Assignment 3: Testing Random Number Generators

1 Introduction

In this assignment, you will create a *linear congruential generator* and perform four empirical tests on the generated random numbers:

- Uniformity test
- Series test
- Runs test
- Correlation test

2 Generation of Random Numbers

The equation to generate random numbers is as follows:

$$Z_i = (65539 * Z_{i-1})\ mod\ 2^{31}$$

and

$$U_i = \frac{Z_i}{2^{31}}$$

Where Z_0 (or the *seed*) would be **your 7 digit student id**. For each test you have to generate n random numbers first and then perform the tests. n should be kept as a parameter in your code and we would check your code with different values of n. Note that you will have to generate n values first in each execution of each test.

3 Uniformity Test

In this test, you will check whether the U_i's generated appear to be uniformly distributed or not. You will perform *chi-square test* with all parameters known. First divide [0,1] into k sub-intervals. Then if f_j is the number of U_i's that are in the j^{th} sub-interval, χ^2 can be calculated as:

$$\chi^2 = \frac{k}{n} \sum_{j=1}^{k} \left(f_j - \frac{n}{k} \right)^2$$

For large n, χ^2 will have an approximate chi-square distribution with $k-1$ degrees of freedom under the null hypothesis:

$$H_0 = U_i\text{'s are IID random variables.}$$

We reject this hypothesis at level α if $\chi^2 > {\chi^2}_{k-1,1-\alpha}$ where ${\chi^2}_{k-1,1-\alpha}$ is the upper $1-\alpha$ critical point of the chi-square distribution with $k-1$ degrees of freedom. k and α should be kept as a parameter in your code. Test the null hypothesis and report if it is rejected or not.

(* To calculate ${\chi^2}_{df,q}$ in your code, in python you can use the library function *scipy.stats.chi2.ppf(q,df)* where q is the level and df is the degrees of freedom. We will share a stub code for your convenience. Other language users can use any built in function to calculate this value or keep a file containing a table for these values generated from the code.)

4 Serial Test

In this test you will calculate d-tuples where $\mathbf{U_1} = (U_1,\ U_2,\ ...,\ U_d)$, $\mathbf{U_2} = (U_{d+1},\ U_{d+2},\ ...,\ U_{2d})$ and so on. You will check whether d-tuples are uniformly distributed on the d-dimensional unit hypercube $[0,1]^d$. Divide [0,1] into k sub-intervals and generate $\mathbf{U_1},\ \mathbf{U_2},\ ...,\ \mathbf{U_l}$ where $l = \lfloor \frac{n}{d} \rfloor$. Let $f_{j_1 j_2 \dots j_d}$ be the number of U_i's having first component at j_1, second component at j_2 etc. Calculate

$$\chi^2(d) = \frac{k^d}{n} \sum_{j_1=1}^{k} \sum_{j_2=1}^{k} \dots \sum_{j_d=1}^{k} \left(f_{j_1 j_2 \dots j_d} - \frac{n}{k^d} \right)^2$$

Then $\chi^2(d)$ will have an approximate chi-square distribution with $k^d - 1$ df. Test the null hypothesis as in the uniformity test and report it is rejected or not. Keep k, α and d as parameters in your code.

5 Runs Test

In this test you will evaluate the independence assumption of the random number generator. You will examine the sequence of U_i's generated and check for unbroken sequences of maximal length within which U_i's increase monotonically (or a *run up*). From a sequence of n U_i's count the number of runs up of length 1, 2, 3, 4, 5 and ≥ 6 and then define:

$$r_i = \begin{cases} \text{no. of runs of length i} & \text{For } 1 < i < 6 \\ \text{no. of runs of length } \geq 6 & \text{For } i = 6 \end{cases} \tag{1}$$

Then calculate the test statistic:

$$R = \frac{1}{n} \sum_{i=1}^{6} \sum_{j=1}^{6} a_{ij} (r_i - nb_i)(r_j - nb_j)$$

where a_{ij} is the (i, j)th element of the matrix

$$\begin{bmatrix} 4{,}529.4 & 9{,}044.9 & 13{,}568 & 18{,}091 & 22{,}615 & 27{,}892 \\ 9{,}044.9 & 18{,}097 & 27{,}139 & 36{,}187 & 45{,}234 & 55{,}789 \\ 13{,}568 & 27{,}139 & 40{,}721 & 54{,}281 & 67{,}852 & 83{,}685 \\ 18{,}091 & 36{,}187 & 54{,}281 & 72{,}414 & 90{,}470 & 111{,}580 \\ 22{,}615 & 45{,}234 & 67{,}852 & 90{,}470 & 113{,}262 & 139{,}476 \\ 27{,}892 & 55{,}789 & 83{,}685 & 111{,}580 & 139{,}476 & 172{,}860 \end{bmatrix}$$

and the b_i's are given by

$$(b_1, b_2, \ldots, b_6) = \left(\frac{1}{6}, \frac{5}{24}, \frac{11}{120}, \frac{19}{720}, \frac{29}{5040}, \frac{1}{840}\right)$$

The values of a_{ij}'s and b_i's are shown in the Figure above.

For large n, R will have an approximate chi-square distribution with 6 df. Calculate if $R \geq {\chi^2}_{6,1-\alpha}$ and if so reject the null hypothesis as in previous test. Keep α as parameter in your code.

6 Correlation Test

In this test you will directly assess whether the generated U_i's exhibit discernible correlation at lag j. Generate a sequence of n U_i's U_1, U_2, ..., U_n and then estimate $\hat{\rho_j}$ as:

$$\hat{\rho_j} = \frac{12}{h+1} \sum_{k=0}^{h} U_{1+kj} U_{1+(k+1)j} - 3$$

where $h = \lfloor \frac{n-1}{j} - 1 \rfloor$. Then calculate $Var(\hat{\rho_j})$ as:

$$Var(\hat{\rho_j}) = \frac{13h+7}{(h+1)^2}$$

Finally calculate:

$$A_j = \frac{\hat{\rho_j}}{\sqrt{Var(\hat{\rho_j})}}$$

Under the assumption that there is no correlation (that is $\rho_j = 0$) and assuming n is large A_j should have an approximate standard normal distribution. Test the null hypothesis:

$$H_0 : U_i\text{'s have zero lag j correlation}$$

at level α, reject the hypothesis if $|A_j| > z_{1-\alpha/2}$. Here j and α should be kept as parameters.

(* To calculate z_α in your code, in python you can use the library function *scipy.stats.norm.ppf(α)* where α is the level. We will share a stub code for your convenience. Other language users can use any built in function to calculate this value or keep a file containing a table for these values generated from the code.)

7 Report Writing

You will have to create a consolidated report showing the generated results for different values of n and test-wise parameters **(For both lab and theory assignment)**. You also have to write down a *detailed* analysis of the generated results **(For theory assignment only)**.

You will calculate the values for n = 20, 500, 4000 and 10000. For each n, perform 10 tests in total:

- Uniformity test at $k = 10$ and $k = 20$
- Serial test using $d = 2,\ 3$ and $k = 4,\ 8$
- Run length test
- Correlation test with j =1, 3 and 5.

The value of α for all tests will be 0.1 .

8 Submission and Deadline

Deadline is set at **Saturday, 12 December 2020 at 11:59pm.**

We will create two separate submission links for code and report submission respectively.

For code submission, place all your source codes in a folder, rename the folder as your 7 digit student ID, zip the folder and submit it.

For the report submission, create a pdf document. **Rename the pdf as your 7 digit student ID (e.g. 1505003.pdf). Also the report inside should contain your name and roll number in the front page.** Submit this pdf file in the link.

Copying is strictly prohibited and if found will be dealt harshly.