






# Generación de números (seudo)aleatorios

---

Patricia Kisbye

Modelos y Simulación

# Referencias bibliográficas

-  OSCAR BUSTOS Y ALEJANDRO FRERY, *Simulacao estocastica : teoria e algoritmos (versao completa)*. Series Monografías en Matemática. Vol 42. Edit. IMPA. 1992.
-  DONALD E. KNUTH, *Seminumerical Algorithms. The Art of Computer Programming*. Vol 2. Edit. Addison-Wesley. 1998.
-  PIERRE L'ECUYER, *Efficient and portable combined random number generators*, Communications of the ACM, (1988) 31 (6)
-  GEORGE MARSAGLIA AND ARIF ZAMAN, *Some portable very-long-period random number generators*, Computers in Physics, (1994) 8 (6).  
*Efficient and Portable Combined Random Number Generators*, Communications of the ACM, (1988) 31(6), pp.742–774.
-  *Numerical recipes in C: The Art of Scientific Computing*. Cambridge University Press. (1988-1992)

# Secuencias de números aleatorios

En simulación estocástica los generadores de números (pseudo)aleatorios con distribución uniforme en el intervalo  $[0, 1]$  son empleados para diversos usos:

1. en forma directa, es decir, porque se desea obtener valores uniformemente distribuidos en  $[0, 1]$ ,
2. para generar muestras de otras variables aleatorias, con distribuciones discretas o continuas,
3. para generar muestras de un conjunto de variables aleatorias dependientes, como por ejemplo procesos estocásticos y distribuciones multivariadas.

# Qué se espera de un generador

Por **generador de números pseudoaleatorios** entenderemos un algoritmo capaz de producir secuencias de números:

$$u_1, u_2, \dots, u_N,$$

que sean realizaciones de muestras de tamaño  $N$  de una variable aleatoria uniforme continua  $U \sim \mathcal{U}(0, 1)$ .

Para ello debemos acordar qué entendemos por una **muestra** de esta variable.

## Muestra de una distribución uniforme $\mathcal{U}(0, 1)$

Una  $N$ -upla de variables aleatorias  $(U_1, U_2, \dots, U_N)$  es una muestra de tamaño  $N$  de una variable aleatoria uniforme en  $(0, 1)$  si cumple:

1. Para cada  $i = 1, \dots, N$  y cada  $u \in \mathbb{R}$  vale

$$\mathbb{P}(U_i \leq u) = \begin{cases} 0 & \text{si } u < 0 \\ u & \text{si } 0 \leq u \leq 1. \\ 1 & \text{si } u > 1. \end{cases}$$

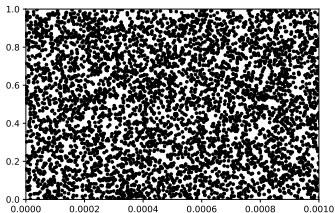
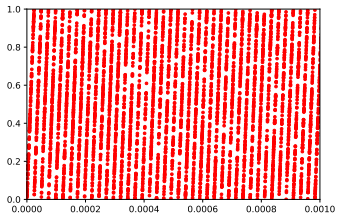
2. Para cada  $k$ ,  $2 \leq k \leq N$  y cada  $k$ -upla  $(i_1, i_2, \dots, i_k)$  con  $1 \leq i_1 < \dots < i_k \leq N$  vale que

$$\mathbb{P}(U_{i_1} \leq u_1, \dots, U_{i_k} \leq u_k) = \mathbb{P}(U_{i_1} \leq u_1) \dots \mathbb{P}(U_{i_k} \leq u_k),$$

cualesquiera sean  $u_1, \dots, u_k$ ,

# Muestra de una distribución uniforme

Los siguientes gráficos muestran pares  $(U_i, U_{i+1})$  de dos generadores de números pseudoaleatorios.



Antes de las computadoras, existieron diferentes métodos para generar secuencias de números aleatorios:

1. Procedimientos físicos (monedas, dados, bolilleros, ...).
2. (1927) Tippett: tabla de 41600 dígitos agrupados en 10400 números aleatorios de cuatro cifras.
3. (1939) Kendall y Babbington: dispositivo mecánico. Tabla de 100.000 números aleatorios.
4. (1955) Rand Corporation: ruido electrónico. Tabla de 1 millón de números aleatorios.

# Inconvenientes de tablas y métodos físicos

1. No puede repetirse una misma secuencia.
2. No hay velocidad computacional.
3. Incorporar una tabla a la computadora implica gran costo de almacenamiento en relación a la cantidad de números.

Con la aparición de las computadoras, surgen métodos de generación de secuencias de números aleatorios.



# Propiedades deseables de un generador

Un algoritmo de generación de números aleatorios razonable debe cumplir:

- **repetibilidad**  
Al repetir los parámetros del generador, se repite la secuencia.
- **portabilidad**,  
El algoritmo no debe depender del lenguaje computacional ni de la computadora utilizada.
- **velocidad computacional**.
- La secuencia generada debe ser **intuitivamente** aleatoria.
- Esa aleatoriedad debe ser establecida teóricamente o, al menos, debe pasar ciertos **tests de aleatoriedad**.
- Deben conocerse las **propiedades teóricas del generador**.

Secuencia de von Neumann (1946):

1.  $X_0$ : número de 4 dígitos. (1234)
2.  $X_i^2$ : escrito con 8 dígitos. (01522756)
3.  $X_{i+1}$ : 4 dígitos centrales. (5227)
4. Volver a 2

# Un ejemplo

Secuencia de von Neumann (1946):

1.  $X_0$ : número de 4 dígitos. (1234)
2.  $X_i^2$ : escrito con 8 dígitos. (01522756)
3.  $X_{i+1}$ : 4 dígitos centrales. (5227)
4. Volver a 2

1234, 5227, 3215, 3362, 3030, 1809, 2724, 4201, 6484, 422, 1780

Otras secuencias:

2100, 4100, 8100, 6100, 2100 ...

3792, 3792, 3792, 3792, ...

4138, 1230, 5129, 3066, 4003, 240, 576, 3317, 24, 5, 0, 0, 0, ...

**No tiene buenas propiedades.**

# Generador congruencial lineal

- $a, c, M, y_0 \in \mathbb{Z}$ .

$$y_i = ay_{i-1} + c \mod M, \quad i \geq 1,$$

La secuencia en  $[0, 1)$  se obtiene dividiendo los valores  $y_i$  por  $M$ :

$$x_i = \frac{y_i}{M}.$$

- $y_0$ : semilla
- $a$ : multiplicador
- $c$ : incremento
- $M$ : módulo
- generador *mixto*:  $c \neq 0$
- generador *multiplicativo*:  $c = 0$ .

- La secuencia

$0, 1, 6, 15, 12, 13, 2, 11, 8, 9, \dots$

fue generada por un generador congruencial lineal. ¿Cuál será el próximo número (entre 0 y 15)?

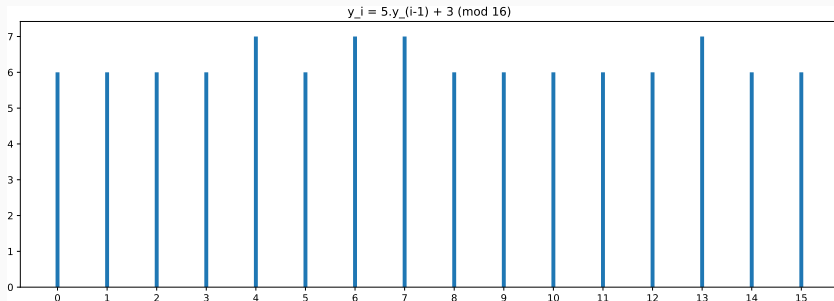
- La secuencia

$1, 12, 1, 12, 1, 12, 1, 12, 1, 12, \dots$

ha sido generada por otro generador congruencial. ¿Podría decirse que genera una secuencia intuitivamente aleatoria?

# Ejemplo

Por ejemplo, para  $y_i = 5y_{i-1} + 3 \pmod{16}$ ,  $y_0 = 3$ , una secuencia de 100 valores se distribuye de acuerdo al siguiente histograma:



## Período de la secuencia

$$y_{i+1} = a y_i + c \mod M$$

- El menor número  $K$  tal que  $y_{n+K} = y_n$  para todo  $n \geq N_0$  es el **período** de la secuencia. Se refiere a la cantidad de números de la secuencia que se suceden hasta que se vuelve a repetir la misma secuencia.

3, 4, 25, 16, 7, 28, 19, 10, 1, 22, 13, 4, 25, 16,

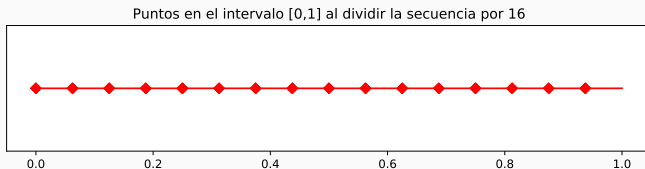
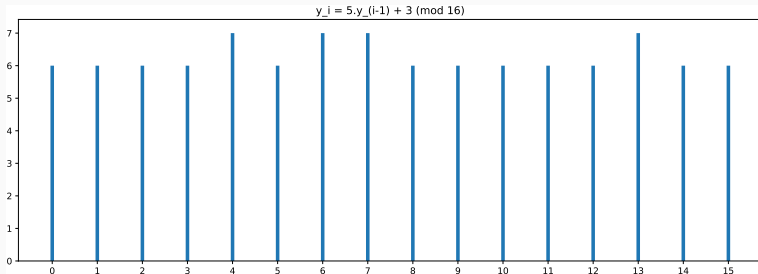
3, 18, 17, 12, 6, 14, 16, 7, 0, 3, 18, 17,

1, 8, 11, 10, 5, 12, 15, 14, 9, 0, 3, 2, 13, 4, 7, 6, 1, 8

- Todo generador congruencial genera secuencias de período finito.
- El período de una secuencia está acotado por  $M$ .
- Mayor período no implica mayor aleatoriedad.

# Ejemplo

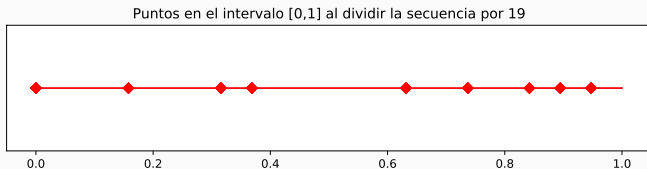
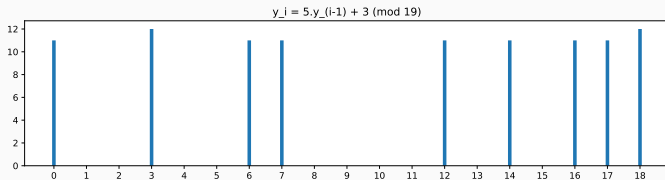
Por ejemplo, para  $y_i = 5y_{i-1} + 3 \pmod{16}$ ,  $y_0 = 3$ , una secuencia de 100 valores se distribuye de acuerdo a las siguientes frecuencias:





# Ejemplo

Para  $y_i = 5y_{i-1} + 3 \pmod{19}$ ,  $y_0 = 3$ , una secuencia de 100 valores se distribuye de acuerdo al siguiente histograma:



## Elección de $a$ , $c$ y $M$

$$y_{i+1} = a y_i + c \mod M$$

Las buenas propiedades del generador congruencial dependen de una elección apropiada de  $a$ ,  $c$  y  $M$ , y en algunos casos  $y_0$ .

- La elección de  $M$  se relaciona con: longitud de la secuencia y velocidad computacional.
- La elección de  $a$  y  $c$ , en función de  $M$ , influyen en el período  $K$ .
- Las propiedades de aleatoriedad deben ser testeadas.

# Período máximo

## Período máximo en GC Lineal Mixto

$$y_{i+1} = a y_i + c \pmod{M}, \quad c \neq 0$$

tiene período  $M$  si y sólo si

1.  $m.c.d.(c, M) = 1$
2.  $a \equiv 1 \pmod{p}$ , para cualquier factor primo  $p$  de  $M$ .
3. Si  $4 \mid M$ , entonces  $a \equiv 1 \pmod{4}$ .

**Ejemplo:**  $y_{i+1} = 5 y_i + 3 \pmod{8}$

3, 2, 5, 4, 7, 6, 1, 0, 3, ...

Corolario: Si  $M$  es primo, el período máximo ocurre sólo si  $a = 1$ .

# Generadores multiplicativos

## Raíz primitiva

$$y_{i+1} = a y_i \mod M, \quad c = 0$$

- $a$  es raíz primitiva de  $M$  si  $a^{(M-1)/p} \not\equiv 1 \mod (M)$  para cualquier factor primo  $p$  de  $M - 1$ .

Ejemplo:  $M = 7$ .

$a = 2$	$a = 3$
$2^1 \equiv 2 \mod 7$	$3^1 \equiv 3 \mod 7$
$2^2 \equiv 4 \mod 7$	$3^2 \equiv 2 \mod 7$
$2^3 \equiv 1 \mod 7$	$3^3 \equiv 6 \mod 7$
$2^4 \equiv 2 \mod 7$	$3^4 \equiv 4 \mod 7$
$2^5 \equiv 4 \mod 7$	$3^5 \equiv 5 \mod 7$
$2^6 \equiv 1 \mod 7$	$3^6 \equiv 1 \mod 7$
$2^7 \equiv 2 \mod 7$	$3^7 \equiv 3 \mod 7$

## Período máximo en GC Lineal Multiplicativo

Para un generador multiplicativo  $y_{i+1} = a y_i \pmod M$ , la longitud  $K$  de la secuencia verifica

1. Si  $K = M - 1$  entonces  $M$  es primo.
2.  $K$  divide a  $M - 1$ .
3.  $K = M - 1$  si y sólo si  $a$  es raíz primitiva de  $M$  y  $M$  es primo.

**Problema:** Encontrar raíces primitivas.

**Propiedad útil:** Si  $a$  es raíz primitiva y  $(k, M - 1) = 1$ , entonces  $a^k$  es raíz primitiva.

## Un ejemplo con $M$ primo

$$M = 2^{31} - 1 \quad a = 16807$$

$$M - 1 = 2^{31} - 2 = 2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331,$$

- $M$ : es un primo de Mersenne.
- 7: raíz primitiva, pero  $7y_i, 7^2y_i, 7^3y_i \dots$  es fácilmente predecible.
- $(5, M - 1) = 1$ , implica que  $7^5 = 16807$  es raíz primitiva.
- secuencia de longitud máxima  $M - 1 = 2\,147\,483\,646$ .

Si  $M = 2^k$ ,  $c = 0$ , tomar módulo es computacionalmente sencillo.

$$y_j = a^j y_0 \mod (2^k)$$

- secuencia de longitud máxima  $= 2^{k-2}$ , para  $a$  raíz primitiva.
- facilita cálculos (desplazamiento de bits).
- fenómeno de **no aleatoriedad** en bits menos significativos.
- RANDU:  $M = 2^{31}$ ,  $a = 2^{16} + 3 = 65539$ .

## Desventaja de un generador congruencial

En una secuencia  $y_1, y_2, \dots$  dada por un generador congruencial **cualquiera**, los puntos

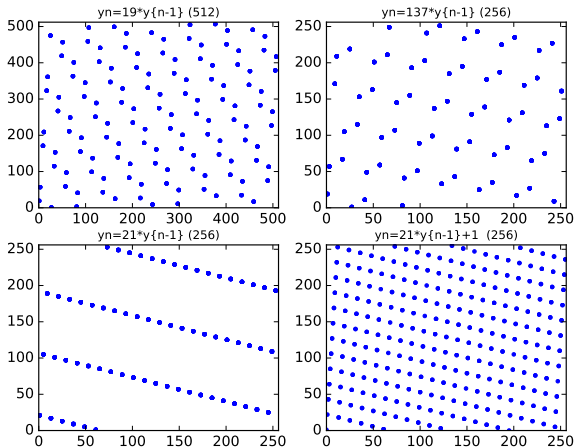
$$(y_j, y_{j+1}, \dots, y_{j+k-1}), \quad j = 0, 1, 2, \dots$$

están ubicados en no más de  $(k!M)^{1/k}$  hiperplanos paralelos.

- Cota máxima:  $(k!M)^{1/k}$ : Estructura de red
- Generador RANDU: Ternas ubicadas en 15 planos paralelos.



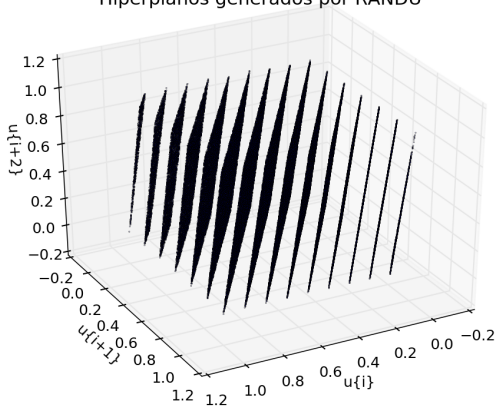
# Hiperplanos. Generadores congruenciales



**Figure 1:** Generadores congruenciales. Distribución de pares  $(y_i, y_{i+1})$

$$y_i = (2^{16} + 3) y_{i-1} \mod (2^{31})$$

Hiperplanos generados por RANDU



## Teorema

Sean  $W_1, W_2, \dots, W_n$  variables aleatorias discretas independientes, tales que  $W_1 \sim U([0, d - 1])$ . Entonces

$$W = \left( \sum_{j=1}^n W_j \right) \bmod d$$

es una v.a. uniforme discreta en  $[0, d - 1]$ .

Ejemplo: tirar 2 dados, y sumar módulo 6.

# Suma de congruenciales

## Generador A:

$$x_i = 4x_{i-1} + 1 \pmod{9}, \quad x_0 = 3, \quad K = 9$$

3, 4, 8, 6, 7, 2, 0, 1, 5, 3,

## Generador B:

$$y_i = 5y_{i-1} + 3 \pmod{13}, \quad y_0 = 3. \quad K = 4$$

3, 5, 2, 0, 3, 5, 2, 0, 3, 5, 2, 0, 3,

## Generador A+B, módulo 9

$$z_i = x_i + y_i \pmod{M}, \quad M = 9. \quad K = 36$$

6, 0, 1, 6, 1, 7, 2, 1, 8, 8, 6, 8, 0, 3, 4, 0, 4, 1, 5, 4, 2, 2, 0,

2, 3, 6, 7, 3, 7, 4, 8, 7, 5, 5, 3, 5, 6,

Tiene período  $m.c.m.(9, 4) = 36$ .

# Combinación de congruenciales

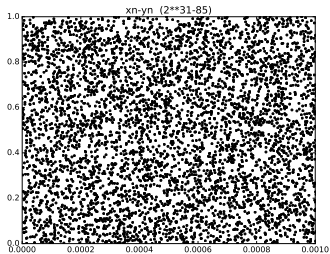
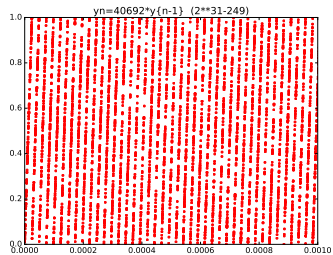
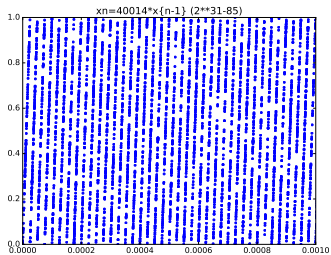
- Combinar secuencias de generadores congruenciales. Sugerencia: restar.
- Se obtiene un generador de v.a. uniformes.
- La longitud de la secuencia es mayor = mínimo común múltiplo de los generadores.

$$x_n = 40014x_{n-1} \mod 2^{31} - 85$$

$$y_n = 40692y_{n-1} \mod 2^{31} - 249$$

El 2 es el único factor común.

$$K \approx 2^{61} = 2\,305\,843\,009\,213\,693\,952 \sim 2.3 \times 10^{18}.$$



Some portable very-long-period random number generators, George Marsaglia and Arif Zaman.

- Sugerencias sobre otros generadores.
- Fibonacci
- Resta con préstamo
- Suma con acarreo
- *Shuffling*

Módulo	Secuencia	Período
$2^{32}$	$x_n = 69069 x_{n-1} + \text{impar}$	$2^{32}$
$2^{32}$	$x_n = x_{n-1} * x_{n-2}$	$2^{31}$
$2^{32}$	$x_n = x_{n-1} + x_{n-2} + C$	$2^{58}$
$2^{31} - 69$	$x_n = x_{n-3} - x_{n-1}$	$2^{62}$
$2^{32} - 18$	$x_n = x_{n-2} - x_{n-3} - C$	$2^{95}$

## El generador mzran( )

Combina los generadores:

$$x_n = 69069x_{n-1} + 1013904243 \mod 2^{32}$$

Período:  $\approx 2^{32}$

$$x_n = x_{n-3} - x_{n-1} \mod 2^{31} - 69$$

Período:  $\approx 2^{94}$

El período es mayor a  $2^{94}$ , o  $10^{28}$ .



## El generador mzran13( )

Combina los generadores:

$$x_n = 69069x_{n-1} + 1013904243 \mod 2^{32}$$

Período:  $\approx 2^{32}$

$$x_n = x_{n-2} - x_{n-3} - c' \mod 2^{32} - 18$$

Período:  $\approx 2^{95}$

El período de la combinación es del orden de  $2^{125}$ , o  $10^{36}$ .

# Mersenne-Twister

- Estructura más compleja que los generadores congruenciales.
- Utiliza corrimiento de bits, *twists*.
- Período:  $K = 2^{19937} - 1$ . (32 \* 624 - 1 = 19937.)

