

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Electrical, Electronic and Computer Engineering

ESC 320

Special Assignment 1

Random Number Generator, Probability Density Function

7 August 2015

Name	Student no.	Discipline	Signature
Botha, F.	29042951	Electronic	

FORM A

UNIVERSITY OF PRETORIA

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND IT

DEPARTMENT OF ELECTRICAL, ELECTRONIC AND COMPUTER ENGINEERING

The **Department of electrical, electronic and computer engineering** places specific emphasis on integrity and ethical behavior with regard to the preparation of all written work to be submitted for academic evaluation.

Although academic personnel will provide you with information regarding reference techniques as well as ways to avoid plagiarism, you also have a responsibility to fulfill in this regard. Should you at any time feel unsure about the requirements, you must consult the lecturer concerned before you submit any written work.

You are guilty of plagiarism when you extract information from a book, article or web page without acknowledging the source and pretend that it is your own work. In truth, you are stealing someone else's property. This doesn't only apply to cases where you quote verbatim, but also when you present someone else's work in a somewhat amended format (paraphrase), or even when you use someone else's deliberation without the necessary acknowledgement. You are not allowed to use another student's previous work. You are furthermore not allowed to let anyone copy or use your work with the intention of presenting it as his/her own.

Students who are guilty of plagiarism will forfeit all credit for the work concerned. In addition, the matter can also be referred to the Committee for Discipline (Students) for a ruling to be made. Plagiarism is considered a serious violation of the University's regulations and may lead to suspension from the University.

For the period that you are a student at the **Department of electrical, electronic and computer engineering**, the under-mentioned declaration must accompany all written work to be submitted. No written work will be accepted unless the declaration has been completed and attached.

I (full names) **F BOTHA**

Student number **29042951**

Subject of the work **Special Assignment 1 - Random Number Generator, Probability Density Function**

Declaration

1. I understand what plagiarism entails and am aware of the University's policy in this regard.
2. I declare that this **REPORT** (e.g. essay, report, project, assignment, dissertation, thesis etc) is my own, original work. Where someone else's work was used (whether from a printed source, the internet or any other source) due acknowledgement was given and reference was made according to departmental requirements.
3. I did not make use of another student's previous work and submitted it as my own.
4. I did not allow and will not allow anyone to copy my work with the intention of presenting it as his or her own work.

Signature _____

Contents

1. Introduction and Aim....4
2. Method and Implementation....4
3. Results....7
4. Discussion and Conclusion....13
5. References....13

Tables

Table 1 Results....7

Figures

Figure 1 Wichmann-Hill Sequence....11
Figure 2 Marsaglia Sequence....11
Figure 3 Experimental uniform distribution....12
Figure 4 Experimental normal distribution 12

1. Introduction and Aim

Random number sequences are essential to modern digital communications. Although no true random sequences can be created many techniques exist to generate pseudo random sequence. In this assignment the aim is to study the real world performance of some methods of generating random sequences. Random sequences are not always evenly distributed but can be created to resemble some real world scenarios for example; Gaussian white noise. In this study a sequence of a normal and uniform distribution are considered.

A practical algorithm for achieving a uniform distribution was described by Wichman and Hill (1982). This algorithm persisted for its high degree of portability originally written in BASIC but easily implemented in any language. They further described how to improve the algorithm for machines supporting longer integer words.

The Marsaglia polar method (Marsaglia and Bray) is a particularly robust and flexible method to achieve a normal, or Gaussian, distribution. It is easily adjusted to a specific mean and standard deviation.

2. Method and Implementation

The random number generators where implemented in C++ and the results saved in a text file. This file was subsequently imported into Microsoft Excel for processing.

The implementation of the Wichman-Hill generator in C++ is shown next. The method makes use of the long integer word method to make maximum use of modern 64-bit computing.

```
#include "stdafx.h"
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <ctime>
#include <cmath>

using namespace std;

double myRandom(long long int *s1,long long int *s2,long long int *s3);


int _tmain(int argc, _TCHAR* argv[])
{
	long long int N1 = time(NULL);
	long long int N2 = time(NULL);
	long long int N3 = time(NULL);
	double TheRandom1 = 0;
```

```
	ofstream myfile;
	myfile.open("RandomNumbers.txt");

	for (size_t i = 0; i < 100001; i++)
	{
		TheRandom1 = myRandom(&N1, &N2, &N3);
		cout <<i<<"="<< TheRandom1<<endl;
		myfile << TheRandom1 << endl;
	}
	myfile.close();
	system("pause");
	return 0;
}

double myRandom(long long int *s1,long long int *s2,long long int *s3)
{
	double value;

	*s1 = ((171 * *s1) % 30269);
	*s2 = ((172 * *s2) % 30307);
	*s3 = ((170 * *s3) % 30323);

	value = fmod((double(*s1) / 30269
		+ double(*s2) / 30307
		+ double(*s3) / 30323), 1);
	return value;
}
```

The C++ Implimentation of the Marsaglia polar method is shown below for a Random sequence centered about 0.5 with a Standard deviation of 0.2.

```
#include "stdafx.h"
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <ctime>
#include <cmath>

using namespace std;

double generateGaussianNoise(const double &mean, const double &variance);

int _tmain(int argc, _TCHAR* argv[])
{
	const double N1 = 0.5;
	const double N2 = 0.2;
	double TheRandom1 = 0;
```

```
	ofstream myfile;
	myfile.open("RandomNumbers.txt");

	for (size_t i = 0; i < 100001; i++)
	{
		TheRandom1 = generateGaussianNoise(N1,N2);
		cout << i << "=" << TheRandom1 << endl;
		myfile << TheRandom1 << endl;
	}
	myfile.close();
	system("pause");
	return 0;
}

	double generateGaussianNoise(const double& mean, const double &variance)
	{
		static bool hasSpare = false;
		static double spare;

		if (hasSpare)
		{
			hasSpare = false;
			return mean + variance * spare;
		}

		hasSpare = true;
		static double u, v, s;
		do
		{
			u = (rand() / ((double)RAND_MAX)) * 2.0 - 1.0;
			v = (rand() / ((double)RAND_MAX)) * 2.0 - 1.0;
			s = u * u + v * v;
		} while ((s >= 1.0) || (s == 0.0));

		s = sqrt(-2.0 * log(s) / s);
		spare = v * s;
		return mean + variance * u * s;
    }
```

3. Results

The results are summarized in the table that follows. Just a extract of the random sequence is shown to give a feel as the actual data analyzed is several pages long.

Table 1 Results

Wichman-Hill	Marsaglia polar method	X	Number Wichman-Hill	Number Theoretical Uniform	Number Marsaglia	Theoretical normal distribution
		0	0	0	0	87.64150247
0.927457	0.394958	0.01	998	1000	98	99.18677196
0.340299	0.605742	0.02	1004	1000	124	111.9726515
0.162264	1.01412	0.03	1008	1000	109	126.0910996
0.652309	0.378277	0.04	966	1000	124	141.6351887
0.201039	0.44228	0.05	1013	1000	159	158.6982592
0.010192	0.652662	0.06	981	1000	163	177.3729642
0.070783	0.644846	0.07	1005	1000	173	197.7502079
0.829083	0.610642	0.08	945	1000	227	219.9179799
0.053226	0.452781	0.09	1013	1000	234	243.9600929
0.349237	0.552008	0.1	956	1000	241	269.9548326
0.511001	0.870837	0.11	1025	1000	299	297.9735303
0.607357	0.523844	0.12	1000	1000	307	328.0790739
0.078096	0.35312	0.13	943	1000	323	360.3243717
0.29463	0.451274	0.14	1032	1000	406	394.7507915
0.885187	0.450728	0.15	1030	1000	444	431.3865941
0.419639	0.453325	0.16	1055	1000	421	470.2453869
0.373043	0.55195	0.17	1012	1000	466	511.3246228
0.596329	0.494225	0.18	992	1000	542	554.6041734
0.180888	0.721653	0.19	991	1000	564	600.0450035
0.624384	0.998318	0.2	1013	1000	591	647.5879783
0.867925	0.804088	0.21	1029	1000	673	697.1528322
0.021331	0.820229	0.22	1048	1000	722	748.6373282
0.055373	0.304076	0.23	1025	1000	756	801.9166367
0.537443	0.595709	0.24	951	1000	870	856.8429602
0.015183	0.363633	0.25	965	1000	877	913.2454269

0.786959	0.090223	0.26	930	1000	913	970.9302749
0.013325	0.38184	0.27	956	1000	978	1029.681344
0.047	0.528725	0.28	1001	1000	1065	1089.260885
0.764209	0.618362	0.29	1000	1000	1163	1149.410703
0.875259	0.62642	0.3	952	1000	1177	1209.853623
0.425412	0.538347	0.31	1008	1000	1272	1270.295282
0.971251	0.118029	0.32	1072	1000	1243	1330.426249
0.672561	0.623542	0.33	1021	1000	1295	1389.924431
0.19375	0.574484	0.34	1034	1000	1393	1448.457764
0.047413	0.525093	0.35	987	1000	1410	1505.687161
0.977937	0.271941	0.36	1020	1000	1568	1561.269667
0.060884	0.455118	0.37	985	1000	1579	1614.861798
0.20876	0.186729	0.38	1018	1000	1614	1666.123014
0.625676	0.584977	0.39	1021	1000	1656	1714.719275
0.466049	0.431876	0.4	995	1000	1753	1760.326634
0.926011	0.587785	0.41	979	1000	1803	1802.634812
0.423872	0.607874	0.42	1030	1000	1901	1841.350702
0.863091	0.811611	0.43	1007	1000	1896	1876.201735
0.367252	0.199392	0.44	1007	1000	1752	1906.939077
0.651029	0.290655	0.45	1062	1000	2031	1933.340584
0.748855	0.318475	0.46	1061	1000	1983	1955.21347
0.019918	0.453293	0.47	1000	1000	1897	1972.396655
0.252745	0.346721	0.48	1018	1000	1948	1984.762737
0.310867	0.53157	0.49	1011	1000	1967	1992.21957
0.576258	0.420937	0.5	990	1000	2015	1994.711402
0.700775	0.31169	0.51	1010	1000	1925	1992.21957
0.582219	0.309177	0.52	999	1000	1917	1984.762737
0.535639	0.655461	0.53	954	1000	1955	1972.396655
0.319835	0.838095	0.54	971	1000	1950	1955.21347
0.892406	0.651097	0.55	991	1000	2090	1933.340584
0.295454	0.601366	0.56	1005	1000	1947	1906.939077
0.569801	0.485169	0.57	973	1000	1856	1876.201735

0.906445	0.209849	0.58	1027	1000	1947	1841.350702
0.49547	0.83431	0.59	1021	1000	1835	1802.634812
0.015129	0.456738	0.6	943	1000	1750	1760.326634
0.554343	0.488044	0.61	1003	1000	1748	1714.719275
0.544759	0.627739	0.62	946	1000	1687	1666.123014
0.470423	0.559755	0.63	1031	1000	1631	1614.861798
0.536145	0.404241	0.64	1012	1000	1659	1561.269667
0.388374	0.785501	0.65	1026	1000	1618	1505.687161
0.948221	0.169196	0.66	1001	1000	1439	1448.457764
0.787388	0.79428	0.67	968	1000	1432	1389.924431
0.934454	0.636449	0.68	1013	1000	1349	1330.426249
0.472088	0.699584	0.69	963	1000	1300	1270.295282
0.015457	0.117783	0.7	1043	1000	1160	1209.853623
0.129124	0.391589	0.71	1001	1000	1209	1149.410703
0.021584	0.414472	0.72	1018	1000	1102	1089.260885
0.54412	0.434099	0.73	990	1000	1048	1029.681344
0.885988	0.76668	0.74	911	1000	1020	970.9302749
0.917608	0.501954	0.75	1013	1000	955	913.2454269
0.835499	0.555986	0.76	970	1000	948	856.8429602
0.039792	0.747471	0.77	981	1000	817	801.9166367
0.04534	0.348064	0.78	970	1000	707	748.6373282
0.043442	0.298647	0.79	1030	1000	743	697.1528322
0.889512	0.595688	0.8	964	1000	677	647.5879783
0.168961	0.497265	0.81	1027	1000	582	600.0450035
0.629672	0.352655	0.82	958	1000	585	554.6041734
0.480908	0.882973	0.83	976	1000	495	511.3246228
0.274704	0.509215	0.84	959	1000	538	470.2453869
0.634409	0.71596	0.85	1010	1000	476	431.3865941
0.123287	0.279495	0.86	990	1000	437	394.7507915
0.211462	0.475026	0.87	1039	1000	358	360.3243717
0.516525	0.353906	0.88	1022	1000	385	328.0790739
0.583228	0.748039	0.89	983	1000	282	297.9735303

0.505156	0.563266	0.9	1069	1000	290	269.9548326
0.606144	0.406546	0.91	974	1000	275	243.9600929
0.524288	0.402471	0.92	1007	1000	257	219.9179799
0.944008	0.768128	0.93	1023	1000	202	197.7502079
0.044251	0.479091	0.94	1019	1000	190	177.3729642
0.490418	0.255851	0.95	964	1000	205	158.6982592
0.097635	0.50278	0.96	1056	1000	154	141.6351887
0.553283	0.808028	0.97	1007	1000	130	126.0910996
0.82921	0.374662	0.98	1063	1000	125	111.9726515
0.863145	0.16434	0.99	992	1000	101	99.18677196
0.607904	0.604683	1	939	1000	101	87.64150247

Plotted below Are the first 100 Elements of each sequence, it is clear from the trend line that the longer the sequence gets the closer it gets to the expected average.

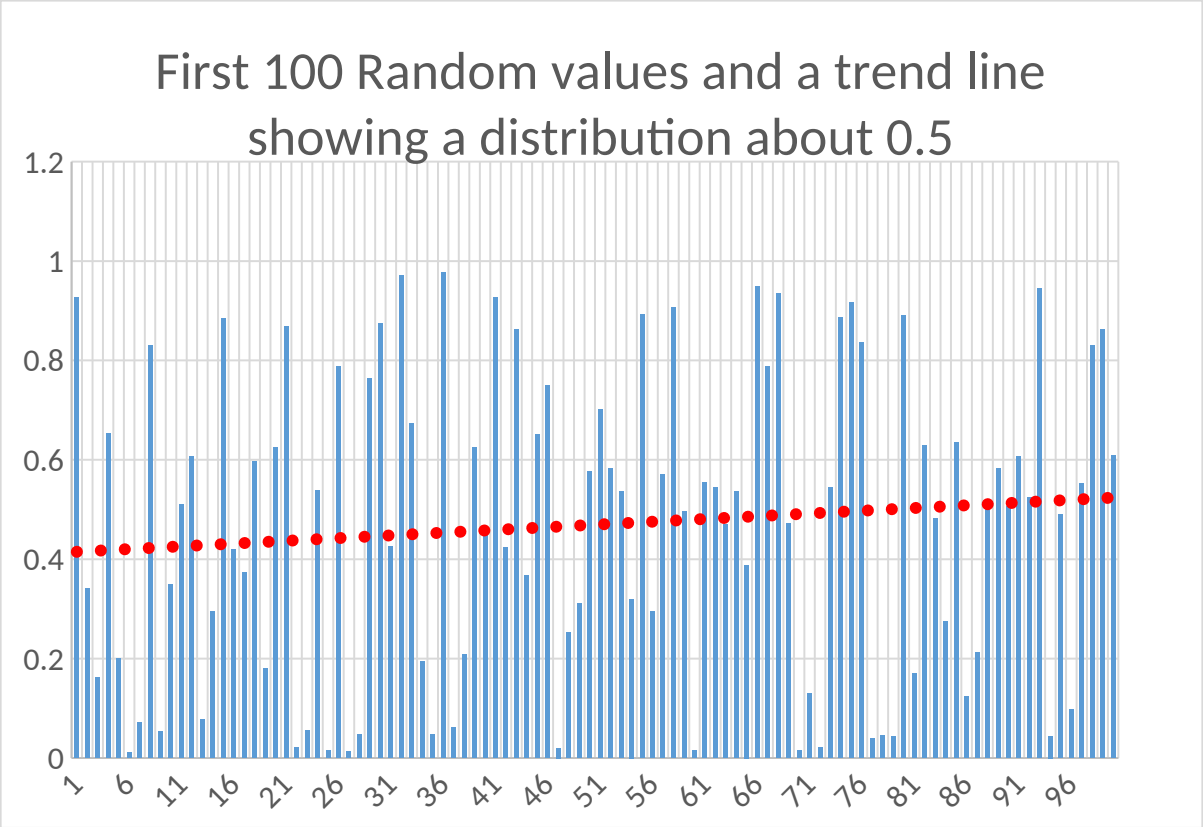


Figure 1 Wichmann-Hill Sequence

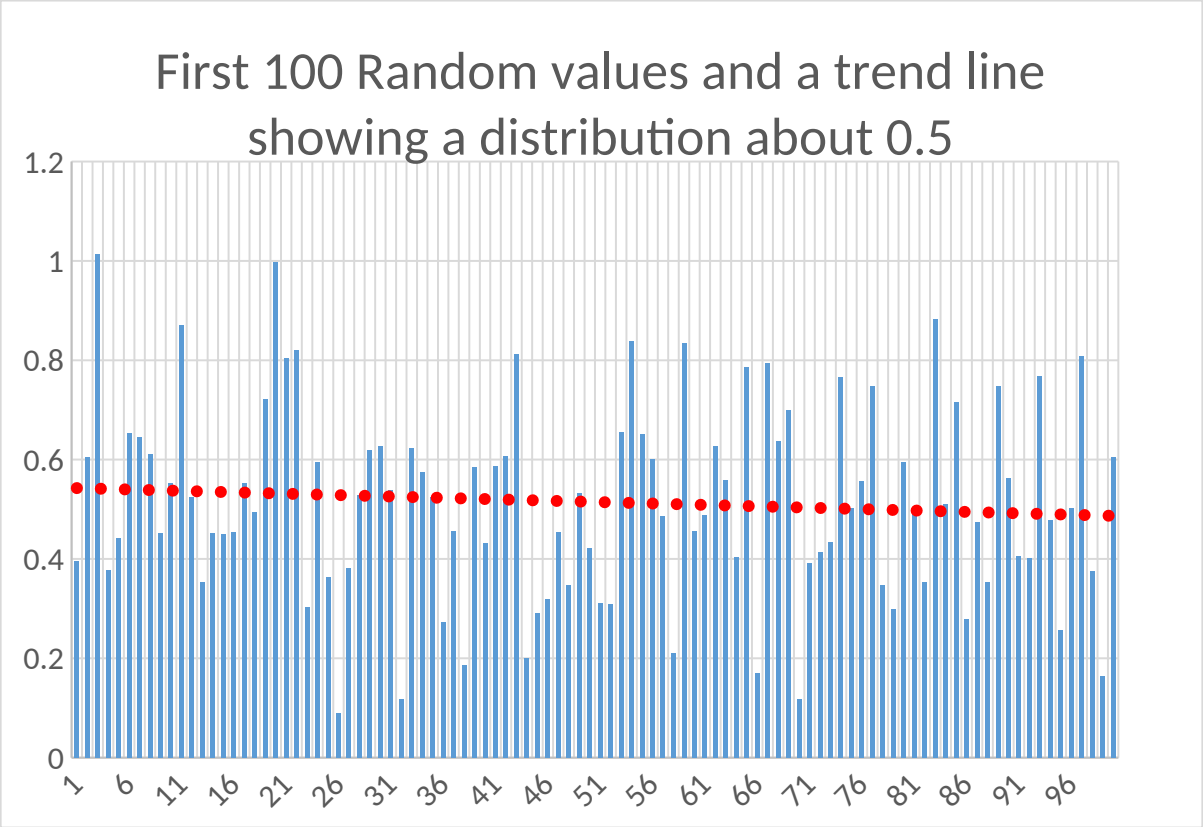


Figure 2 Marsaglia Sequence

Below is shown the distributions of the sequences compared to the distributions produced by the built in functions of Microsoft Excel. It is clearly demonstrated that, although grainy, the sequences are a close resemblance to the expected distribution.

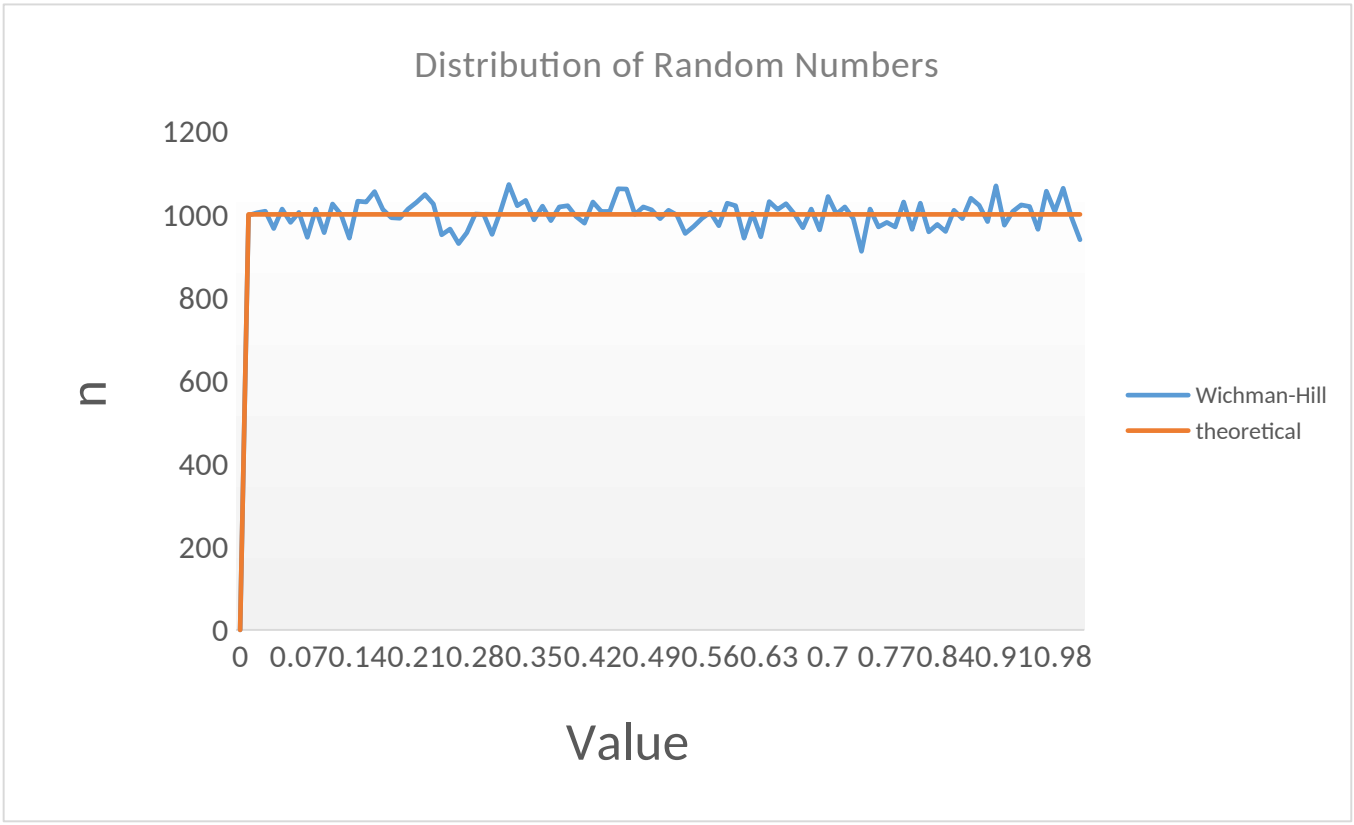


Figure 3 Experimental uniform distribution compared to the theoretical. Experimental mean of 0.50039

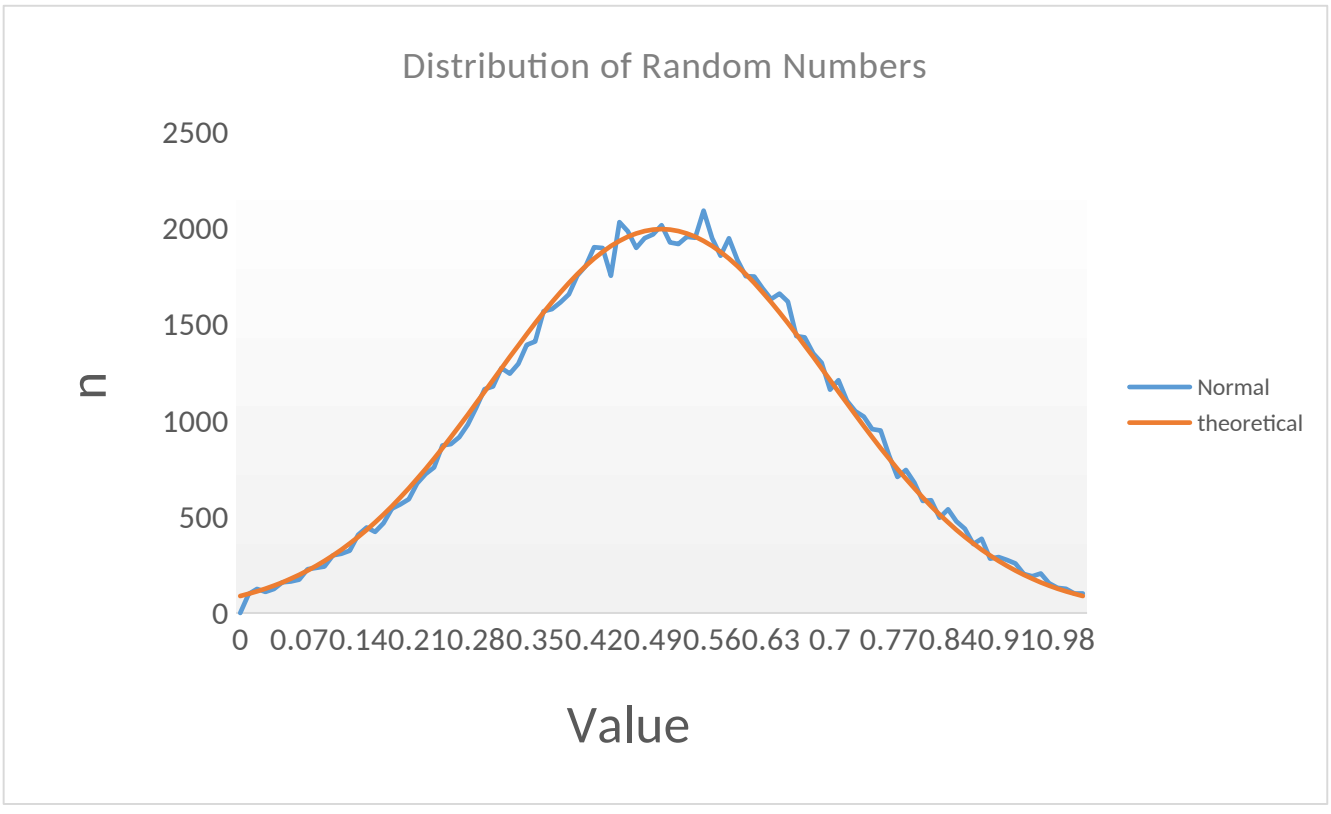


Figure 4 Experimental normal distribution (mean 0.01047, standard deviation 0.040089) compared to Theoretical (Mean 0.5, Standard deviation 0.2)

4. Discussion and Conclusion

It is clear from this study that both methods are indeed very useful for use in the real word. They closely resemble the expected distributions. Further it is clear from these results that the longer the sequence the closer it resembles the statistically expected distribution.

5. References

Lathi B.P & Ding Z. 2010. *Modern Digital and Analogue Communication Systems*, 4th ed. Oxford: Oxford University Press.

Marsaglia, G. and Bray, T. A. “A Convenient Method for Generating Normal Variables” *SIAM Review* Vol. 6, No. 3 (Jul., 1964), pp. 260-264

Wichmann, B; Hill, D "Algorithm AS 183: An Efficient and Portable Pseudo-Random Number Generator" *Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 31, No. 2 (1982), pp. 188-190*.