Test of a random number generator using random walks

Nagendra K. Panduranga

April 20, 2018

1 Introduction

The quality of a random number generator can be tested in various ways. One possible method is by performing a random walk based on the random number generated and comparing the thus simulated distribution of the final position of random walks with the theoretical distribution. In this report we check the quality of each of the 64 bits of random integer numbers that are generated using a linear congruential generator of the form

$$r_{n+1} = a * r_n + c \tag{1}$$

with a = 2862933555777941757 and c = 1013904243.

The probablity distribution of ending up at a final position x after taking n steps is given by the expression

$$P_n(x) = \frac{1}{2^n} \frac{n!}{[(n+x)/2!]\,[(n-x)/2!]} \tag{2}$$

On performing Nw random walks, let $Cn(x)$ be the number of times the final position of a n-step random walk is x. The deviation of the simulated distribution from the actual probablity distribution, for the final postition x is defined to be

$$\Delta_n(x) = \frac{Cn(x)}{Nw} - P_n(x) \tag{3}$$

The total devitation of the simulated distribution from the actual distribution is quantified by

$$\Delta = \sqrt{\sum_{-n}^{n} \Delta_n^2(x)} \tag{4}$$

The total deviation Δ will not goto zero for finite Nw and will goto zero as Nw goes to infinity. If we assume that the $Cn(x)$ follows a Poisson distribution,

then we get the relation between Δ and Nw as

$$\Delta = \frac{1}{\sqrt{Nw}} \tag{5}$$

.

Thus, for a good random number generator the quantity $\sqrt{Nw}\Delta$ should approach 1 and should diverge with Nw in the case of bad random number generators.

2 Program

A fortran 90 programme was written to test the randomness of each bit of the random integers generated by the generator (1). The programme takes the values for n, Nw and the seed for the generator from the file 'read.in'. 64 n-step random walks are performed simultaneously based on the value of each of the 64 bits of the random number generated. A forward step is taken in a random walk if the value of the bit is 1 and a backward step on the contrary. Thus Nw number of random walks were performed and distribution of final positions were got for each of the 64 bits.

The actual probablity distribution was calculated from the equation (2). The factorials were calculated using a truncated stirling's series of the form given below.

$$ln\,(n!) = \left(n + \frac{1}{2}\right) * ln\,(n) - n + \frac{1}{2} ln\,(2\pi) + \frac{1}{12n} \tag{6}$$

The total deviation of thus simulated distribution was calculated for random walks corresponding to each of the bits and was written to a file 'd.dat'. The simulated and actual probablity distribution was written to the files by names- 'p00.dat' to 'p63.dat' with 00 corresponding to 0th bit and 63 to 63rd bit.

3 Results and Discussion

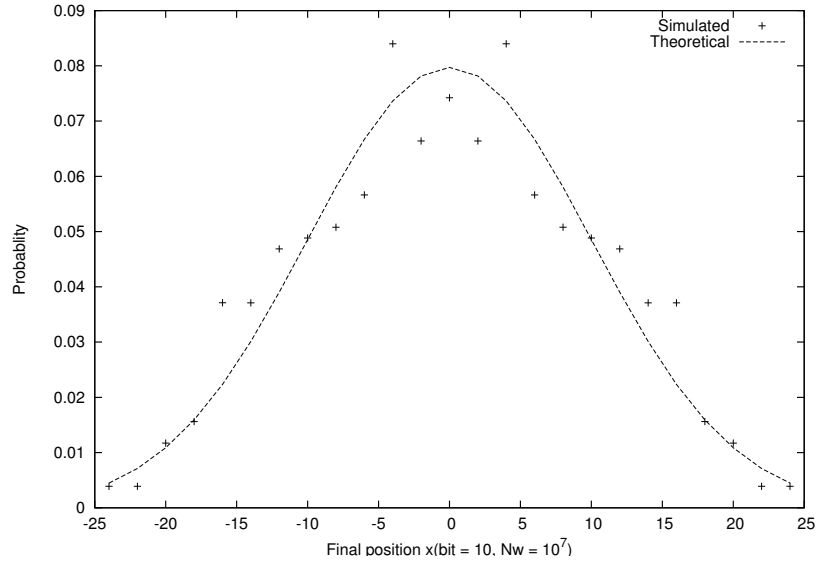
The programme was run for $n = 100$ and for Nw values ranging from 10^2 to 10^7. Some sample graphs of the simulated distribution for particular bits are plotted in figs. 1 to 4. The graphs show that there is good agreement between the simulated and theoretical distributions except for the case of bit number 10 where the simulated distribution is significantly off from the theoretical prediction.

The variation of $\Delta * \sqrt{Nw}$ with the bit number is plotted for various Nw in fig.5. The salient features that are to be noted from the graphs are as follows

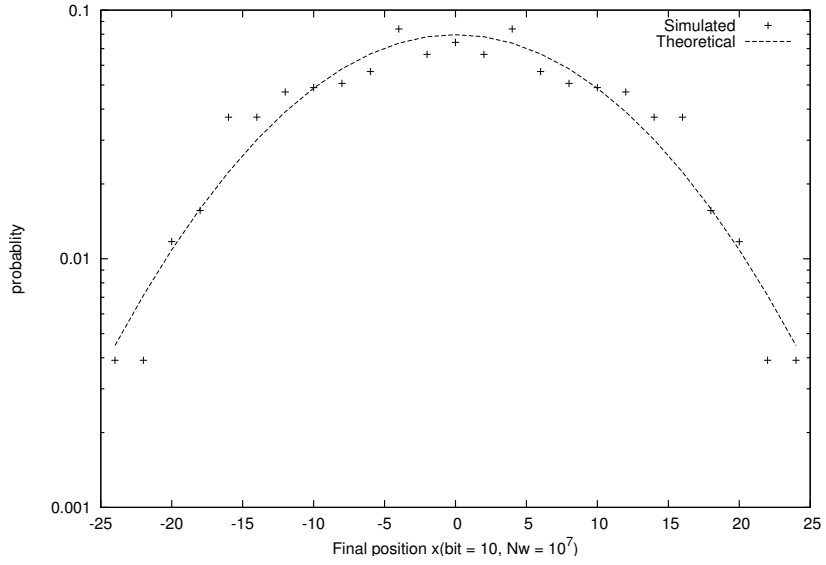
- The quantity $\Delta * \sqrt{Nw}$ diverges as we increase Nw for bits less than bit number 10. The divergence of Δ with Nw indicates that the values of these bits dont behave randomly on the scale of Nw. In other words, the period of the sequence generated by these bits is less than Nw.

- We see that for bit numbers 10-20, the value of $\Delta * \sqrt{Nw}$ approaches one for lesser Nw and starts diverging as we increase Nw. This indicates that these bits have a period that is close to Nw in the range that we have simulated.
- For bits greater than bit number 20, we see that the value of $\Delta * \sqrt{Nw}$ approaches 1 for all the cases of Nw and is still 1 for the case of $Nw = 10^7$), thus indicating that the period of these bits are greater than 10^7.

We see that the bits greater than $20th$ are random on the scale of 10^7. Thus, when we convert the generated integers linearly into numbers between 0 and 1, these floating point numbers will not be random at the precision of $\frac{2^{20}}{2^{63}}$ which is $2^{-41} \simeq 10^{-13}$. Thus, we can get good enough random numbers with this generator upto a precision of 10^{-13}.

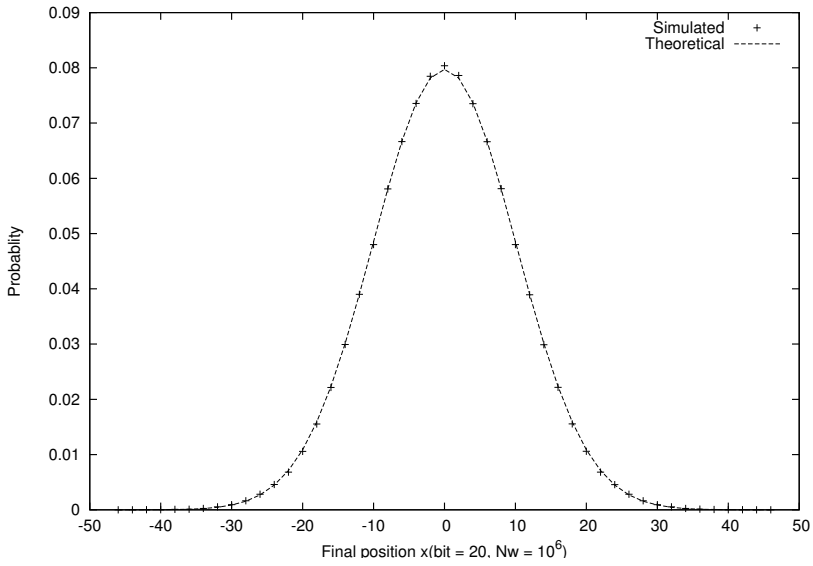


(a)

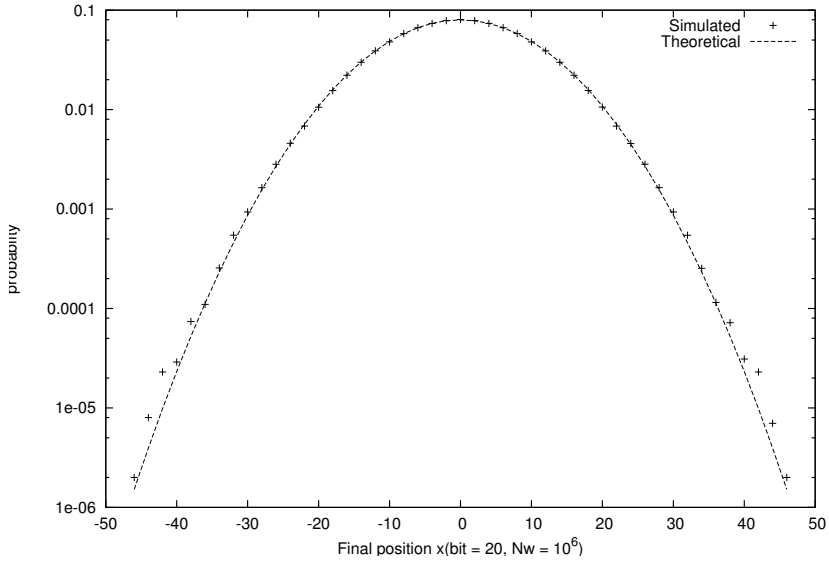


(b)

Figure 1: Simulated and Theoretical probablity distribution for $Nw = 10^7$ and bit number 10.

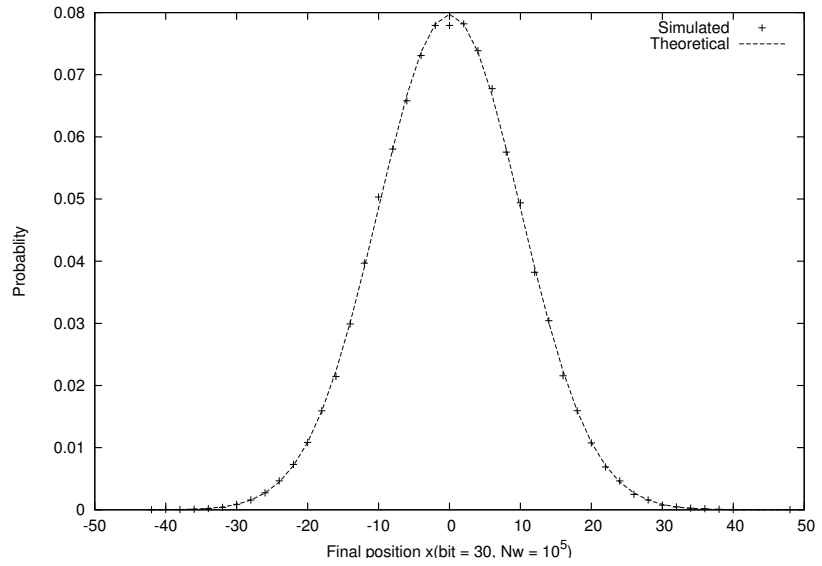


(a)

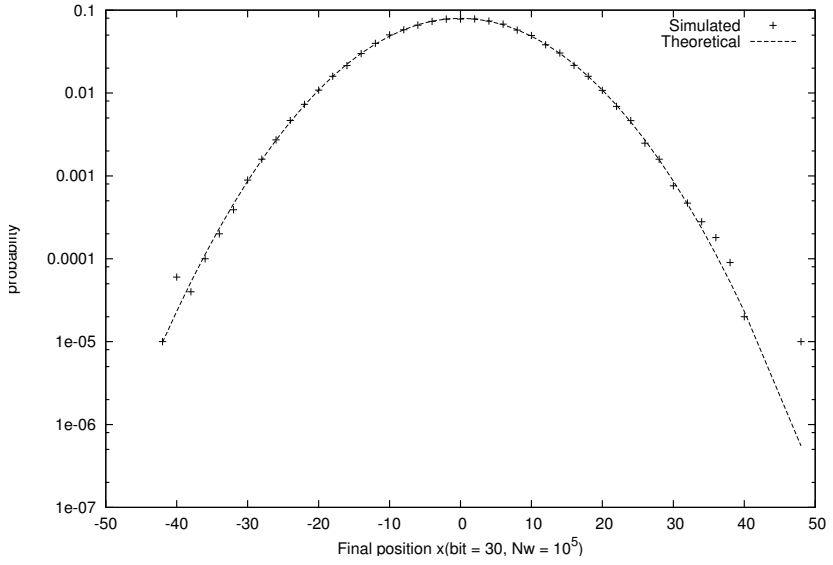


(b)

Figure 2: Simulated and Theoretical probablity distribution for $Nw = 10^6$ and bit number 20.

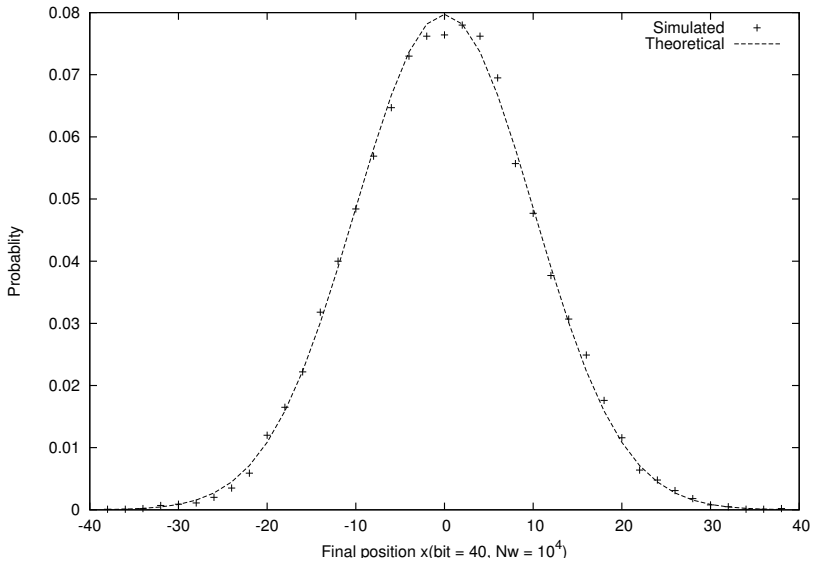


(a)

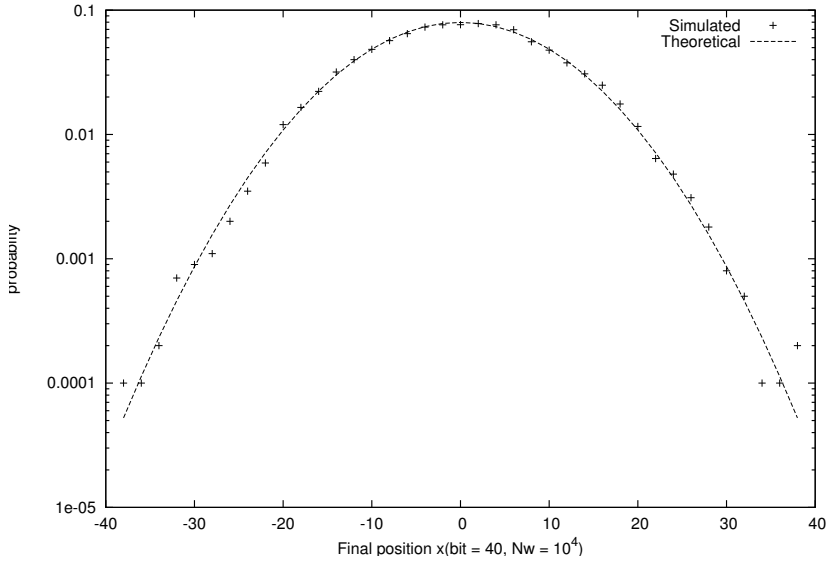


(b)

Figure 3: Simulated and Theoretical probablity distribution for $Nw = 10^5$ and bit number 30.



(a)



(b)

Figure 4: Simulated and Theoretical probablity distribution for $Nw = 10^4$ and bit number 40.

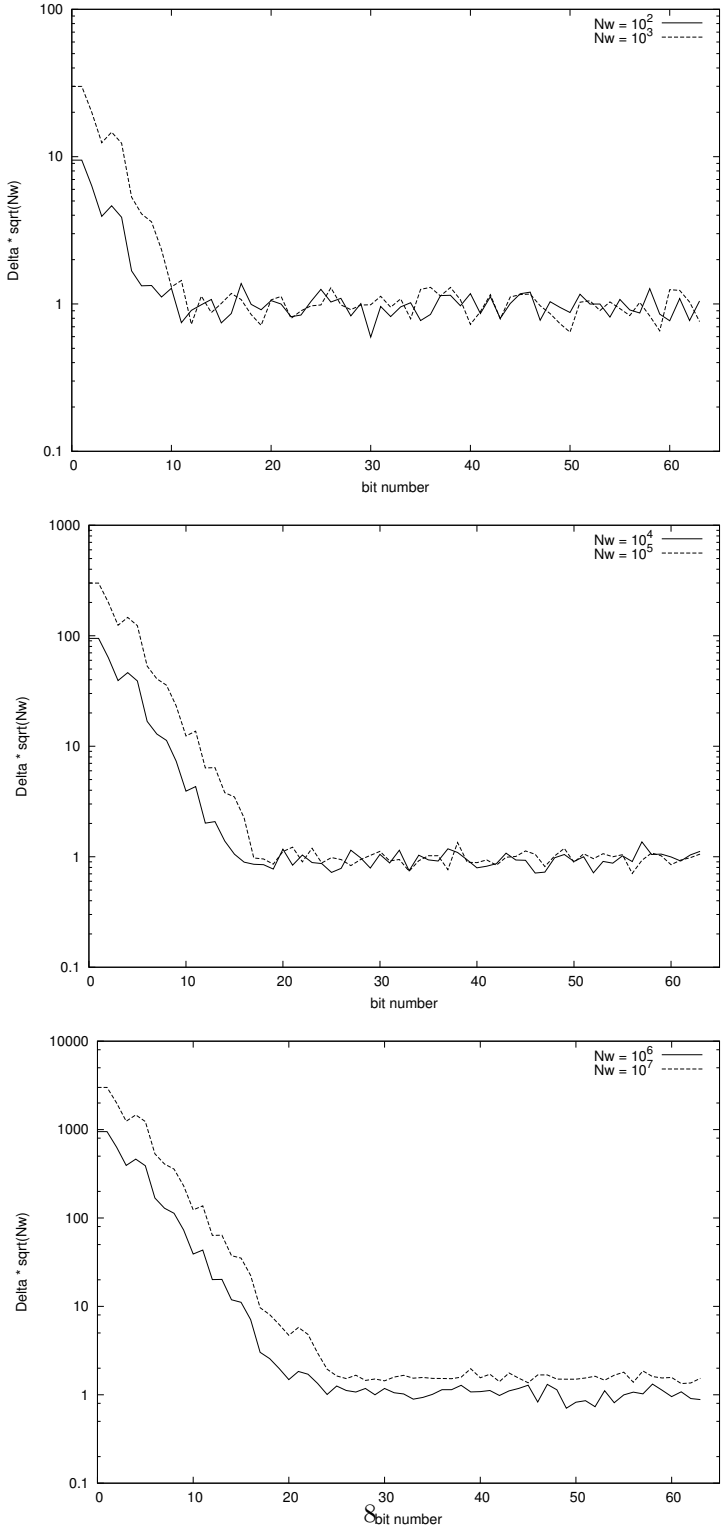


Figure 5: Variation of $\Delta * \sqrt{Nw}$ againt bit nmumber