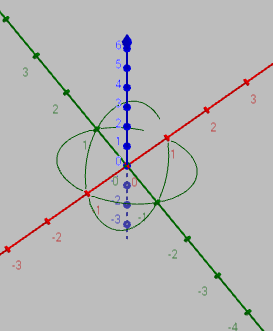
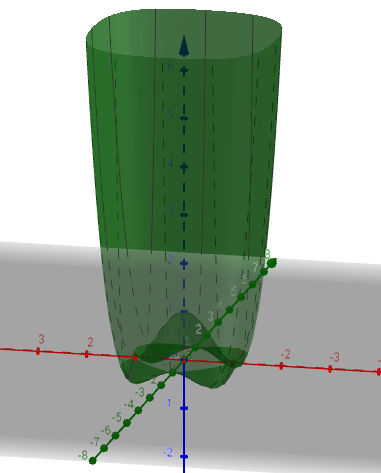
**问题描述：**

x,y分别服从N(μ1, σ12), N(μ2, σ22)的分布，向心型方程(x^2+y^2-1)^2-x^2\*y^2 = 0中投点，坐落在心型内的概率是多少，精确到0.1。



**f(x,y)=(x2 + y2 - 1)2 – x2y2 和 (x2 + y2 - 1)2 – x2y2 = 0**

**代码：**if ((pow(pow(x, 2) + pow(y, 2) - 1, 2) - pow(x, 2)\*pow(y, 2)) < 0)有问题，不能有效求心型内区域

#include <iostream>

#include <vector>

#include <numeric>

#include <limits>

#include <algorithm>

#include <iomanip>

using namespace std;

/\*\* 请完成下面这个函数，实现题目要求的功能 \*\*/

/\*\* 当然，你也可以不按照这个模板来作答，完全按照自己的想法来 ^-^ \*\*/

class Class\_getTheProbability {

private:

const int N = 1000;

const int M = 1000;//控制精度

const double mu1;

const double sigma1;

const double mu2;

const double sigma2;

double Lx;

double Ly;//正态分布系数

private:

double getLocalValue(double x, double y)

{

const double Px = Lx\*exp(-pow(x - mu1, 2) / pow(sigma1, 2) / 2);

const double Py = Ly\*exp(-pow(y - mu2, 2) / pow(sigma2, 2) / 2);

return Px\*Py;

}

double addAllValue() //得到所有的点的值的和

{

Lx = 1 / sqrt(2 \* 3.1415926) / sigma1;

Ly = 1 / sqrt(2 \* 3.1415926) / sigma2;

double allValue = 0;

for (int i = -2 \* N; i < 2 \* N; i++)

{

for (int j = -2 \* M; j < 2 \* M; j++)

{

double x = 0;

double y = 1;

if ((pow(pow(x, 2) + pow(y, 2) - 1, 2) - pow(x, 2)\*pow(y, 2)) < 0)

allValue += getLocalValue(x, y);

}

}

return allValue;

}

public:

Class\_getTheProbability(double m1, double s1, double m2, double s2) :mu1(m1), sigma1(s1), mu2(m2), sigma2(s2)

{}

double getTheProbability()

{

return addAllValue();

}

};

int main() {

double res;

//Class\_getTheProbability thesolution(0,1,0,1);

//res = thesolution.getTheProbability();

double x = 1;

double y = 0.9;

cout << (pow(pow(x, 2) + pow(y, 2) - 1, 2) - pow(x, 2)\*pow(y, 2));

cin.get();

return 0;

}