**ENSAYO DE LOS NUMERO PSEUDOALEATORIOS.**

**ASIGNATURA: SIMULACION**

**DOCENTE: JOSE ANTONIO LOPEZ TELLO**

**ALUMNO: MAURICIO CASILLAS ZENDEJAS**

**GRUPO: 51-T**

**CARRERA: INGENIERIA EN SISTEMAS COMPUTACIONALES**

**NUMERO DE CONTROL: 15560499**

**2.1.- Los números pseudoaleatorios.**

Para poder realizar una simulación que incluya variabilidad dentro de sus eventos, es preciso generar una serie de números que sean aleatorios por sí mismos, y que su aleatoriedad se extrapole al modelo de simulación que se está construyendo. Como puede comprender, en la construcción del modelo los números aleatorios juegan un papel relevante.

Así, una de las primeras tareas que es necesario llevar a cabo consiste en determinar si los números que utilizaremos para "correr" o ejecutar la simulación son realmente aleatorios o no. Por desgracia, precisar lo anterior con absoluta certidumbre resulta muy complicado, ya que para ello tendríamos que generar un número infinito de valores que nos permitiera comprobar la inexistencia de correlaciones entre ellos. Esto sería muy costoso y tardado, además volvería impráctico el uso de la simulación aun con las computadoras más avanzadas.

A pesar de lo anterior, podemos asegurar que el conjunto de números que utilizaremos en una simulación se comporta de manera m uy similar a un conjunto de números totalmente aleatorios; por ello es que se les denomina números pseudoaleatorios. Casi todas las aplicaciones comerciales tienen varios generadores de números pseudoaleatorios que pueden generar un conjunto m uy grande de números sin mostrar correlación entre ellos. En el presente capítulo discutiremos algunos de los métodos de generación de números pseudoaleatorios, y precisaremos qué características deben tener para emplearlos como una fuente confiable de variabilidad dentro de los modelos. Asimismo, se mostrarán algunas de las pruebas más comunes para comprobar qué tan aleatorios son los números obtenidos con dichos generadores.

**2.2.1.- Algoritmo de cuadrados medios.**

Este algoritmo requiere un número entero detonador (llamado semilla) con D dígitos, el cual es elevado al cuadrado para seleccionar del resultado los D dígitos del centro; el primer número r se determina simplemente anteponiendo el "0." a esos dígitos. Para obtener el segundo r¡ se sigue el mismo procedimiento, sólo que ahora se elevan al cuadrado los D dígitos del centro que se seleccionaron para obtener el primer r Este método se repite hasta obtener **n** números r A continuación se presentan con más detalle los pasos para generar números con el algoritmo de cuadrados medios.

1. Seleccionar una semilla (X₀) con D dígitos (D > 3).

2. Sea Y₀ = resultado de elevar X₀ al cuadrado; sea X₁= los D dígitos del centro, y sea r¡ = 0. D dígitos del centro.

3. Sea Y₁. = resultado de elevar X al cuadrado; sea XM = los D dígitos del centro, y sea r. = 0. D dígitos del centro para toda /= 1, 2,3 ... n.

4. Repetir el paso 3 hasta obtener los n números r. deseados.

Generar los primeros 5 números r¡ a partir de una semilla X0 = 5735, de donde se puede observar que D = 4 dígitos.

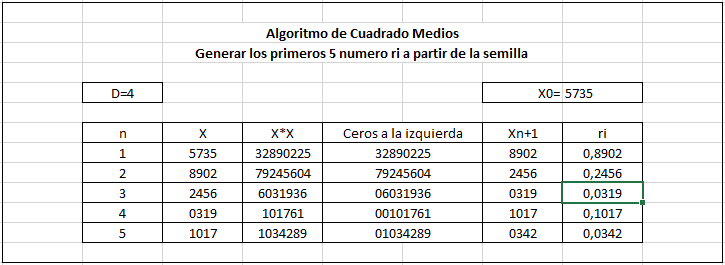
Solución:

Y0 = (5735)2 = 32890225 X, = 8902 r, = 0.8902

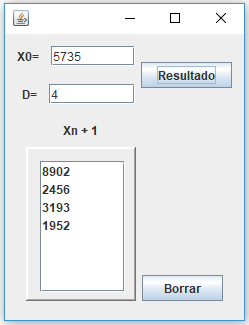
Y1 = (8902)2 = 79245604 X2 = 2456 r2 = 0.2456

Y2 = (2456)2 =06031936 X3 = 0319 r3 = 0.0319

Y3 = (0319)2 = 101761 X4 = 0176 r4 = 0.0176



**Programa en Java (Netbeans)**



**2.2.2.- Algoritmo de productos medios.**

La mecánica de generación de números pseudoaleatorios de este algoritmo no congruencial es similar a la del algoritmo de cuadrados medios. La diferencia entre ambos radica en que el algoritmo de productos medios requiere dos semillas, ambas con D dígitos; además, en lugar de elevarlas al cuadrado, las semillas se multiplican y del producto se seleccionan los D dígitos del centro, los cuales formarán el primer número pseudoaleatorio r, = 0.D dígitos. Después se elimina una semilla, y la otra se multiplica por el primer número de D dígitos, para luego seleccionar del producto los D dígitos que conformarán un segundo número rr Entonces se elimina la segunda semilla y se multiplican el primer número de D dígitos por el segundo número de D dígitos; del producto se obtiene el tercer número r,. Siempre se irá eliminando el número más antiguo, y el procedimiento se repetirá hasta generar los n números pseudoaleatorios.

A continuación, se presentan con más detalle los pasos del método para generar números con el algoritmo de producto medios.

1. Seleccionar una semilla (X0) con D dígitos (D > 3)

2. Seleccionar una semilla (X,) con D dígitos (D > 3)

3. Sea Y0= X ^ X, sea X2= los D dígitos del centro, y sea r¡ = 0.D dígitos del centro.

4. Sea Y = X \*X m; sea XV2= los D dígitos del centro, y sea r/+1 = 0.D dígitos del centro

para toda / = 1, 2, 3 ... n.

5. Repetir el paso 4 hasta obtener los n números r, deseados.

Nota: Si no es posible obtener los D dígitos del centro del número Yf agregue ceros a la izquierda del número Yr

Generar los primeros 5 números r, a partir de las semillas X0= 5015 y X, = 5734; observe que ambas semillas tienen D = 4 dígitos.

Solución:

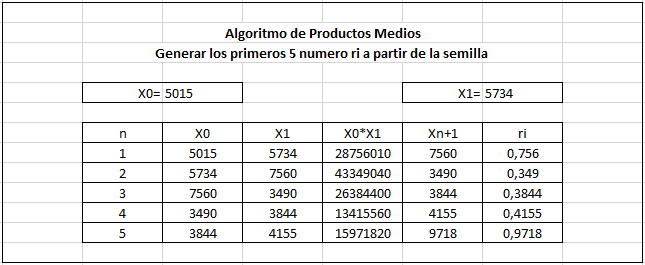
v; = (5015) (5734) = 28756010 X 2 = 7560 r, = 0.7560

Y} = (5734) (7560) = 43349040 X 3 = 3490 r2 = 0.3490

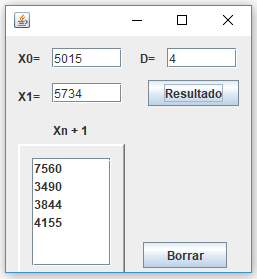
Y2 = (7560) (3490) = 26384400 X4 = 3844 r3 = 0.3844

Y3 = (3490) (3844) = 13415560 X 5 =4155 r4 = 0.4155

Y4 = (3844) (4155) = 15 971820 X6 = 9718 r5 = 0.9718



**Programa en Java (Netbeans)**



**2.2.3.- Algoritmo de multiplicador constante.**

Este algoritmo no congruencial es similar al algoritmo de productos medios. Los siguientes son los pasos necesarios para generar números pseudoaleatorios con el algoritmo de multiplicador constante.

1. Seleccionar una semilla (X0) con D dígitos (D > 3).

2. Seleccionar una constante (a) con D dígitos (D > 3).

3. Sea Y0= a \*X ¿sea X 1 = los D dígitos del centro, y sea r = O.D dígitos del centro.

4. Sea Y, = a\*X¡; sea X^, = los D dígitos del centro, y sea rM = O.D dígitos del centro

para toda / = 1, 2, 3 ... n.

5. Repetir e l paso 4 hasta obtener los n números r. deseados.

Generar los primeros 5 números r. a partir de la semilla X 0 = 9803 y con la constante a = 6965. Observe que tanto la semilla como la constante tienen D = 4 dígitos.

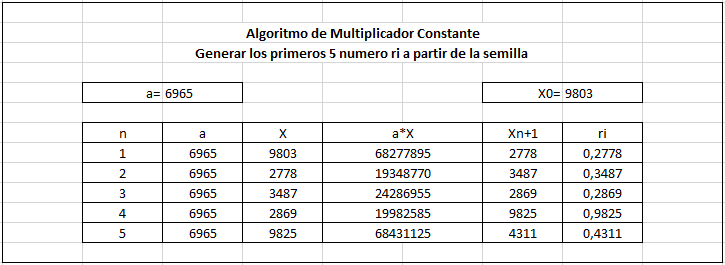
Solución:

Y0 = (6965) (9803) = 68277895 X, = 2778 r, =0.2778

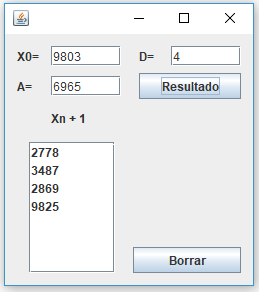
V, = (6965) (2778) = 19348770 X2 = 3487 r2 =0.3487

Y2 = (6965) (3487) = 24286955 X3 = 2869 r3 =0.2869

Y3 = (6965) (2869) = 19982585 X4 = 9825 r4 = 0.9825

YA = (6965) (9825) = 68431125 X5 = 4311 r5 =0.4311

**Programa en Java (Netbeans)**



**2.2.4.- Algoritmo lineal.**

Este algoritmo congruencial fue propuesto por D. H. Lehmer151 en 1951. Según Law y Kelton,PI no ha sido el más usado. El algoritmo congruencial lineal genera una secuencia de números enteros por medio de la siguiente ecuación recursiva:

X + 1 = (ax¡ + c)mod(m) / = 0 , 1, 2, 3 , . . n

donde XQ es la semilla, a es la constante multiplicativa, c es una constante aditiva, y m es el módulo. Xo > 0 ,a > 0 ,c > 0 y m > 0 deben ser números enteros. La operación "mod (m)" significa multiplicar^, por a, sum are, y dividir el resultado entre m para obtener el residuo X/+1. Es importante señalar que la ecuación recursiva del algoritmo congruencial lineal genera una secuencia de números enteros S = {0, 1 ,2 , 3 ,..., m - 1}, y que para obtener números pseudoaleatorios en el intervalo (0, 1) se requiere la siguiente ecuación:

X, í = 0, 1 ,2 ,3 ,...,n

Analice el siguiente ejemplo para comprender mejor la mecánica del algoritmo congruencial lineal:

Generar 4 números entre 0 y 1 con los siguientes parámetros: X0= 37, a = 19, c = 33 y m = 100.

Solución:

X 1 = (19\*37 + 33) mod 100 = 36 r, = 36/99 = 0.3636

X 2 = (19\*36 + 33) mod 100 = 17 r2 = 17/99 = 0.171 7

X 3 = (19\* 17 + 33) mod 100 = 56 r3 = 56/99 = 0.5656

X 4 = (19\*56 + 33) mod 100 = 97 r4 = 97/99 = 0.9797

En el ejemplo anterior se dieron de manera arbitraria cada uno de los parámetros requeridos: Xq, a, c, m. Sin embargo, para que el algoritmo sea capaz de lograr el máximo periodo de vida N, es preciso que dichos parámetros cum plan ciertas condiciones. Banks, Carson, Nelson y Nicol111 sugieren lo siguiente:

m = 29

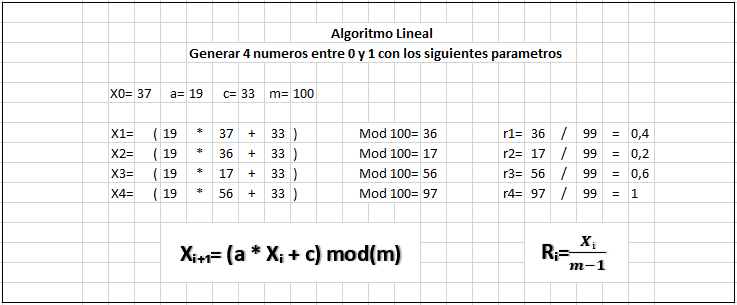
a = 1 + 4k

k debe ser entero

c relativamente primo a m

g debe ser entero

Bajo estas condiciones se obtiene un periodo de vida máximo: N = m = 2g.



**2.2.5 Algoritmo congruencial multiplicativo.**

El algoritmo congruencial multiplicativo surge del algoritmo congruencial lineal cuando c = 0. Entonces la ecuación recursiva es:

XM = {aX) mod (m) i = 0 , 1 ,2 ,3 ,...,n

En comparación con el algoritmo congruencial lineal, la ventaja del algoritmo multiplicativo es que implica una operación menos a realizar. Los parámetros de arranque de este algoritmo son XQ, a y m, los cuales deben ser números enteros y mayores que cero. Para transformar los números X. en el intervalo (0,1) se usa la ecuación r¡= x l/(m - 1 ). De acuerdo con Banks, Carson, Nelson y Nicol,111 las condiciones que deben cumplir los parámetros para que el algoritmo congruencial multiplicativo alcance su máximo periodo A/, son:

m = 2 i

a = 3 + 8k o a = 5 + 8k

k = 0 , 1, 2, 3 ,...

X0 = debe ser un número impar

g debe ser entero

A partir de estas condiciones se logra un periodo de vida máximo N = k/4 = 2g'2

Generar suficientes números entre 0 y 1 con los siguientes parámetros: X0 = 17, k = 2 y

g = 5, hasta encontrar el periodo o ciclo de vida.

Solución:

a = 5 + 8 (2) = 21 y m = 32

X0= 17

X, =(21\*17) mod 32 = 5 r , =5/31 = 0.612

X 2 = (21\*5) mod 32 = 9 r2 = 9/31 = 0.2903

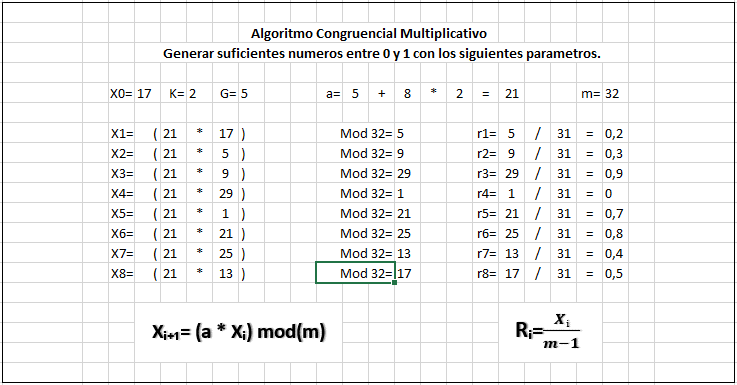
X 3 = (21 \*9) mod 32 = 29 r3 = 29/31 = 1.9354

X4 = (21\*29) mod 32 = 1 r4 = 1/31 =0.3225

X 5 =(21\*1) mod 32 = 21 r5 = 21/31 = 0.6774

X 6 = (21\*21) mod 32 = 2 5 r6 = 25/31 = 0.8064

X? = (21\*25) mod 32 = 13 r? = 13/31 = 0.4193

X8 = (21\*13) mod 32 = 17 r8= 17/31 = 0.5483

**2.2.6.- Algoritmo congruencial aditivo**

Este algoritmo requiere una secuencia previa de n números enteros X,, X2, X3,X4, ...,X n para generar una nueva secuencia de números enteros que empieza en X ^ , Xn^ Xn i, Xn+A, ... Su ecuación recursiva es:

X = (x/+1 +X.\_n) mod (m) /'= n + 1, n + 2, n + 3 ,...A/

Los números r, pueden ser generados mediante la ecuación

r. = x./m - 1

Generar 7 números pseudoaleatorios entre cero y uno a partir de la siguiente secuencia de números enteros: 6 5 ,8 9 , 9 8 ,0 3 ,6 9 ; m = 100.

Sean X, = 65, X2= 89, X3 = 98, X4 = 03, X5 = 69. Para generar r}, r2, r3, r4, ry r6 y r7 antes es necesario generar X^ X7, X8, Xg, X 10, X ,,, X12.

Solución:

X6 = (X5 + X ,)m o d 100 = (69 + 65) mod 100 = 34

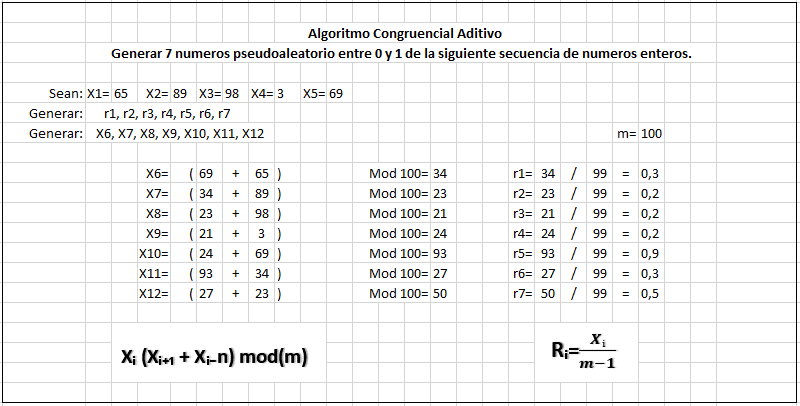
X 7 = (X6 + X2) mod 100 = (34 + 89) mod 100 = 23

X8 = (X7 + X3) mod 100 = (23+ 98) mod 100 = 21

X 9 = (X8 + X4) mod 100 = (21 + 03) mod 100 = 24

X 10= (X 9 + X5) mod 100 = (24 + 69) mod 100 = 93

X u = (X10 + X J mod 100 = (9 3 + 3 4 ) mod 1 0 0 = 2 7



**2.2.7 Algoritmos congruenciales no lineales.**

En esta sección se analizarán dos algoritmos congruenciales no lineales: el congruencial cuadrático y el algoritmo presentado por Blum, Blum y Shub.121

**2.2.7.1.- Algoritmo congruencial cuadrático.**

Este algoritmo tiene la siguiente ecuación recursiva:

XM = (aX¡2 + bX¡ + c ) mod (m) i = 0 ,1, 2,3 , ...A/

En este caso, los números pueden ser generados con la ecuación r¡= x ¡/ ( m - 1). De acuerdo con L'Ecuyer,141 las condiciones que deben cumplir los parámetros m , a, b y c para alcanzar un periodo máximo de N = m son:

m = 2<>

a debe ser un número par

c debe ser un número impar

g debe ser entero

(b - 1) mod 4 = 1

De esta manera se logra un periodo de vida máximo N = m.

Generar, a partir del algoritmo congruencial cuadrático, suficientes números enteros hasta alcanzar el periodo de vida, para esto considere los parámetros Xq = 13, m = 8 , a = 26, b = 27 y c = 27. Como todas las condiciones estipuladas para los parámetros se satisfacen, es de esperarse que el periodo de vida del generador sea N = m = 8 , tal como podrá comprobar al revisar los cálculos correspondientes, que se presentan a continuación.

Solución:

X, = (26\*132 + 27\*13 + 27) mod (8 ) = 4

X 2 = (26\*42 + 27\*4 + 27) mod (8 ) = 7

X 3 = (26\*72 + 27\*7 + 27) mod (8 ) = 2

X4 = (26\*22 + 27\*2 + 27) mod (8 ) = 1

X 5 = (26\*12+27\*1 + 27) mod (8 ) = 0

X6 = (26\*02 + 27\*0 + 27) mod (8 ) = 3

X 7 = (26\*32 + 27\*3 + 27) mod (8 ) = 6

X8 = (26\*62 + 27\*6 + 27) mod (8 ) = 5

X 9 = (26\*52 + 2 7 \*5 + 2 7 ) mod (8 ) = 4

Por otro lado, el algoritmo cuadrático genera una secuencia de números enteros S = {0 ,1 ,2, 3 , m - 1}, al igual que el algoritmo congruencial lineal.

**2.2.7.2 Algoritmo de Blum, Blum y Shub.**

Si en el algoritmo congruencial cuadrático a = ‘\ ,b = 0 y c = 0, entonces se construye una nueva ecuación recursiva:

X í+1 = (X .2)m od (m ) / = 0,1,2,3 ,...n

La ecuación anterior fue propuesta por Blum, Blum y Shub como un nuevo método para generar números que no tienen un comportamiento predecible.