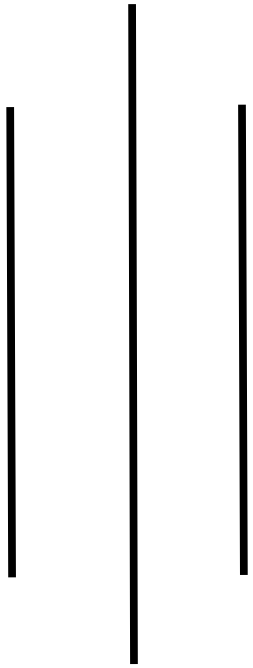
Deerwalk Institute of Technology

Sifal, Kathmandu



Simulation and Modelling Practical

Submitted By:
Name: Sagar Giri
Roll No: 205
Section: A

Submitted To:
Binod Sitaula

Date:

Background Theory

Kolmogorov-Smirnov Test

This test compares the continuous cdf (cumulative distribution function) F(x) of the uniform distribution, with the empirical cdf, sn(xi) for the sample of N random numbers.
The largest absolute derivation between F(x) and Sn(x) is determined and is compared with the critical, value which is available as function of N in the KS-table for various level of confidence.

At first, we generate random numbers using Linear Congruential method and test the uniformity of the generated random numbers using KS-Test.

Background theory for Linear Congruential method:

It is the most commonly used technique for generating random numbers and was initially purposed by Lehmer in 1951. This technique provides a sequence of integer between 0 and 1 by following recursive relation:

$$x_{(i+1)} = (a * X_i + c) \bmod m \quad \text{for i=0,1,2,.....}$$

where Xo = Seed
a = multiplier
c = increment
m = modulus

If c≠0, then the method is called mixed congruential method, else it is called multiplicative Congruential method.

Algorithm for KS-Test:

1. Rank the data from smallest to largest. If R(i) denotes the ith smallest observation, R(i)<R(2)<R(3)<.....R(N).
2. Compute:

$$D+ = max\left(\frac{i}{N} - Ri\right)$$

$$D- = max\left(Ri - \frac{(i-1)}{N}\right)$$

3. Compute: D = max of (D+, D-)
4. Determine the critical value D(alpha) from KS table for the specified level of significance and given sample size N.
5. If D > D(alpha), null hypothesis is rejected else null hypothesis is accepted which says that the data is uniformly distributed.

Program Coding:

```
#include <stdio.h>
#include <math.h>
#define MAX_NO 500
int main()
{
	//linear congruental method to generate random numbers:
	int a,m,c=0,i,x,j,flag=0,N=0, choice;
	float rand[MAX_NO], temp;
	printf("Mixed (1) or Multiplicative (2)?\n");
	scanf("%d",&choice);
	if(choice == 1)
	{
		printf("Enter the value of c: ");
		scanf("%d",&c);
	}
	printf("Enter the value of x0, a & m\n");
	scanf("%d%d%d",&x,&a,&m);
	while(1) {
		rand[i] = (float)x/m;

		for(j=0;j<=i;j++) {
			if(i==j) {
				continue;
			}
			else if(rand[i] == rand[j]) {
				flag = 1;
				break;
			}
			else{
				flag = 0;
			}
		}
		x = (a*x+c)%m;
		N++; i++;
		if(flag ==1){
			break;
		}
	}
	printf("\nno of randoms generated= %d",N);


	//Rank the random numbers in increasing order
	for(i=0;i<N;i++){
		for(j=0;j<N-i-1;j++){
			if(rand[j]>rand[j+1]){
				temp=rand[j];
				rand[j]=rand[j+1];
				rand[j+1]=temp;
			}
		}
```

```
	}

	//Display the random numbers
	printf("\n");
	for(i=0;i<N;i++){
		printf("%f ",rand[i]);
	}

	//BEGIN OF K-S test
	//setup the hypothesis
	printf("\n\nHypothesis\n");
	printf("H0 : Numbers are uniformally distributed\n");
	printf("H1 : Numbers are not uniformally distributed\n");

	float tempX[MAX_NO], tempY[MAX_NO];
	float Dpo, Dne, D, Dalpha;

	for(i = 0;i<N;i++){
		tempX[i] = (float)(i/N) - rand[i];
		tempY[i] = rand[i] - (float)((i-1)/N);
	}
	Dpo = tempX[0];
	for (i = 0; i<10; i++){
		if (tempX[i]>Dpo){
			Dpo = tempX[i];
		}
	}
	Dne = tempY[0];
	for (i = 0; i<10; i++){
		if (tempY[i]>Dne){
			Dne = tempX[i];
		}
	}

	//Find the actual value of D
	D = (Dpo > Dne ? Dpo: Dne);

	printf("\nEnter critical value at 0.05 level of significance from
table: ");
	scanf("%f",&Dalpha);

	printf("\n\n\t\tConclusion\t\t\n");
	printf("\n\n");

	printf("D-calculated=%.4f and D-tabulated=%.4f",D,Dalpha);
	printf("\n\n");

	if(D<=Dalpha){
		printf("D-calculated <= D-tabulated. \nNull Hypothesis H0 is
accepted and numbers are uniformally distributed.\n");
	}
```

```
      else{
            printf("D-calculated > D-tabulated.\nNull Hypothesis H0 is
rejected and numbers are not uniformally distributed.\n");
      }
      return 0;
}
```

Output:

```
Mixed (1) or Multiplicative (2)?
2
Enter the value of x0, a & m
118 45 1000

no of randoms generated= 5
0.118000 0.310000 0.750000 0.750000 0.950000

Hypothesis
H0 : Numbers are uniformally distributed
H1 : Numbers are not uniformally distributed

Enter critical value at 0.05 level of significance from table: 0.565


                Conclusion


D-calculated=0.0000 and D-tabulated=0.5650

D-calculated <= D-tabulated.
Null Hypothesis H0 is accepted and numbers are uniformally distributed.


------------------
(program exited with code: 0)
Press return to continue
```

Conclusion:

Hence, 5 random numbers were generated using linear congruential method and Kolmogorov Smirnov test was done which verified the uniform distribution of the random numbers.