. Метод градиентного спуска
3. Метод наискорейшего спуска

Для функции y= x2+3y2+xy

1. Метод покоординатного спуска:



Блок-схема метода:

Ввод h,F,*x0,,n*

f0=F(x);i=0

xi=xi+h;f1=F(x);L=0

f1<f0 нет

да

f0=f1;L++;xi=xi+h;f1=F(x)

нет

L=0

да

h=h/3

i++; i=i%n

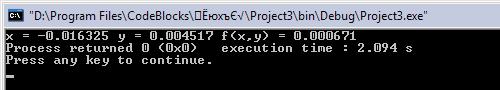
h< нет  **- Вот это действие не верно**!!!

да

Вывод x, F0,.

конец

Результат:



#include <stdio.h>

#include <math.h>

double q(double x, double y)

{

return pow(x,2)+3\*pow(y,2)+\*x\*y;

}

double dix(int i, double y)

{

double a=3, b=0, d=0.1,x1,x2,F1,F2,qmin,min,k=0;

while(fabs(a-b)>d )

{ k++;

if(k==50) break;

x1=(a+b)/2-d/2; x2=(a+b)/2+d/2;

printf("%lf %lf \n", x1,x2);

if(i==0)

{F1=q(x1,y); F2=q(x2,y);}

else {F1=q(y,x1); F2=q(y,x2);}

if(F1<F2)

{

b=x1;

qmin=F1;

min=x1;

}

else

{

a=x2;

qmin=F2;

min=x2;

}

}

return min;

}

int main(){

double h=0.1,e=0.1,x[2],p,dx,dy,lx,ly,f[50];

int k=0;

x[0]=3; x[1]=1;

f[0]=q(x[0],x[1]);

for (k=1; k<50; k++)

{

for (int i=0; i<2; i++)

{

x[0]=dix(i,x[1]); i++;

x[1]=dix(i,x[0]);

printf("%lf %lf \n",x[0], x[1]);

}

if (f[k]-f[k-1]<e)

{

printf("%lf %lf %lf", x[0], x[1], q(x[0],x[1]));

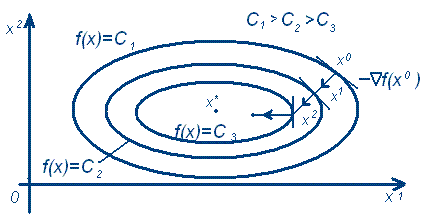
return 0;

}

}

return 0;

}

2.Метод градиентного спуска

Блок-схема метода:

Ввод F, h, *n,*

=()

i=0;i<n;i++

Вычисление

=

=()

<

Нет Да

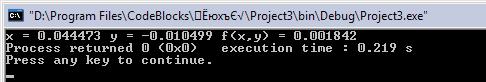
h= h/2

h <

Нет Да

Вывод x, F(x)

Результат:



#include <stdio.h>

#include <math.h>

double q(double x, double y)

{

return pow(x,2)+3\*pow(y,2)+x\*y;

}

double px(double x, double y)

{

return 2\*x+y;

}

double py(double x, double y)

{

return 6\*y+x;

}

int main(){

double h=0.1,e=0.1,x[50],f0,f1,p,dx,dy, y[50];

int k=0;

x[0]=3; y[0]=1;

p=sqrt(pow(px(x[0],y[0]),2)+pow(py(x[0],y[0]),2));

while(p>e)

{

dx=x[k];

dy=y[k];

p=sqrt(pow(px(dx,dy),2)+pow(py(dx,dy),2));

x[k+1]=x[k]-h\*px(x[k],y[k]);

y[k+1]=y[k]-h\*py(x[k],y[k]);

if(q(x[k+1],y[k+1])<q(x[k],y[k]))

{

k++;

}

else h=h/2;

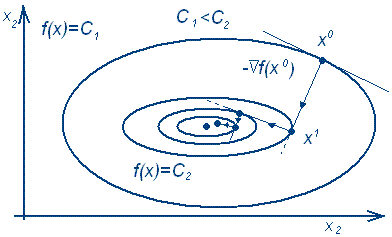
}

printf("x = %lf y = %lf f(x,y) = %lf ",x[k],y[k],q(x[k],y[k]));

return 0;

}

3.Метод наискорейшего спуска



Блок-схема метода:

Ввод F,n, h, *,*

=()

L=0

=

=()

<

L=L+1

L=0

h = h/3

h<

Вывод x,

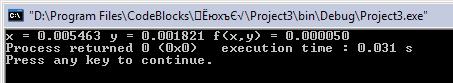
I=0;i<n;i++

Да Нет

Нет Да Нет

Да

Результат:



#include <stdio.h>

#include <math.h>

double q(double x, double y)

{

return pow(x,2)+3\*pow(y,2)+x\*y;

}

double px(double x, double y)

{

return 2\*x+y;

}

double py(double x, double y)

{

return 6\*y+x;

}

int main(){

double h=0.1,e=0.1,x[50],f0,f1,p,dx,dy, y[50],lx,ly;

int k=0;

x[0]=3; y[0]=1;

p=sqrt(pow(px(x[0],y[0]),2)+pow(py(x[0],y[0]),2));

while(p>e)

{

dx=x[k];

dy=y[k];

p=sqrt(pow(px(dx,dy),2)+pow(py(dx,dy),2));

lx=px(x[k],y[k]); ly=py(x[k],y[k]);

h=(2\*x[k]\*lx+6\*y[k]\*ly+x[k]\*ly+y[k]\*lx)/(2\*pow(lx,2)+6\*pow(ly,2)+2\*ly\*lx);

x[k+1]=x[k]-h\*px(x[k],y[k]);

y[k+1]=y[k]-h\*py(x[k],y[k]);

k++;

}

printf("x = %lf y = %lf f(x,y) = %lf ",x[k],y[k],q(x[k],y[k]));

return 0;

}