Random numbers

Technology, Information Technology

Generating Random Numbers Number) February 10, (Faculty) Generating Random Numbers Random numbers can be termed as a group of numbers showing absolutely no relationship to each other in a sequence of occurrence. The fashion in which the numbers occur should be unpredictable and allowing each integer a chance to occur. Random numbers are used in random statistics such as random sampling as well as random treatments when it comes to treating experimental units. Most commonly, the idea of random numbers is applicable in simulation studies. Various phenomena in physics are random in nature and many of the scientists apply algorithms in solving most of their scientific problems.

While practicing the use of random numbers it is highly advisable to choose few random number generators that have an acceptable reputation. Indeed, deriving an excellent random number is still a plight since computation of a series of random numbers cannot be generated by computers. True random numbers generation has turned to be one of the important factors in the increasing levels of innovation. The increase in innovation has made the developing of algorithms that generate random numbers that entail all properties of being true random numbers (Haule, 2014)

Malcolm (2010) argues that all output produced by the machines including computers today are totally predictable hence making it impossible for them to generate true random numbers. In today’s world there are various methods of generating random numbers being used. Various methods have been innovated to produce true random numbers and they include use of radioactive decay or quantum system, Lehmer’s multiplicative congruently algorithm from which lots of random number generators are based, pseudo-

random generators which are simple and when used together with Monte Carlo or multidimensional integration give a much satisfying result. There are several others like the numerical inversion method which is mostly used to generate Gaussian distributed random numbers. From a real experience of random number generating machine that is online, the resultant random number is quite unpredictable from the start. A continuous use of the random number generator revealed a slight clue of predictability as the results seemed to be shifting towards the larger integer out of the two keyed in. This could probably be due to the narrow integral limit of the numerical algorithm.

Pseudo-random generator being one of the easiest random number generators to use and being readily available in various libraries is an excellent generator and is used by several institutions in the globe. They are fast though not of highest quality but it simplicity grants it distinctness from others. It is given by a simple algorithm:

Such that ‘ a’ and ‘ b’ are all constants.

To generate a list of random numbers between 0 and m the values are substituted and a series of integers is displayed. Increasing the value of m increases the total number of random numbers generated. The series is continuously repeated as long as the value of.

Pseudo-random numbers in C-programming is applied by use of command like srand48 and drand48 which are both stated in cstdlib. Example we have commands like random (32767) which displays random numbers that lie between 0 and 32767. To obtain random numbers between 0 and 1 the expression below can be used.

r= random(32767)/32767. 0;

In addition to make a program where the user picks a random number as seed where the default gcc no is ; the codes used above is:

```
#include stdio. h
int main()
{
int r;
double m;
m= pow(2, 31)-1. 0;
r= random()/m;
return 0;
}
```

From the above examples, generating random numbers between c and d is given by:

```
#include stdio. h
int main()
{
int r, c, d;
double m;
x= c+r*(c-d);
}
```

Such that x will result to a random uniform probability between’ a’ and ‘ b’.

r= random ()/m;

Conclusively, randomness of a list of numbers is in the eye of a beholder for what may see to be random may not as random to another person. Random

generators which are branded as good are ought to pass a list of sensible statistical tests. (Flannery et al, 2007) Any show of failure of the random generator to actually generate true random numbers should be marked as poor and the programmer should seek another generator. Pseudo-random generator lies among the easy and efficient generators of random numbers base on the test conducted such as the random walker and spreading of points on a square. The algorithm examples above depict how fast and adaptable generating a random number using pseudo-random generator is.

References

Forsythe, G., Malcoln, M., & Moler, C. (2010). Computer Method for Mathematical Computations. Englewood Cliffs, NJ: Prentice-Hall.

Haule, K. (2014). Random Numbers High Dimensional Integrals. New York, NY: Cambridge University Press.

Teukolsky, S., Vatterling, T., & Flannery, P. (2007). Numerical Recipes 3rd Edition: The Art of Scientific Computing (3rd ed.). New York, NY: Cambridge University Press.