## **CE 412 – PROJECT 1**

## **“MONTE CARLO SIMULATION – RANDOM MATRICES”**

**Project Description:** In this project, our emphasis will be limited to the following question: if the elements of a 3x3 matrix are independent random numbers with positive diagonal elements and negative off-diagonal elements, what is the probability that the matrix has a positive determinant? That is find the probability that

+U11 -U12 -U13

-U21 +U22 -U23 > 0

-U31 -U32 +U33

Where the “Uij’s ” are independent random numbers uniformly distributed between 0 and 1.

In order to calculate this probability, you may write a program which generates random matrices in a loop, count the number of these matrices that have a positive determinant and print the ratio of this count to the number of replications. In order to estimate the probability with some precision, it’s reasonable to make one long run.

**Source code of the project:**  I prefer to use C# programming for this project. You’ll see the C# code below;

private void button1\_Click(object sender, EventArgs e)

{

CalculateDeterminant(3, 10000000);

}

public static void CalculateDeterminant(int size, double times)

{

Random rnd = new Random();

double[,] matrix = new double[size, size];

double det = 0;

int i, j, k;

int counter = 0;

if (size <= 0)

{

MessageBox.Show("There is no element inside of the matrix.");

}

else

{

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

{

for (i = 0; i < size; i++)

{

for (j = 0; j < size; j++)

{

if (i == j)

{

matrix[i, j] = rnd.NextDouble();

}

else

{

matrix[i, j] = rnd.NextDouble() \* (-1);

}

}

}

det = matrix[0, 0] \* (matrix[1, 1] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 1]) - matrix[0, 1] \* (matrix[1, 0] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 0]) + matrix[0, 2] \* (matrix[1, 0] \* matrix[2, 1] - matrix[1, 1] \* matrix[2, 0]);

if (det >= 0)

{

counter++;

}

}

MessageBox.Show("Number of positive determinants are : " + counter);

double ratio = (double)(counter / times);

MessageBox.Show("The ratio of positive determinant : " + ratio);

}

}

}

**Project Details:**

* I have created a method that contains 2 parameters which are array size and number of replicants. So, we are interested in 3x3 matrix so i used to this format “CalculateDeterminant(3, 100000000);” in our project.
* I need to use random number for matrix elements. So, i obtained “random” class for using random numbers.

Random rnd = new Random();

Then, I have created a 2 dimensional array to show matrix elements as index.

double[,] matrix = new double[size, size];

* Lastly, I have described some variables to use in the project. I’ll explain when their term comes.

double det = 0;

int i, j, k;

int counter = 0;

**Let’s analyze the source code step by step;**

* Firstly, I checked the size of the matrix. If we don’t have the value, we won’t calculate the determinant of the matrix.

if (size <= 0)

{

MessageBox.Show("There is no element inside of the matrix.");

}

* Then, I started to create random matrixes as number of replicants.

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

“times" variable is coming from the methods as an parameter. This parameter is double variable type because when i calculate to ratio of the positive determinant i need to get double result so double ratio = (double)(counter / times); in this formula either “counter” or “times” should be double to get double ratio.

After that, I started to locate each value to the each element of matrix.

for (i = 0; i < size; i++)

{

for (j = 0; j < size; j++)

{

* Then, I controlled the index of each element because the matrix should be positive diagonal elements and negative off-diagonal elements. If we have U11,U22 or U33, I multiplied by minus 1.

if (i == j)

{

matrix[i, j] = rnd.NextDouble();

}

else

{

matrix[i, j] = rnd.NextDouble() \* (-1);

}

* Calculation of determinant;

det = matrix[0, 0] \* (matrix[1, 1] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 1]) - matrix[0, 1] \* (matrix[1, 0] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 0]) + matrix[0, 2] \* (matrix[1, 0] \* matrix[2, 1] - matrix[1, 1] \* matrix[2, 0]);

* The determinant of the matrix has

a[0][0]\*((a[1][1]\*a[2][2]) - (a[2][1]\*a[1][2])) -a[0][1]\*(a[1][0]\*a[2][2] - a[2][0]\*a[1][2]) + a[0][2]\*(a[1][0]\*a[2][1] - a[2][0]\*a[1][1]);

**for 3x3 matrix.**

+a00 -a01 -a02

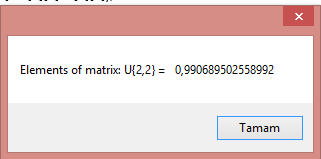
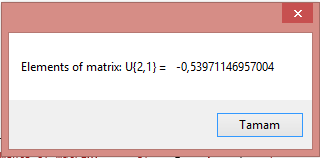
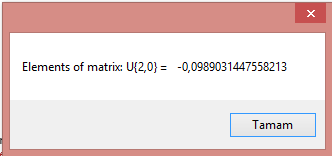
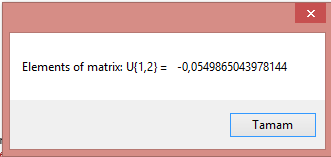
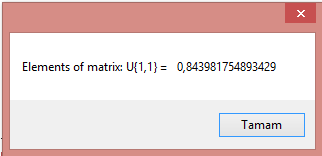
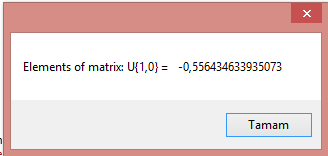
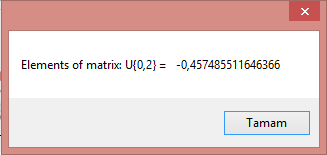
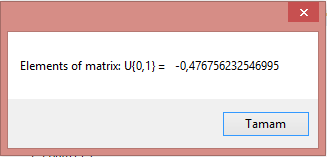
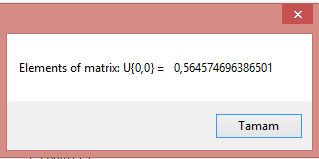
-a10 +a11 -a12

-a20 -a21 +a22

* I checked the elements of matrix with this code

MessageBox.Show("Elements of matrix: " + "U{" + i + "," + j + "} = " + matrix[i, j] + "\n");

* However I just want to show sign of the elements so i saw them and i didn’t include the code because it would be in a loop so we would get lots of elements everytime.



This is just one matrix to be controlled.

* After, I controlled the value of determinant in terms of positive or negative? And, I counted the number of positive determinants calculating the ratio.

if (det > 0)

{

counter++;

}

* Lastly,

I used to “messagebox” to show the related value. I prefer to show percentage of the “number of positive determinant” inside of the replicants.

MessageBox.Show("Number of positive determinants are : " + counter);

double ratio = (double)(counter / times);

MessageBox.Show("The ratio of positive determinant : " + ratio);

My result is converge to “ 0.0502” approximately.

