МІНІСТЕРСТВО НАУКИ І ОСВІТИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра обчислювальної техніки

**Лабораторна робота № 3**

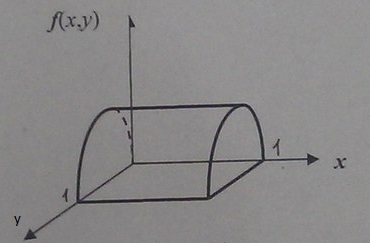
**з теорії ймовірностей**

Виконав студент групи ІО-22

Бас А. В.

Київ — 2013

**Завдання**



(в основі циліндра еліпс із параметрами a = , b = ).

Аналітичний вираз:

**Функція часткового розподілу**

## Функція умовного розподілу

## Математичне очікування

## Дисперсія

## Середньоквадратичне відхилення

## Коефіцієнт кореляції



**Лістинг програми:**

**package** edu.teover.lab03;

**import** java.util.Random;

**publicclass** Main {

**publicstaticdouble***eps* = 0.00000001;

**publicstaticint***n* = 1000;

**publicstaticvoid** main(String[] args) {

**double**[] x = **newdouble**[*n*];

**double**[] y = **newdouble**[*n*];

Random rand = **new** Random();

**for** (**int** i = 0; i <*n*; i++) {

x[i] = *getX*(rand.nextDouble());

y[i] = *getY*(rand.nextDouble());

}

**double** MX = *M*(x);

**double** MY = *M*(y);

**double** DX = *D*(x, MX);

**double** DY = *D*(y, MY);

**double** deviationX = Math.*sqrt*(DX);

**double** deviationY = Math.*sqrt*(DY);

**double** corel = *covariation*(x, MX, y, MY) / deviationX / deviationY;

System.*out*.println("MX == " + MX);

System.*out*.println("MY == " + MY);

System.*out*.println("DX == " + DX);

System.*out*.println("DY == " + DY);

System.*out*.println("deviationX == " + deviationX);

System.*out*.println("deviationY == " + deviationY);

System.*out*.println("Corelation == " + corel);

}

**publicstaticdouble** M(**double**[] a) {

**double** s = 0.0;

**for** (**int** i = 0; i < a.length; i++) {

s += a[i];

}

**return** s / a.length;

}

**publicstaticdouble** D(**double**[] a, **double** MA) {

**double** s = 0.0;

**for** (**int** i = 0; i < a.length; i++) {

s += (a[i] - MA) \* (a[i] - MA);

}

**return** s / a.length;

}

**publicstaticdouble** covariation(**double**[] x, **double** MX, **double**[] y,

**double** MY) {

**double** s = 0.0;

**for** (**int** i = 0; i < x.length; i++) {

s += (x[i] - MX) \* (y[i] - MY);

}

**return** s / x.length;

}

**publicstaticdouble** getX(**double** s) {

**return** s;

}

**privatestaticdouble** elipseIntegral( **double** x) {

// (+0.5) з'являється припідстановцілівоїграниці -0.5 у формулу,

// правушукаємобінарнимпошуком

**return** ((8.0 / Math.*PI*) \* (0.5 \* x \* Math.*sqrt*(0.25 - x \* x) + 0.125 \* Math

.*asin*(2.0 \* x))) + 0.5;

}

**publicstaticdouble** getY(**double** s) {

**double** l = -0.5;

**double** r = 0.5;

**double** m = 0.5 \* (l + r);

**double** s0 = *elipseIntegral*(m);

**while** (Math.*abs*(s - s0) >*eps*) {

m = (l + r) \* 0.5;

s0 = *elipseIntegral*(m);

**if** (s0 < s) {

l = m;

} **else** {

r = m;

}

}

// (+0.5) - зарахуноктого, щодляспрощеннязсувалиеліпс в центр

// координат

**return** m + 0.5;

}

}

**Результат виконання програми :**

MX == 0.48942576902368207

MY == 0.5048094397187233

DX == 0.08551361459002625

DY == 0.06197299952762579

deviationX == 0.29242710987530934

deviationY == 0.24894376780234084

Corelation == 0.007560863181301247