C++代码

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

double e = 0.000001;  //精度

double the0 = 0;

double the1 = 1;

double learn = 0.05;  //学习率

struct poly

{

    double x;

    double y;

};

poly Points[20];

double values[20];

void descent()

{

    double ort0 = 0, ort1 = 0;

    while(true)

    {

        for(int i=0; i<20; i++)

        {

            values[i] = the0 + the1 \* Points[i].x;

            ort0 += values[i] - Points[i].y;

            ort1 += (values[i] - Points[i].y) \* Points[i].x;

        }

        ort0 /= 20;

        ort1 /= 20;

        the0 -= learn \* ort0;

        the1 -= learn \* ort1;

        cout << the0 << " " << the1 << endl;

        if(pow(ort0,2) + pow(ort1,2) <= pow(e,2))

            break;

    }

}

int main()

{

    int a,b;

    cout << "enter the range of x" << endl;

    cin >> a >> b;

    for(int i=0; i<20; i++)

    {

        Points[i].x = (rand() % (b - a + 1)) + a;

        Points[i].y = (rand() % (b - a + 1)) + a;

    }

    descent();

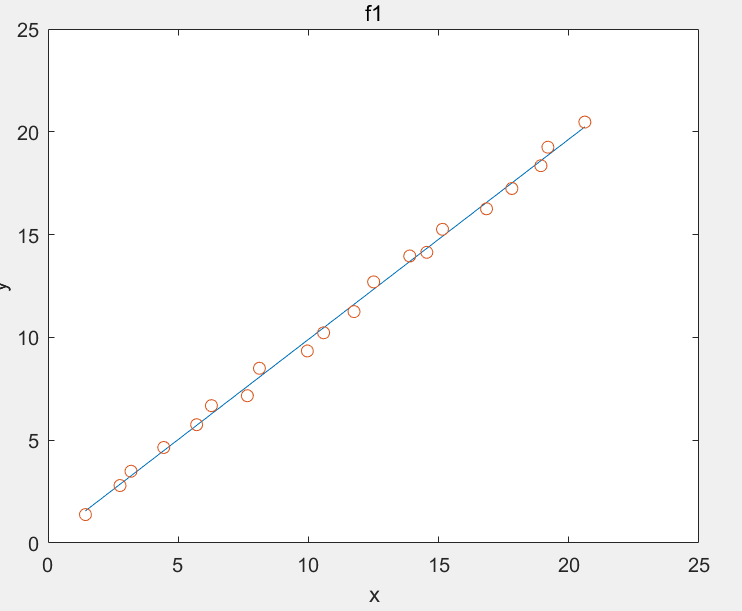
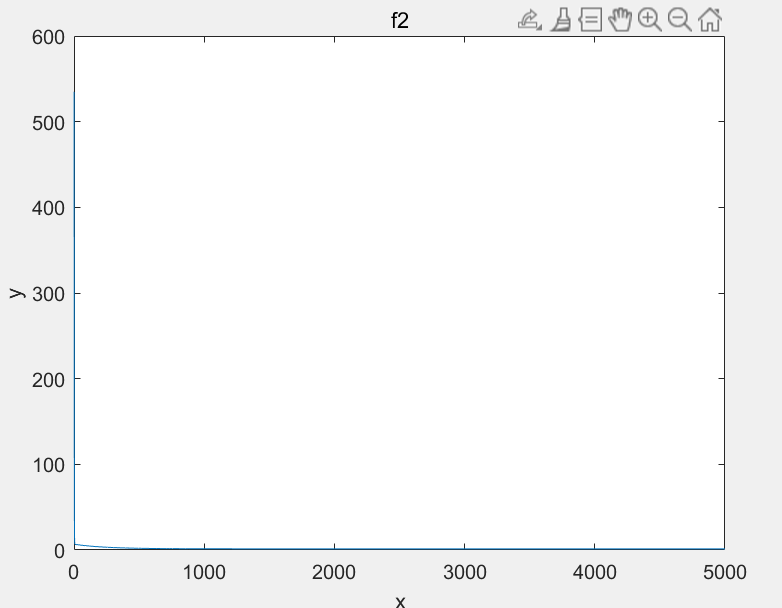
    cout << "the0: " << the0 << endl;

    cout << "the1: " << the1 << endl;

    system("pause");

    return 0;

}

Matlab 作图

散点分布及直线 方差图像

Matlab代码

X=[];Y=[];

for i=1:20

X=[X;i+rand];

Y=[Y;i+rand];

end

k=1.5;b=1.5;

learning\_rat=0.01;

epoch=5000;

Jcost=[];

for i=1:epoch

y=[];

y=b+k\*X;

Jcost=[Jcost;sum(((y-Y).^2).^1/2)];

b=b-learning\_rat\*sum((y-Y))/20;

k=k-learning\_rat\*(sum((y-Y).\*X)/20);

end

y=[];

y=b+k\*X;

figure(1);%figure1

plot(X,y);

title('f1');xlabel('x');ylabel('y');

hold on;

plot(X,Y,'o');

hold off

figure(2);%figure2

x=1:epoch;

plot(x,Jcost);

title('f2');xlabel('x');ylabel('y');