Investigating whether the inputs of a Multiplicative Congruential Generator can be determined using the outputs

Yohwllo

March 10th 2024

Contents

1 Introduction

Ever since I was a young child the concept of random has amazed me. For randomness could at times not seem random, for example, number generators more often than not are far from random, instead they are psuedo random number generators, which are not truely random. These pesudo random number genrators are not truely random, but rather just a mathematical equation which is being done itteratively, and if the developer was forward thinking, it used a different seed every time too.
Now among random number generators there is a particularly common one called the Multiplicative Congruential Generator (MCG for short), it has been extensively used for various tasks that could require a lot of random numbers, that only need to be 'random enough'. For example creating test data, or for things such as dice rolling in games. There is also another type of random number genreatofr similar to it called the Linear Congruential Generator which is very similar in terms of how it functions.

2 background

2.1 Random Number Generators

2.1.1 Multiplicitve Random Number Generator

Usually an MCG works based off of the equation $x_{i+1} = (a * x_i) \bmod m$. Where the next value is equal to the current value multiplied by a constant, and then its modulus is set as the remainder. [1] [4]

2.1.2 Linear Congruential Generators

In contrast an LCG (Linear Congruential Generator) works in a very similar way, just adding a constant. Its equation looks like $x_{i+1} = (a * x_i + b) \bmod m$ [1]

2.1.3 How LCG/MCG's are used

Linear/Multiplicative congruential generators are used in a variety of settings, from monte carlo simulations [4], to the shuffle function in music players [2], these are jus the implmentations that are public and known about. Considering that these types of random number generators are also used in glibc's [6] random function, no one knows just how many things use these types of generators for their random numbers. So the implication of these random number generators not being random would have an unknown impact on the world.

2.1.4 Flaws of LCG/MCG's

Now these types of linear congruential generators have some flaws that were found by Gorge Marsaglia [4]. That is, if you know how to arange these numbers on a 3d object, all of said numbers will fall upon $\sqrt[n]{n! \cdot m}$ hyper planes [4]. A hyperplane is a plane of $n-1$ dimensions, with n being the number of diemensions. As George Marsaglia put it, "the points are about as randomly spaced in the unit n-cube as the atoms in a perfect crystal at absolute zero". Later on in his paper he goes over a method that can be used to determine the inputs to such an equation.

2.2 Modular Arithmetic

Modular arithmetic, as well as its properties play an important roll in both the generation, and solving of linear/multiplictive congruential generators. Infact the equations themselves use modular arithmetic, so knowing how it works, as well as its properties will prove useful when solving for the variables. [7]

2.2.1 Residues

An important concept in modular arithmetic is residues, a residue is what is left when you subtract the modulo value as much as you can, before subtracing anymore would result in a negative number. It is also called the remainder in the contest of division. An example would be, 10 mod 3, would have a residue of 1, as $10-(3+3+3) = 1$, subtracing anymore would result in a negative number. Thus the residue is 1. It is also important to note that a residue can be 0, such is the case of 12 mod 3, where $12-(3+3+3+3) = 0$, this is a case where the residue is equal to 0. [7]

2.2.2 Congruence

We say a number, or equation is congruent to one another when all of residues/remainders of that value modulo a constant are the same. It is often shown through the sign $\equiv$. An example of it being used correctly would be the equation; $2 \equiv 7 \equiv 12(\text{mod}5)$, as each of the values 5 mod 5 will have the same result, meaning they are congruent. [7]

2.2.3 Relations

The relation between $x \equiv b(\text{mod}m)$ and $x \equiv b(\text{mod}m)$. The first one is for equivalence, which i sthe same as equality. In comparision the second equation is equality. As a note from Cornell University put it;
"$x = b(\text{mod}m)$ is the smallest positive solution to the equation $x \equiv b(\text{mod}m)$" [8]

2.2.4 Rules to note:

Sum Rule: if $a \equiv b(\text{mod}m)$ then $a+c \equiv b+c(\text{mod}m)$ [8]

Multiplication Rule: if $a \equiv b(\text{mod}m)$ and if $c \equiv d(\text{mod}m)$ then $a \cdot c \equiv b \cdot d(\text{mod}m)$ [8]

2.2.5 Otherways It Can Be Written

Another way which $a \equiv b(\text{mod}m)$ can be written is $a = k \cdot m + b$, where k is an arbitary integer. Yet another way it can be written is $n|(a-b)$, which means, $a-b$ is a multiple of n. This becomes very useful when solving for the variables later on.

Some asdclaisdm [3]
Department [1]
noi [5]

References

[1] University Of Waterloo Math Department. generating random numbers. https://wiki.math.uwaterloo.ca/statwiki/index.php?title=generating_Random_Numbers#Inverse_Transform_Method, 2009.

[2] Juk Developers. playlist.cpp. https://github.com/KDE/juk/blob/master/playlist.cpp#L676, 2024.

[3] Michel Goossens, Frank Mittlebach, and Alexander Samarin. *The Latex Companion A*. Addison-Wesley, Reading, Massachusetts, 1993.

[4] George Marsaglia. Random numbers fall mainly in the planes. *Mathematics research Laboratory, Boeing Scientific Research Laboratories*, 1968.

[5] noi. *the whack man*. nomanssky, reading Massachusetts, 2001.

[6] Huzaifa Sidhurwala. Understanding random number generators and their limitations, in linux. *Red Hat Blog*, 2019.

[7] Art Of Problem Solving. Modular arithmetic/introduction. https://artofproblemsolving.com/wiki/index.php/Modular_arithmetic/Introduction, 2024.

[8] Cornell University. Everything you need to know about modular arithmetic. https://pi.math.cornell.edu/~morris/135/mod.pdf, 2006.