ISSN 0974 - 9373

Vol. 17 No.2 (2013) **Journal of International Academy of Physical Sciences** pp. 203-222

One Dimensional Numerical Integration by Monte Carlo Method using Random and Equispaced Points

Saurabh Saxena
Department of Applied Mathematics FIET, Bareilly
Email: saurabh_saxena22@rediffmail.com

A. K. Saxena
Department of Mathematics, Bareilly College Bareilly
Email: aksaxena.saxena61@gmail.com

(Received March 12, 2013)

Abstract: Monte Carlo Method has taken extensive applications in many fields using only random numbers generated by different and efficient random number generators. The proposed research work deals with the use of Monte Carlo Method for Numerical Integration. The basic requirements for Monte Carlo method is Sample should be random and Sample size should be large. So far the research work in this field only comprises the efficiency of random number generator and how the randomness of these numbers may be increased to get the best approximation of an integral using these numbers. Here we are proposing the same method for numerical integration but the approach takes a new idea of using the equispaced numbers instead of random numbers i.e. when we apply Monte Carlo method for numerical integration then instead of evaluating the function over the random points in the given range of integration we first divide the range of integration into n equal interval, obtain n equispaced points and then evaluate the integral over these points.

Key Words: Monte Carlo Method, Numerical Integration, Random Numbers, Equispaced Points, Integral Evaluation, GW Basic

1. Introduction

Although it may look simple at first sight to give a definition of what a random number[1] is, it proves to be quite difficult in practice. A random number is a number generated by a process, whose outcome is unpredictable, and which cannot be sub sequentially reliably reproduced. This definition works fine provided that one has some kind of a black box such a black box is usually called a random number generator[2] that fulfills this task. However, if one were to be given a number, it is simply impossible to verify whether it was produced by a random number generator or not[3]. In order to study the randomness of the output of such a generator, it is hence absolutely essential to consider sequences of numbers.

The Monte Carlo method[4] is a method for solving problems using random variables. It is a powerful tool in many fields of mathematics, physics and engineering[5].

On the basis of this method Monte Carlo integration becomes a mathematical technique that relies on statistical properties of random variables and random sampling to estimate integrals numerically. Monte Carlo methods estimate integrals or other quantities that can be expressed as an expectation by averaging the results of a high number of statistical trials.

Source of Random Numbers

The basic requirement of Monte Carlo technique for numerical integration is a large sample of random numbers[6] and just for the sake of the same and to prove our claim unbiased we are taking this large sample from two different sources.

Our first source for random numbers is a program (random number generator) of GWBASIC as the author is well versed with GWBASIC. The program for this very purpose is as following

PROG1.BAS

```
10      CLS : KEY OFF
20      D$ = DATE$: T$ = TIME$
30      V = TIMER: V$ = STR$(V): V1$ = LEFT$(V$,5) : DIM N(5000)
40      LOCATE 6,15 : PRINT "Choice 1 for 1000 numbers file"
50      LOCATE 8,15: PRINT "Choice 2 for 2000 numbers file"
60      LOCATE 10,15: PRINT "Choice 3 for 3000 numbers file"
70      LOCATE 12,15: PRINT "Choice 4 for 4000 numbers file"
80      LOCATE 14,15: PRINT "Choice 5 for 5000 numbers file"
90      LOCATE 18,15:INPUT "Type your Choice Number....( 1 to 5 )";CHOICE
100         IF CHOICE < 0 OR CHOICE > 5 THEN 110 ELSE 130
110         LOCATE 18,15:PRINT"                                        "
120     GOTO 90
130     F$ = "H:Mydata"+ RIGHT$(STR$(CHOICE),1) + ".dat"
140     OPEN F$ FOR OUTPUT AS #1
150     RANDOMIZE(V)
160     FOR I = 1 TO CHOICE*1000
170         R$ = STR$(RND)
180          N(I)= VAL(R$)
```

```
190         'IF VAL(LEFT$(R$,2))= 0 THEN PRINT R$,
200     WRITE #1,N(I):COUNT = COUNT + 1
210     NEXT I
220     CLS:PRINT:PRINT TAB(25): PRINT "Name of Data File ", F$
230     PRINT:PRINT:PRINT TAB(25):PRINT "NUMBER of Data ",,COUNT
240     PRINT:PRINT: PRINT TAB(25)"DATE of creation     ",D$
250     PRINT:PRINT: PRINT TAB(25)"TIME of creation     ",T$
260     CLOSE #1:END
```

Since we are going to discuss three different integral of one dimension therefore by the repeated execution(three times for each file size) of this program we store these numbers as **".dat"** files of size 1000, 2000, 3000, 4000, 5000 with the name

Int_1_1, Int_2_1, Int_3_1, Int_1_2, Int_2_2, Int_3_2,
Int_1_3, Int_2_3, Int_3_3, Int_4_1, Int_4_2,
Int_4_3, Int_5_1, Int_5_2, Int_5_3,

Here the nomenclature of these files is having a particular pattern. The first three letters Int refers to the word integral, the next number refers to the file size and the last numeral refers to the no. of integral (i.e. 1st or 2nd or 3rd)

Our second source of random numbers is online generators of random numbers. For this purpose we searched many sites which provide random numbers. Out of them three sides which are taken under consideration are

RANDOM.ORG **(http://www.random.org/decimal-fractions/)**

RANDOM.ORG offers true random numbers to anyone on the Internet. The randomness comes from atmospheric noise, which for many purposes is better than the pseudo-random number algorithms typically used in computer programs. People use RANDOM.ORG for holding drawings, lotteries and sweepstakes, to drive games and gambling sites, for scientific applications and for art and music. The service has existed since 1998 and was built and is being operated by Mads Haahr of the School of Computer Science and Statistics at Trinity College, Dublin in Ireland.

As of today, RANDOM.ORG has generated 1.12 trillion random bits for the Internet community. The numbers used in our work generated by random.org are obtained as fractional values up to four decimal places between 0 and 1 and used directly in our work.

RESEARCH RANDOMIZER (http://www.randomizer.org/form.htm)

RESEARCH RANDOMIZER is the site which is designed for researchers and students who want a quick way to generate random numbers or assign participants to experimental conditions. Research Randomizer can be used in a wide variety of situations, including psychology experiments, medical trials, and survey research. The program uses a JavaScript random number generator to produce customized sets of random numbers.

Since its release in 1997, Research Randomizer has been used to generate number sets over 15.8 million times. This service is part of Social Psychology Network and is fast, free, and runs with any recent web browser as long as JavaScript isn't disabled.

The numbers used in our work generated by randomizer.org are obtained as integral values of four digits between 0 and 9999 and then divided by 10000 to obtain fractional values between 0 and 1 and then they are used in our work.

GRAPH PAD Software (http://www.graphpad.com/quickcalcs/randomn1.cfm)

Graph-Pad Software has been dedicated to creating software exclusively for the international scientific community since 1984. Created by scientists for scientists, its intuitive programs provide researchers worldwide with the tools they need to simplify data analysis, statistics and graphing.

It provides free service as quick calcs which is an online calculator for scientist and researchers to generate random numbers.

The numbers used in our work generated by graphpad.com are obtained as integral values of four digits between 0 and 9999 and then divided by 10000 to obtain fractional values between 0 and 1 then they are used in our work.

Five data files of random numbers from each of the above noted sites are saved as under.

File Name	Site	Size
olrr1.dat	Research Randomizer	1000
olrr1.dat	Research Randomizer	2000
olrr1.dat	Research Randomizer	3000
olrr1.dat	Research Randomizer	4000
olrr1.dat	Research Randomizer	5000
olrorg1.dat	Random.Org	1000
olrorg2.dat	Random.Org	2000
olrorg3.dat	Random.Org	3000
olrorg4.dat	Random.Org	4000
olrorg5.dat	Random.Org	5000
olgp1.dat	Graph Pad	1000
olgp2.dat	Graph Pad	2000
olgp3.dat	Graph Pad	3000
olgp4.dat	Graph Pad	4000
olgp5.dat	Graph Pad	5000

Table 1

As far as the notation and nomenclature of these files are concerned it should be noted

ol	stands	for	ONLINE,
gp	stands	for	GRAPH PAD
rorg	stands	for	RANDOM.ORG
rr	stands	for	RESEARCH RANDOMIZER

and the last numeral n stands for the size of the file which is n multiplied by 1000.

Since it is very cumbersome to present all these numbers in this research paper therefore these may be seen and accessed from the following web address :

http://www.4shared.com/folder/BAytR7eW/data_files.html

All the random numbers in the above noted files are distinct and have no correlation with each other. Before using these numbers in the research paper these numbers have gone through four methods to test[7] their independence and these methods are Poker Test[8], Run Test[9, 10], Frequency Test[11] and Frequency Monobit Test[12].

2. Monte Carlo Method for Numerical Integration

Just to embrace a wide range of problem solving techniques which use random numbers and statistics of probability, the term "Monte Carlo Method[13]" is used. The term Monte Carlo is being coined after the casino in the principality of Monte Carlo. Factually speaking every game in a casino is a game of chance relying on random events. That is why any method that uses random numbers to examine some problem is a Monte Carlo Method. The term Monte Carlo Method[14,15] was first used by Stanslaw Ulam for simulations in Physics and other fields that require solutions for problems that are impractical or impossible to solve by traditional analytical or numerical methods. It is indeed an artificial sampling method which can be used for solving complicated problems in analytic formulation and for simulating purely statistical problems. The method is being used more and more in recent years, especially in those cases where the number of factors included in the problem is so large that an analytical solution is impossible.

The central idea behind the Monte Carlo method is either to construct a stochastic model which is in agreement with the actual problem analytically or to simulate the whole problem directly[16]. In both the cases the element of randomness has to be introduced according to well defined rules. After that a large number of trials are performed and the results are observed and finally a statistical analysis is undertaken in the usual way. The advantages of the method are, above everything is that even very difficult problems can often be treated quite easily and desired modifications can be applied without too much trouble. The poor precision is the main disadvantage of the method and as such large number of trials is necessary. This drawback is of little importance as the calculations are almost exclusively performed on automatic computers.

As far as the "Hit or Miss[17]" method is concerned, we suppose that the function f(x) is bounded by

$$0 \le f(x) \le c \text{ for } a \le x \le b$$

and we wish to evaluate $\mathrm{I} = \int_a^b f(x)\,dx$

For it, we consider the following two regions

$$S = \{(x, y) : a \le x \le b; 0 \le y \le f(x)\}$$

and $\Omega = \{(x, y) : a \le x \le b; 0 \le y \le c\}$

i.e. S is the region below the curve of the function $f(x)$ And Ω is a rectangle that covers S completely.

If |S| stands for the area S then I = |S|

The area Ω is easy to find, which is simply

$$c(b-a)\ \Omega = \{(x, y) : a \le x \le b; 0 \le y \le c\}\ .$$

If by applying any method we are in position to find the proportion $\frac{|S|}{|\Omega|}$.

Then $\mathrm{I} = |\Omega| \times \frac{|S|}{|\Omega|}$

The proportion $\frac{|S|}{|\Omega|}$ can be estimated using simulation by generating random points uniformly distributed in the region Ω and then counting how many of them are falling in S. If n be the total number of random points generated and n_s be the number of points in S then the integral I can be estimated as

$$\mathrm{I} = c(b-a)\frac{n_s}{n}.$$

We can also view the problem of evaluating the integral

$$\mathrm{I} = \int_a^b f(x)\,dx$$
,

where $f(x)$ is supposed to be a continuous and real valued function on $[a,b)$ by defining a second function $g(x)$ as below

$$g(x) = \begin{Bmatrix} \dfrac{1}{(b-a)} when a \le x \le b \\ 0 otherwise \end{Bmatrix}$$

On account of this definition we follow $\int_{-\infty}^{+\infty} g(x)\,dx = 1.$

Showing that we can consider $g(x)$ to be a probability density function. If we insert this function $g(x)$ in the constitution of I, we agree to say

$$\mathrm{I} = (b-a)\int_a^b f(x)g(x)dx = (b-a)\,\mathrm{E}\{f(x)\}.$$

Thus the integral I is simply the width of the integral multiplied by the expectation of the integrand. Now the problem of evaluating I is now in a form that can be solved efficiently using Monte Carlo method. The additional advantage of this statistical interpretation is that we now have a set of algebraic tools for determining the accuracy of the integration because we can calculate the variance of Monte Carlo approximation.

Now the very first step of Monte Carlo Problem is to set up a simulation to approximate the value of $\mathrm{E}\{f(x)\}$

This expectation can be evaluated by taking the mean of the functional value at n randomly selected points emanating from a sequence x_n of pseudo random numbers generated by some PRN generator[18] which provides values in the interval $a \le x \le b$

Hence I can be approximated by

$$\mathrm{I} = (b-a)\int_a^b f(x)\,g(x)\,dx \;= (b-a)\mathrm{E}\{f(x)\} \approx \frac{(b-a)}{n}\sum_i^n f(x_i).$$

On account of the fact that (b-a) is eaxact, the uncertainty of the algorithm is bounded by the uncertainty in our estimate of E{f(x)}.If our estimate for E{f(x)} is precise then the sum of the area between f(x) and E{f(x)} should be exactly equal to zero. The error term can be approximated by the root mean square deviation σ of the expectation which is given by

$$\sigma = \sqrt{\frac{f(x)^2 - \mathrm{E}^2\{f(x)\}}{n}}$$

Where $\langle f(x) \rangle$ is the sequence of the functional values and $\mathrm{E}^2\{f(x)\}$ is the sequence with same size having elements equal to $\mathrm{E}^2\{f(x)\}$.

3. Integral Evaluation (Using Random and Equispaced Nodes)

In the present work, following three types of one dimensional integrals[19] are taken into consideration and are evaluated by Monte Carlo method using RANDOM POINTS (online generated) as well as EQUISPACED POINTS.

i. The first integral under investigation is

$$\mathrm{I}_1 = \int_0^1 \frac{4}{1+x^2}\,dx.$$

Exact value of which is 3.142857

ii. The second integral under investigation is

$$\mathrm{I}_2 = \int_0^1 \sqrt{\mathrm{x} + \sqrt{\mathrm{x}}}\,\mathrm{dx}.$$

Exact value of which is 1.045394

iii. The third and last integral is

$$\mathrm{I}_3 = \int_0^1 \left(\sin\sin \mathrm{x} - \log\mathrm{x} + \mathrm{e}^{\mathrm{x}}\right)\mathrm{dx}.$$

Exact value of which is 3.177979

First Integral (1-D)

In order to evaluate our first (1-D) integral

$$\mathrm{I}_1 = \int_0^1 \frac{4}{1+x^2}\,dx.$$

By Monte Carlo Integration[19,20,21] both by random nodes and equispaced nodes, we now develop a computer program in GWBASIC which can be used for the other two 1-D integrals under consideration but with modifications in those lines of the program which are responsible for the construction of the integrand. The program which is especially for our first integral under investigation is

PROG2.BAS

```
10  REM****SUBROUTINE FOR FILE SELECTION*****
20      CLS:KEY OFF:DIM N$(5)
25      LOCATE 5,5 : INPUT "Give the drive letter of data files
residence";Z$
26      CLS
30      LOCATE 4,5: PRINT "Following files are available...."
40
    LOCATE7,5:PRINT"1......INT_1_1.DAT","2......INT_2_1.DAT","3......I
NT_3_1.DAT"
50      LOCATE 8,5:PRINT "4......olrr1.DAT","5......olrorg1.DAT",
"6......olgp1.DAT"
60      LOCATE 10,5:PRINT"7......INT_1_2.DAT","8......INT_2_2.DAT",
"9......INT_3_2.DAT"
70      LOCATE 11,5:PRINT"10.....olrr2.DAT","11.....olrorg2.DAT",
"12.....olgp2.DAT"
80      LOCATE 13,5:PRINT"13.....INT_1_3.DAT","14.....INT_2_3.DAT",
"15.....INT_3_3.DAT"
90      LOCATE 14,5:PRINT"16.....olrr3.DAT","17.....olrorg3.DAT",
"18.....olgp3.DAT"
100     LOCATE 16,5:PRINT"19.....INT_1_4.DAT","20.....INT_2_4.DAT",
"21.....INT_3_4.DAT"
110     LOCATE 17,5:PRINT"22.....olrr4.DAT","23.....olrorg4.DAT",
"24.....olgp4.DAT"
120     LOCATE 19,5:PRINT"25.....INT_1_5.DAT","26.....INT_2_5.DAT",
"27.....INT_3_5.DAT"
130     LOCATE 20,5:PRINT"28.....olrr5.DAT","29.....olrorg5.DAT",
"30.....olgp5.DAT"
140     LOCATE 22,5 : INPUT "Give File Number ( 1 - 30)";ANS
150     IF ANS < 0 OR ANS > 30 THEN 160 ELSE 170
160     LOCATE 22,5: PRINT "                              " : GOTO  140
170     IF ANS < 7 THEN DM=1000
```

```
180     IF ANS > 6 AND ANS < 13  THEN DM=2000
190     IF ANS > 12 AND ANS < 19 THEN DM=3000
200     IF ANS > 18 AND ANS < 25 THEN DM=4000
210     IF ANS > 24 AND ANS < 31 THEN DM=5000
220     IF ANS=1 OR ANS =7 OR ANS=13 OR ANS=19 OR  ANS=25
THEN F$=Z$+":INT_1_"
230     IF ANS=2 OR ANS =8 OR ANS=14 OR ANS=20 OR ANS=26
   THEN F$=Z$+":INT_2_"
240     IF ANS=3 OR ANS =9 OR ANS=15 OR ANS=21 OR ANS=27
   THEN F$=Z$+":INT_3_"
250     IF ANS=4 OR ANS =10 OR ANS=16 OR ANS=22 OR ANS=28
        THEN F$=Z$+":olrr"
260     IF ANS=5 OR ANS =11 OR ANS=17 OR ANS=23 OR ANS=29
        THEN F$=Z$+":olrorg"
270     IF ANS=6 OR ANS =12 OR ANS=18 OR ANS=24 OR ANS=30
        THEN F$=Z$+":olgp"
280     IF DM=1000 THEN F$=F$+"1.dat"
290     IF DM=2000 THEN F$=F$+"2.dat"
300     IF DM=3000 THEN F$=F$+"3.dat"
310     IF DM=4000 THEN F$=F$+"4.dat"
320     IF DM=5000 THEN F$=F$+"5.dat"
330     DIM N(DM):OPEN F$ FOR INPUT  AS #1
340      INPUT #1,X: N(COUNT+1)=X:COUNT=COUNT+1
350      IF NOT EOF(1) THEN 340
360     FOR I = 1 TO DM
370     SUM=SUM+ 4/(1+N(I)*N(I))
380      NEXT I
390     SUM = SUM/DM:CLS:LOCATE 5,10:PRINT "Value of Integral"
400     LOCATE 8,10: PRINT "Using ";DM;" Random Nodes"
410     LOCATE 10,10: PRINT "from ";F$" =",SUM
420     LO=0:UP=1:H=(UP-LO)/DM:ESUM=0
430      FOR I = 0 TO DM
440      ESUM=ESUM+ 4/(1+LO*LO):LO=LO+H
450      NEXT I
460     ESUM=ESUM/DM
490     LOCATE 16,10: PRINT "Value of Integral"
500     LOCATE 18,10:PRINT "Using ";DM;" Equispaced nodes =
",ESUM
510     CLOSE:END
```

For the self generated data files INT_1_1.DAT; INT_1_2.DAT; INT_1_3.DAT; INT_1_4.DAT; INT_1_5.DAT With their file codes as (1, 7, 13, 19, 25) the value of the integral is given in **table 2**

Integral-1

SELF GENERATED DATA FILES (CODES : 1,7,13,19,25)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.162375	3.1406	3.142879
2	2000	3.158796	3.141085	3.142879
3	3000	3.145404	3.141271	3.142879
4	4000	3.146828	3.141351	3.142879
5	5000	3.134432	3.14142	3.142879

Table 2

For the Online generated data files Olgp1.DAT; Olgp2.DAT; Olgp3.DAT; Olgp4.DAT; Olgp5.DAT With their file codes as (6, 12, 18, 24, 30) the value of the integral is given in **table 3**

Integral-1

ONLINE GENERATED DATA FILES (CODES : 6,12,18,24,30)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.112968	3.1406	3.142879
2	2000	3.129597	3.141085	3.142879
3	3000	3.134179	3.141271	3.142879
4	4000	3.135012	3.141351	3.142879
5	5000	3.144092	3.14142	3.142879

Table 3

For the Online generated data files Olrorg1.DAT; Olrorg2.DAT;Olrorg3.DAT;Olrorg4.DAT;Olrorg5.DAT With their file codes as (5, 11, 17, 23, 29) the value of the integral is given in **table 4**

Integral-1

ONLINE GENERATED DATA FILES (CODES : 5,11,17,23,29)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.1263	3.1406	3.142879
2	2000	3.138961	3.141085	3.142879
3	3000	3.131677	3.141271	3.142879
4	4000	3.145027	3.141351	3.142879
5	5000	3.134969	3.14142	3.142879

Table 4

For the Online generated data files Olrr1.DAT; Olrr2.DAT; Olrr3.DAT; Olrr4.DAT; Olrr5.DAT With their file codes as (4, 10, 16, 22, 28) the value of the integral is given in **table 5**

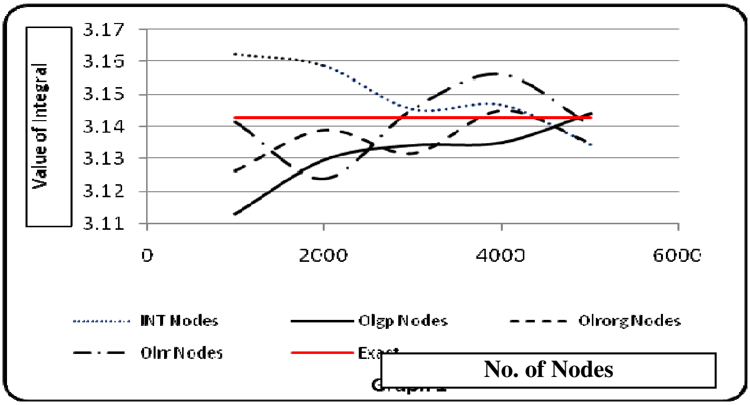
Integral-1

ONLINE GENERATED DATA FILES (CODES : 4,10,16,22,28)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.141363	3.1406	3.142879
2	2000	3.123959	3.141085	3.142879
3	3000	3.145404	3.141271	3.142879
4	4000	3.15603	3.141351	3.142879
5	5000	3.140446	3.14142	3.142879

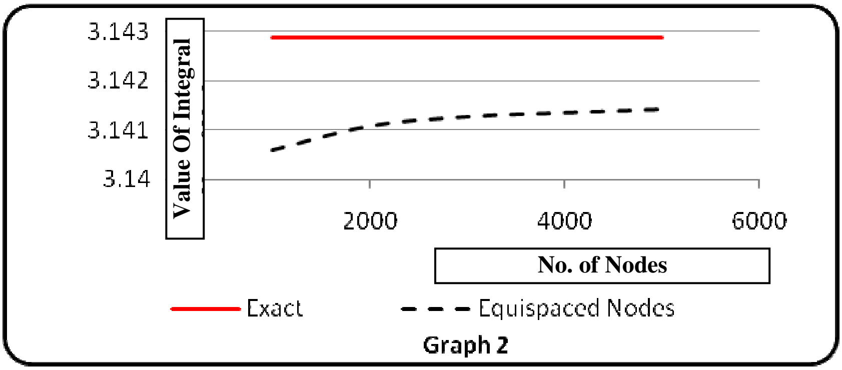
Table 5

The graphical display of values of our first integral corresponding to the random nodes from table 2, 3, 4, 5 is shown in **graph 1**



Graph 1

while the display corresponding to equidistant nodes is shown in **graph2**



Graph 2

Second Integral (1-D)

For the evaluation of our second integral

$$I_2 = \int_0^1 \sqrt{x + \sqrt{x}}\, dx.$$

By Monte Carlo Integration both by random nodes and equispaced nodes, we shall use the same computer program PROG2.BAS in GWBASIC but with a modification in lines as

```
        370     SUM = SUM + SQR( N(I)+SQR(N(I)))
and     440     ESUM = ESUM+SQR( LO+SQR(LO)) : LO=LO+H
```

For the self generated data files INT_2_1.DAT; INT_2_2.DAT;INT_2_3.DAT;INT_2_4.DAT; INT_2_5.DAT With their file codes as (2, 8, 14, 20, 26) the value of the integral is given in **table6**

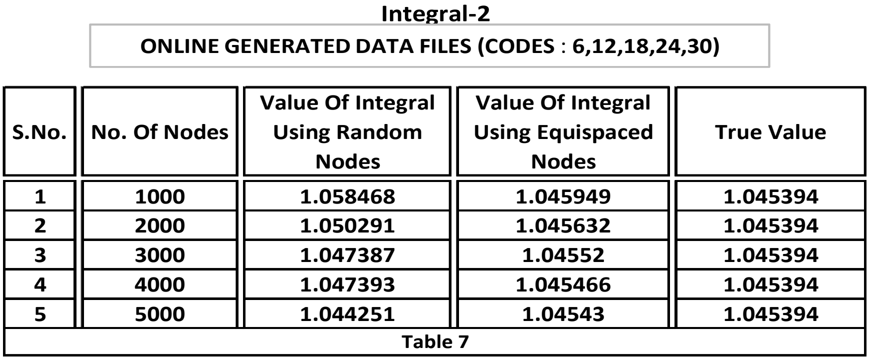
Integral-2

SELF GENERATED DATA FILES (CODES : 2,8,14,20,26)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	1.056092	1.045949	1.045394
2	2000	1.055633	1.045632	1.045394
3	3000	1.051789	1.04552	1.045394
4	4000	1.043059	1.045466	1.045394
5	5000	1.048893	1.04543	1.045394

Table 6

For the Online generated data files Olgp1.DAT; Olgp2.DAT; Olgp3.DAT; Olgp4.DAT; Olgp5.DAT With their file codes as (6,12,18,24,30) the value of the integral is given in **table 7**



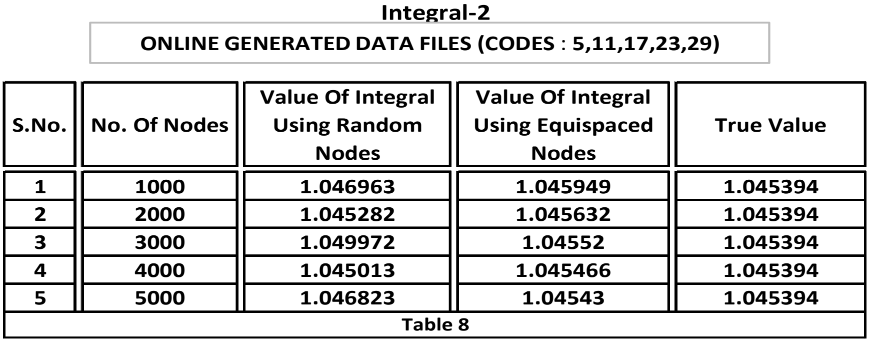
Integral-2

ONLINE GENERATED DATA FILES (CODES : 6,12,18,24,30)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	1.058468	1.045949	1.045394
2	2000	1.050291	1.045632	1.045394
3	3000	1.047387	1.04552	1.045394
4	4000	1.047393	1.045466	1.045394
5	5000	1.044251	1.04543	1.045394

Table 7

For the Online generated data files Olrorg1.DAT; Olrorg2.DAT; Olrorg3.DAT; Olrorg4.DAT; Olrorg5.DAT with their file codes as (5, 11, 17, 23, 29) the value of the integral is given in **table 8**



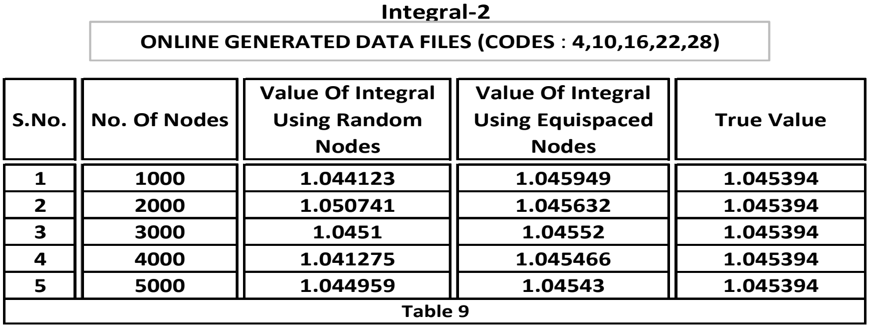
Integral-2

ONLINE GENERATED DATA FILES (CODES : 5,11,17,23,29)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	1.046963	1.045949	1.045394
2	2000	1.045282	1.045632	1.045394
3	3000	1.049972	1.04552	1.045394
4	4000	1.045013	1.045466	1.045394
5	5000	1.046823	1.04543	1.045394

Table 8

For the Online generated data files Olrr1.DAT; Olrr2.DAT; Olrr3.DAT; Olrr4.DAT; Olrr5.DAT with their file codes as (4, 10, 16, 22, 28) the value of the integral is given in **table 9**



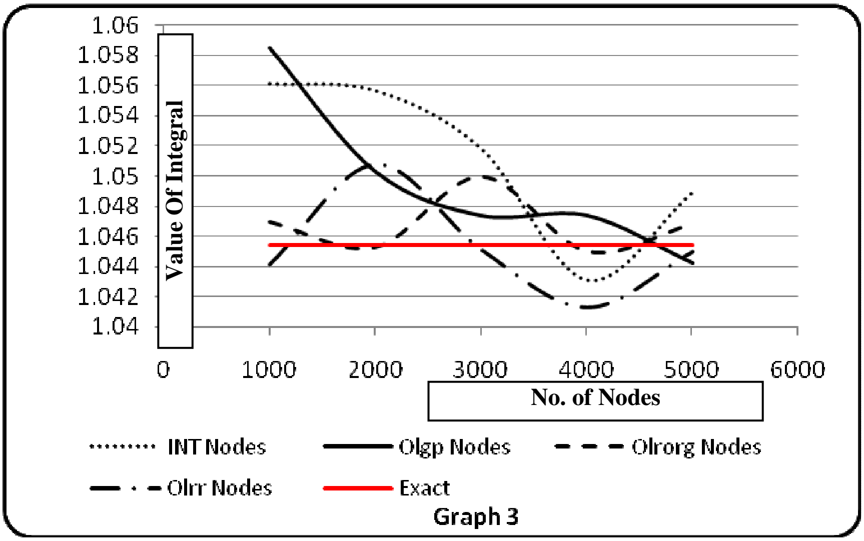
Integral-2

ONLINE GENERATED DATA FILES (CODES : 4,10,16,22,28)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	1.044123	1.045949	1.045394
2	2000	1.050741	1.045632	1.045394
3	3000	1.0451	1.04552	1.045394
4	4000	1.041275	1.045466	1.045394
5	5000	1.044959	1.04543	1.045394

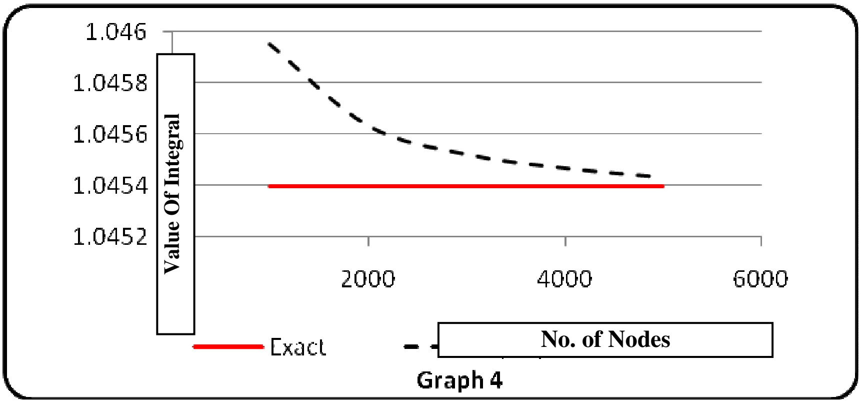
Table 9

The graphical display of values of our second integral corresponding to the random nodes is from table 6, 7, 8, 9 as shown in the **graph 3**



Graph 3

While the display corresponding to equidistant nodes is as shown in **Graph4** .



Graph 4

Third Integral (1-D)

The third and the last one dimensional integral under investigation is

$$I_3 = \int_0^1 \left(\sin \sin x - \log x + e^x\right) dx.$$

For Monte Carlo Integration both by random nodes and equispaced nodes, we shall use the same computer program PROG2.BAS in GWBASIC but with a modification in lines as

```
        370     SUM = SUM + 4/(1+N(I)*N(I))
        420     LO=0:UP=1:H=(UP-LO)/DM:ESUM=0
        430     FOR I = 0 TO DM
and     440     ESUM=ESUM+ 4/(1+LO*LO): LO=LO+H
```

For the self generated data files INT_3_1.DAT; INT_3_2.DAT;INT_3_3.DAT; INT_3_4.DAT;INT_3_5.DAT With their file codes as (3, 9, 15, 21, 27) the value of the integral is given in **table 10**

Integral-3

SELF GENERATED DATA FILES (CODES : 3,9,15,21,27)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.185562	3.174883	3.177979
2	2000	3.166672	3.176267	3.177979
3	3000	3.176696	3.176754	3.177979
4	4000	3.17425	3.177026	3.177979
5	5000	3.165725	3.177175	3.177979

Table 10

For the Online generated data files Olgp1.DAT; Olgp2.DAT; Olgp3.DAT; Olgp4.DAT; Olgp5.DAT with their file codes as (6,12,18,24,30) the value of the integral is given in **table 11**

Integral-3

ONLINE GENERATED DATA FILES (CODES : 6,12,18,24,30)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.166378	3.174883	3.177979
2	2000	3.17613	3.176267	3.177979
3	3000	3.180521	3.176754	3.177979
4	4000	3.182356	3.177026	3.177979
5	5000	3.178202	3.177175	3.177979

Table 11

For the Online generated data files Olrorg1.DAT; Olrorg2.DAT; Olrorg3.DAT; Olrorg4.DAT; Olrorg5.DAT with their file codes as (5, 11, 17, 23, 29) the value of the integral is given in **table 12**

Integral-3

ONLINE GENERATED DATA FILES (CODES : 5,11,17,23,29)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.195598	3.174883	3.177979
2	2000	3.181391	3.176267	3.177979
3	3000	3.172979	3.176754	3.177979
4	4000	3.172535	3.177026	3.177979
5	5000	3.178113	3.177175	3.177979

Table 12

For the Online generated data files Olrr1.DAT; Olrr2.DAT; Olrr3.DAT; Olrr4.DAT; Olrr5.DAT with their file codes as (4, 10, 16, 22, 28) the value of the integral is given in **table 13**

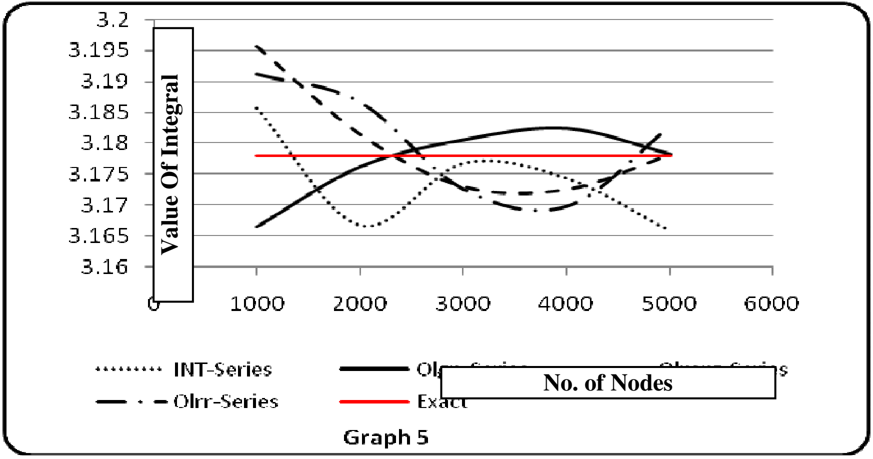
Integral-3

ONLINE GENERATED DATA FILES (CODES : 4,10,16,22,28)

S.No.	No. Of Nodes	Value Of Integral Using Random Nodes	Value Of Integral Using Equispaced Nodes	True Value
1	1000	3.191116	3.174883	3.177979
2	2000	3.186556	3.176267	3.177979
3	3000	3.172532	3.176754	3.177979
4	4000	3.169787	3.177026	3.177979
5	5000	3.182721	3.177175	3.177979

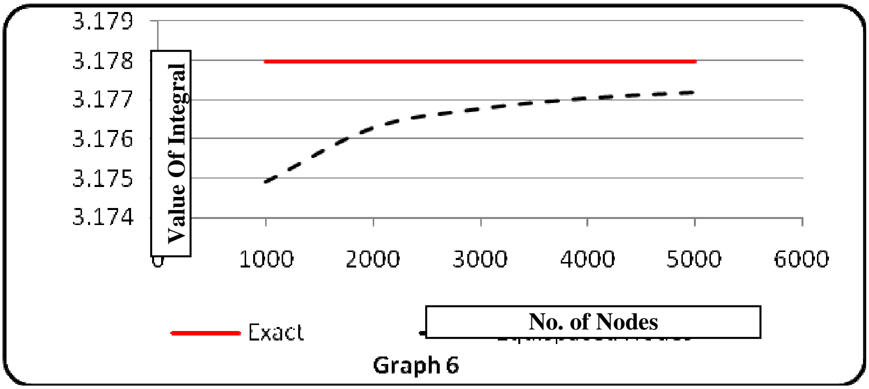
Table 13

The graphical display of values of our third integral corresponding to the random nodes from table 10, 11, 12, 13 is shown in **graph 5**.



Graph 5

while the display corresponding to equidistant nodes is as shown in **graph 6**.



Graph 6

4. Conclusion

Here we observe that Value of all the single integrals doesn't follow any pattern corresponding to different size of random numbers[17,13,2] and also seems to be random in nature whereas using equispaced nodes the value of integral follow a pattern and steadily approaches to the exact value of integral [see graph[1,14,19]] and true value of integral is almost achieved using 5000 equispaced points.

Hence we can conclude the following points

- However random the numbers (True or Pseudo) are used in Monte Carlo integration it is not necessary that the set of random numbers which gives the best approximation of one integral (single or multiple) will also yield the same accuracy in the evaluation of other integral whereas if we use equispaced numbers in a given range of integration then we get almost smooth curve corresponding to the values of integral having a regular decrement in the error of the value of integral.

- It is also not necessary that corresponding to different number of random numbers obtained through same source will give the value of integral more and more approximated i.e. increase in the random numbers doesn't give the assurance for regular decrement in error whereas using equispaced nodes we get a regular decrement in the error of value of integral.

References

1. Gregory J. Chaitin,, Randomness and Mathematical Proof, *Scientific American* **232 (5)** (1975) 47-52.

2. D. Eddelbuettel, *Random: True random numbers using random.org,* 2007, http://www.random.org.

3. Peter Hellekalek, Good random number generators are (not so) easy to find, *Mathematics and Computers in Simulation*, 46 (1998) 485-505.

4. Malvin H. Kalos and Paula A. Whitlock, *Monte Carlo Methods*. Wiley-VCH. ISBN 978-3-527-40760-6, 2008.

5. Mrinal Mishra and Nisha Gupta, Monte carlo Integration Tchnique for method of moment solution of electric field integral equation in scattering problems, *J. Electromagnetic Analysis & applications*, **1** (2009) 254-258.

6. J. E. Gentle, *Random Number Generation and Monte Carlo Methods,* Springer, New York, second edition, 2003.

7. Bhar Lalmohan, *Non-parametric Test*, Indian Agricultural Statistics Research Institute, LIbrary Avenue, New Delhi.

8. Poker Test. www.cse.msu.edu/~cse808/note/lecture6.pp

9. Run test. www.cse.msu.edu/~cse808/note/lecture6.ppt

10. Run test: test statistic. http://www.iasri.res.in/ebook/EB_SMAR/e-book_pdf%20files/Manual%20II/4-Non-Parametric_test.pdf

11. Uniformity test. http://www.lix.polytechnique.fr/~catuscia/teaching/cg597/01Fall/lecture_notes/pseudo_random.ppt

12. Frequency Monobit Test. csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf.

13. Computational Science Education Project., *"Introduction to Monte Carlo Methods"*, Oak Ridge, Tennessee: Oak Ridge National Laboratory. http://csep1.phy.ornl.gov/mc/mc.html.

14. Roger Eckhardt, Stan Ulam, John von Neumann, and the Monte Carlo method, *Los Alamos Science*, Special Issue **(15)** (1995) 131-137.

15. N. Metropolis, The beginning of the Monte Carlo method, *Los Alamos Science*, **Special Issue** (1987) 125-130. http://iibrary.lanl.gov/la-pubs/00326866.pdf.

16. Monte Carlo module. http://math.fullerton.edu/mathews/n2003montecarlomod.html

17. R. E. Caflisch,. Monte Carlo and quasi-Monte Carlo methods, *Acta Numerica*. **7**. (1998)1–49.

18. S. K. Park and K. W. Miller, Random number generators: Good ones are hard to find, Communications of the ACM, **31**(1988)1192-1201.

19. G. Evans, *Practical Numerical Integration,* Wiley, New York, 1993.

20. K. Gordon, *Numerical Integration*, John Wiley & Sons, Ltd, *(ISBN 0471 975761)* Edited by, Peter Armitage and Theodore Colton, Chichester, 1998.

21. Gould, Harvey, Jan Tobochnik, and Wolfgang Christian, *Numerical Integration and Monte Carlo Methods,* 2001.

22. Yang Guan, *A Monte Carlo Method of Integration,* Extended Essay in Mathematics, Denmark, 2002.

23. Random number generator. www.cse.msu.edu/~cse808/note/lecture6.pp