

Una prueba empírica de generadores de números pseudoaleatorios mediante un proceso de decaimiento exponencial

H.F. Coronel-Brizio, A.R. Hernández-Montoya y M.A. Jiménez-Montaña
*Facultad de Física e Inteligencia Artificial, Universidad Veracruzana,
Apartado Postal 475, Xalapa, Veracruz, México,
e-mail: hcoronel@uv.mx, alhernandez@uv.mx*

L.E. Mora-Forsbach
CIMAT, Apartado Postal 402, Guanajuato, Gto., 36000, México.

Recibido el 23 de octubre de 2006; aceptado el 30 de agosto de 2007

Las pruebas empíricas que usan procesos o modelos físicos para probar generadores de números pseudoaleatorios, complementan las pruebas de aleatoriedad teóricas y han sido usadas con mucho éxito. En este trabajo se presenta una metodología estadística para evaluar la calidad de generadores de números pseudoaleatorios, ilustrando el método en el contexto del proceso de decaimiento radiactivo y utilizando para ello algunos generadores de uso común en la física.

Descriptores: Aleatorio; pseudoaleatorio; simulación Monte Carlo; generador de números pseudoaleatorios; teorema central del límite.

Empirical tests for pseudorandom number generators based on the use of processes or physical models have been successfully used and are considered as complementary to theoretical tests of randomness. In this work a statistical methodology for evaluating the quality of pseudorandom number generators is presented. The method is illustrated in the context of the so-called exponential decay process, using some pseudorandom number generators commonly used in physics.

Keywords: Econophysics; power-law; stable distribution; levy regime.

PACS: 02.50.-r; 02.50.Ng

1. Introducción

La simulación mediante el método de Monte Carlo [1, 2] hace uso intensivo de sucesiones de números aleatorios y es una técnica estándar ampliamente aplicada desde hace más de 50 años en diversas ramas de la física, especialmente en la física de altas energías y en la física estadística. Además de su aplicabilidad en el método de Monte Carlo, el uso y obtención de sucesiones de números aleatorios constituye hoy en día una investigación muy activa con aplicaciones en campos tan diversos como la criptografía, integración Monte Carlo, ecología, identificación biométrica y aún inteligencia artificial [3–9].

Estrictamente hablando, obtener sucesiones de números realmente aleatorios implica la utilización de algún fenómeno físico de naturaleza estocástica, como el arrojar una moneda al aire, el ruido de un circuito electrónico, el decaimiento de un material radioactivo, el conteo de fotones mediante detectores centelladores y, más recientemente, se han propuesto métodos menos tradicionales basados en fenómenos tales como el flujo turbulento de aire formado por el movimiento de los discos duros en una computadora y otro tipo de hardware, péndulos caóticos e incluso del tipo biométricos [10–13], pero debido a las inherentes dificultades que ofrece este enfoque, entre las que podemos mencionar los errores sistemáticos introducidos por el arreglo experimental, la nula reproducibilidad² de la sucesión obtenida, así como la baja frecuencia en la generación de números aleatorios, han hecho necesaria la búsqueda de otras formas más eficientes para obtener estos números.

Desde hace ya algunos años, se utilizan computadoras digitales para implementar programas a los que llamamos generadores de números pseudoaleatorios o simplemente generadores, los cuales mediante reglas deterministas y operaciones aritméticas muchas veces sencillas, producen sucesiones de números que se asemejan en un sentido limitado [14, 15], a las obtenidas mediante un experimento aleatorio y que se denominan sucesiones de números *pseudoaleatorios*.

Se conocen muchas implementaciones diferentes para generar números pseudoaleatorios [14, 15] que hacen uso de una gran variedad de técnicas y algoritmos que comprenden desde el uso de algoritmos de congruencias lineales hasta otros asociados con autómatas celulares, algoritmos de criptografía de curvas elípticas, etcétera [16–19].

La actual utilización de series muy grandes de números pseudoaleatorios en muchas aplicaciones, así como algunos episodios de resultados dudosos, obtenidos debido a la baja calidad de los generadores utilizados [15, 20–22], ha fortalecido la necesidad de contar con mejores y cada vez más eficientes pruebas de la calidad. El campo de investigación de las pruebas de calidad de generadores de números pseudoaleatorios (y por supuesto, también de su implementación), es tan activo que prácticamente no hay mes en el que no se reporten en la literatura científica nuevas pruebas de calidad que utilizan una gran variedad de criterios y técnicas (teoría de la información, técnicas estadísticas, *power spectrum*, *gambling tests*, sistemas físicos, entropía, etc.) [3, 23–31].

Las pruebas de calidad de los generadores de números pseudoaleatorios se pueden dividir en:

- *Pruebas teóricas.* Se realizan estudiando los algoritmos generadores de números pseudoaleatorios mediante el uso de herramientas como la teoría de números. Estos tipos de pruebas son útiles por su generalidad y están basadas en el estudio de algunas propiedades tales como la longitud del periodo de la secuencia y la uniformidad del algoritmo.
- *Pruebas empíricas.* Estas pruebas se concentran en las sucesiones de números pseudoaleatorios y sus propiedades. Son usadas para encontrar correlaciones locales no triviales presentes en las sucesiones de números pseudoaleatorios y mostrar aspectos desapercibidos en las pruebas teóricas.

En los últimos años se han explorado pruebas empíricas basadas en procesos físicos [32, 33], principalmente la caminata aleatoria y el modelo de Ising [28, 29, 32–34].

En este trabajo se presenta un criterio para evaluar a un generador de números pseudoaleatorios dado, sugiriendo que su calidad está directamente relacionada con la capacidad del mismo para producir simulaciones que reproduzcan el comportamiento y propiedades teóricas de un proceso o modelo de referencia. En otras palabras, nuestro enfoque radica en valorar la calidad de un generador mediante una medida única y global de congruencia entre las propiedades teóricas del modelo y las producidas por el generador, en lugar de aplicar un conjunto de pruebas de aleatoriedad aisladas que se basan en desviaciones particulares. Esta idea es aplicada, a manera de ilustración, para evaluar la calidad de algunos generadores con respecto a su capacidad de simular un proceso de decaimiento radiactivo. En la siguiente sección haremos un breve repaso de la ley del decaimiento radiactivo y de su simulación mediante el método de Monte Carlo; en la Sec. 3 presentamos los generadores que se probarán en el presente trabajo; en la Sec. 4 explicamos los criterios de nuestro método de prueba de generadores de números pseudoaleatorios y en la Sec. 5 ilustramos la aplicación de nuestro método. Finalmente, en la Sec. 6 se hace una discusión de los resultados obtenidos.

2. La Ley del decaimiento radiactivo

Por razones de autocontención y con el fin de establecer la notación usada en este trabajo, revisemos muy brevemente [35] la ley de decaimiento radiactivo:

Considérese al tiempo $t=0$, una muestra grande con $N(0)$ partículas inestables y supongamos que cada partícula tiene una probabilidad $\lambda\Delta t$ de decaer durante el intervalo de tiempo pequeño Δt . Aquí λ es la constante llamada razón de decaimiento. Si tenemos un número $N(t)$ de partículas sin decaer al tiempo t , entonces $\lambda N\Delta t$ partículas decaerán durante el intervalo de tiempo $[t, t+\Delta t]$. Claramente, esto decrecerá

el número de partículas que aún no decaen en

$$\Delta N = -\lambda N \Delta t. \quad (1)$$

Cuando el intervalo de observación en el tiempo Δt tiende a cero, podemos integrar la ecuación diferencial resultante obteniendo la ley de decaimiento exponencial:

$$N(t) = N(0)e^{-\lambda t}. \quad (2)$$

Las unidades de λ son tiempo⁻¹.

Se define la semi-vida como el intervalo de tiempo en el cual la mitad de la muestra inicial habrá decaído:

$$N(T_{1/2}) = N(0)/2, \quad (3)$$

es decir,

$$T_{1/2} = \frac{\ln 2}{\lambda} = \frac{0.693}{\lambda}. \quad (4)$$

Finalmente, otra definición importante a considerar es la de la vida media τ . Esta se define como el recíproco de la razón de decaimiento:

$$\tau = \frac{1}{\lambda}. \quad (5)$$

2.1. Algoritmo para simular la ley de decaimiento radiactivo

El algoritmo utilizado para llevar a cabo la simulación del decaimiento radiactivo es el siguiente:

1. Se eligen los valores iniciales del decaimiento (cantidad de partículas iniciales, valor de λ , el intervalo de tiempo que se simulará $[0, T]$) y el tamaño del incremento en el tiempo Δt .
2. Para cada valor discreto del tiempo $t_i \in [0, T]$, $i=1, 2, \dots, n$; considérense todas las $N^*(t_i)$ partículas que no han decaído hasta ese instante. Por medio del generador de números aleatorios bajo prueba, genérese un número pseudoaleatorio.
3. Si el número pseudoaleatorio generado en el paso anterior es menor que λ , la partícula decae, y quedarán $N^*(t_{i+1}) = N^*(t_i) - 1$ partículas al tiempo t_{i+1} .
4. Repítase el procedimiento a partir del paso 2 para todas las demás $N^*(t_i) - 1$ partículas y para cada t_i en el intervalo especificado. Termínese el procedimiento cuando el número de partículas restantes sea menor que 10. (Esto con el fin de evitar fluctuaciones estadísticas demasiado grandes).

El símbolo * es utilizado de aquí en adelante para indicar que los anteriores valores son los valores producidos por la simulación y no por la ley (2). En la Fig. 1 se muestra el resultado de realizar esta simulación 10 000 veces, con razón de decaimiento $\lambda = 0.035$ y con un número inicial de partículas de $N_0 = 1000$.

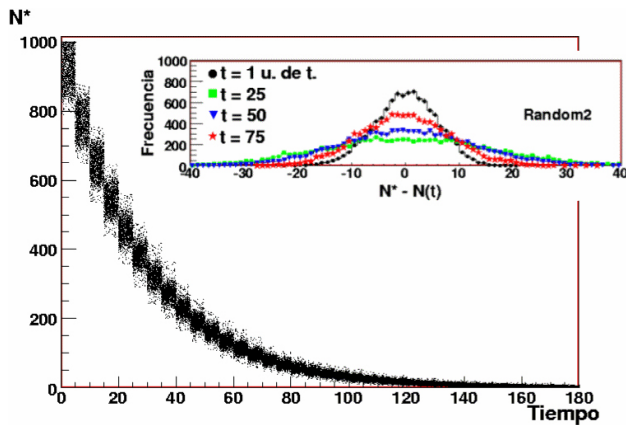


FIGURA 1. 10 000 simulaciones del decaimiento radioactivo con $\lambda = 0.035$, $N_0 = 1000$ partículas y $\Delta t = 0.001$; la figura contiene 2×10^9 puntos y se obtuvo usando el generador Random3 descrito en la siguiente sección. El panel superior muestra la distribución de $N^* - N(t)$ para 1, 25, 50 y 75 u. de tiempo. Para un buen generador y como consecuencia del teorema central del límite, estos valores se distribuirán gaussianamente, lo cual es uno de los criterios de nuestra prueba, como se explicará más adelante en la Sec. 4.

3. Generadores

En este trabajo, y para ilustrar nuestro método, se utilizaron los generadores ran0, ran48 y los generadores implementados en las clases basadas en C++ TRandom, TRandom2 y TRandom3 y que son específicos del ambiente ROOT [36], de amplia utilización en física de altas energías. Por simplicidad los llamaremos Random1, Random2 y Random3 y los explicaremos a continuación:

- ran48: Pertenece a la familia de funciones rand48() generadoras de números aleatorios del lenguaje C estándar y es un generador MLCG (Multiplicative Linear Congruential), es decir, usa la relación recursiva $r_i = (\alpha + c) \text{mod } m$, para $i = 1, 2, 3, \dots$. Donde r_0, α, m y c son números enteros de 48 bits con los siguientes valores: $\alpha = 25214903917$, $c = 11$ y $m = 2^{48}$.
- ran0: Es un generador cuya implementación actualizada puede encontrarse en la Ref. 15 y que fue propuesto por Park & Miller [37, 38] como un modelo de generador con calidad mínima. La implementación usada de este algoritmo genera números pseudoaleatorios con distribución uniforme entre 0 y 1. Se trata de un algoritmo lineal congruencial, donde, siguiendo la notación de la entrada anterior, se hace $\alpha = 7^5 = 16807$, $c = 0$ y $m = 2^{31} - 1 = 2147483647$.
- Random1: La clase TRandom1 implementa la función Rndm(), que es una traducción y actualización a C++ de la función Fortran generadora de pseudoaleatorios RNDM, la cual pertenece a las famosas bibliotecas científicas CERNLIB [39]. Random1 genera números pseudoaleatorios mediante el método de congruencias

lineales, con una distribución uniforme en el intervalo $[0, 1]$ con un periodo de 10^8 . Los parámetros usados por esta función son $\alpha = 5^{15} = 30517578125$, $c = 0$ y $m = 2^{47} = 140737488355328$.

- Random2: Implementa un generador de números pseudoaleatorios basado en el método de Tausworthe máximamente equidistribuido por L'Ecuyer y usa 3 palabras por estado. Toda la información sobre este generador se puede consultar en las Refs. 40 y 41. El periodo de Random2 es $2^{88} = 309485009821345068724781056$.
- El generador Random3, se basa en el método "Mersenne Twister" en la implementación de M. Matsumoto y T. Nishimura [42]. Random3 es un generador 623-dimensional que produce números equiprobables en el intervalo $[0, 1]$. Su gran ventaja es su periodo largo ($2^{19937} - 1$), aunque su desventaja es su estado interno relativamente grande de 624 enteros.

Los códigos fuente de los tres últimos generadores se pueden consultar en la página de ROOT [36].

3.1. Breve estudio preliminar de la calidad de los generadores usados en este trabajo

Con el objetivo de hacer una comparación independiente de nuestra prueba, la Tabla I muestra los resultados de aplicar 5 sencillas pruebas de aleatoriedad a los generadores usados en nuestro estudio. Estos se realizaron sobre una pequeña muestra de 30 000 eventos para cada generador y consistieron en:

- Cálculo de μ , el valor esperado promedio. Óptimo valor $\mu = 0.5$.
- Cálculo de la desviación estandar. Óptimo valor:

$$\sigma = \sqrt{\frac{7}{12}} \simeq 0.28861.$$

- Prueba de Anderson-Darling [43]. El mejor generador es aquel que minimize al estadístico A^2 , el cual nos da la distancia entre la función de distribución empírica y la teórica, en este caso la uniforme en $(0, 1)$.
- Estimación de $\pi = 3.14159 \dots$
- Cálculo del coeficiente de correlación de Pearson C_p . Óptimo valor $C_p = 0$.
- Compresión de los archivos de las muestras usando el algoritmo de Lempel & Ziv [44] y comparando el grado de compresión con una muestra del mismo tamaño de números verdaderamente aleatorios obtenidas de la medición del ruido atmosférico [45]. Óptimo (ideal) valor 0 %.

TABLA I. Algunas sencillas pruebas de calidad realizadas a una muestra de cada generador de 30 000 números aleatorios uniformemente distribuidos en el intervalo (0,1). Cuadro superior: Test de uniformidad de los generadores. Cuadro inferior: Cálculo de π mediante el método de Monte Carlo, coeficiente de autocorrelación y compresibilidad de las muestras generadas. Los parámetros se explican en el final de la presente sección.

Generador	$\mu \pm 0.002$	$\sigma \pm 0.0012$	A^2
Random1	0.500	0.2885	1.145
Random2	0.502	0.2889	0.280
Random3	0.496	0.2891	0.523
drand48	0.502	0.2877	0.859
Ran0	0.500	0.2885	1.048
Generador	π	C_p	Compresión
Random1	3.1443	0.0019	1 %
Random2	3.1404	0.0003	1 %
Random3	3.1364	0.0011	1 %
drand48	3.1308	0.0018	1 %
Ran0	3.1343	0.0089	1 %

4. Criterios de calidad

Como se indicó en la Sec. 1, el criterio básico aquí utilizado para valorar la calidad de un generador en el contexto de un modelo dado, se relaciona directamente con la capacidad del generador para reproducirlo y es en ese sentido que presentamos una medida estadística de congruencia, fundamentada en las propiedades teóricas del proceso bajo simulación. En nuestro caso, consideraremos tres características:

1. Normalidad conjunta de los valores simulados.
2. Convergencia de los promedios de los valores simulados a sus promedios teóricos (convergencia de primeros momentos).
3. Convergencia de las varianzas y covarianzas muestrales a sus valores teóricos (convergencia de segundos momentos).

Bajo un proceso de decaimiento exponencial como el que aquí se ha referido, puede verificarse que el valor esperado y la varianza de $N^*(t_i)$, están dados por

$$\mu(t_i) = E[N^*(t_i)] = N(0) \exp(-\lambda t_i),$$

$$\sigma^2(t_i) = V[N^*(t_i)] = N(0)\lambda \sum_{j=1}^{t_i} \exp[-\lambda(t_i + j - 1)],$$

Además, la covarianza entre las dos variables aleatorias $N^*(t_i)$ y $N^*(t_j)$ se expresa mediante

$$\sigma(t_i, t_j) = (1 - \lambda)^{t_j - t_i} \sigma^2(t_i),$$

donde $t_i \leq t_j$ y $h_{ij} = t_j - t_i, i, j = 1, 2, \dots, n$, con n igual a nuestro número de observaciones.

Para cada uno de los n instantes de tiempo t_1, \dots, t_n en el intervalo $[0, T]$, denotemos por $\mathbf{N}^* = [N^*(t_1), \dots, N^*(t_n)]'$ al vector aleatorio que representa los valores del proceso. El vector \mathbf{N}^* tiene entonces como valor esperado, al vector $\mu = [\mu(t_1), \dots, \mu(t_n)]'$ y por matriz de covarianzas $\Gamma = [\sigma(t_i, t_j)]$.

Bajo el supuesto de normalidad conjunta, la cantidad

$$D = [\mathbf{N}^* - \mu]' \Gamma^{-1} [\mathbf{N}^* - \mu],$$

tiene una distribución χ^2 con n grados de libertad.

Dadas m realizaciones del proceso, denotemos por $\bar{\mathbf{N}}^*$ al vector de promedios

$$\bar{\mathbf{N}}^* = [\bar{N}^*(t_1), \dots, \bar{N}^*(t_n)]',$$

donde $\bar{N}^*(t_i) = \frac{1}{m} \sum_{j=1}^m N_j^*(t_i)$ y $N_j^*(t_i)$ denota el j -ésimo valor simulado en el punto t_i .

Independientemente del supuesto de normalidad del vector \mathbf{N}^* , por aplicación del teorema central del límite, el vector $\sqrt{m}(\bar{\mathbf{N}}^* - \mu)$ converge en distribución a la de una variable aleatoria normal n -variada con vector de medias $\mathbf{0}$ y matriz de covarianzas Γ ; de ahí que la cantidad

$$D_m = m(\bar{\mathbf{N}}^* - \mu)' \Gamma^{-1} (\bar{\mathbf{N}}^* - \mu),$$

tiene como distribución límite, una χ^2 con n grados de libertad.

D_m es una medida estadística del ajuste entre los valores teóricos de los parámetros de primero y segundo orden derivados del modelo, en este caso el de decaimiento exponencial, y sus contrapartes muestrales obtenidos a partir de la simulación. Esta medida es la que en este trabajo se propone como criterio cuantitativo para la comparación de los generadores, ya que su valor se incrementa en presencia de desviaciones en los valores teóricos de los parámetros o bien en presencia de desviaciones de su estructura teórica de covarianzas. En el contexto de nuestro modelo, el mejor generador será aquel que produzca el menor valor de D_m . Cabe señalar que, con este enfoque, la(s) causa(s) específica(s) de las desviaciones que pudieran presentarse debidas a deficiencias del generador (como correlaciones, ciclos o cortos períodos, etc) nos resultan irrelevantes en tanto que nuestro interés radica en detectar su efecto final, observado como incongruencia estadística de las simulaciones con respecto a las propiedades del proceso que se simula.

5. Implementación del método

Ilustremos ahora el procedimiento anteriormente descrito, considerando un proceso con los siguientes parámetros: $N(0) = 1000$ partículas iniciales, constante de decaimiento $\lambda=0.035$, $\Delta t = 0.0025$ y $m = 2000$ simulaciones para cada generador. Los $n = 6$ instantes observados en el intervalo

[0, 250] fueron $t_1 = 1, t_2 = 25, t_3 = 50, t_4 = 75, t_5 = 100$ y $t_6 = 125$.

La Tabla II muestra los valores calculados de D_m para cada uno de los generadores y sus valores de probabilidad (p) asociados, con base en la distribución χ^2 con 6 grados de libertad.

Para estos valores, el vector de medias y la matriz de covarianzas teóricos del modelo son

$$\mu = [965.605 \quad 416.862 \quad 173.774 \quad 72.440 \quad 30.197 \quad 12.588],$$

$$\Gamma = \begin{bmatrix} 33.796 & 14.372 & 5.898 & 2.420 & 0.9933 & 0.4076 \\ & 247.367 & 101.514 & 41.659 & 17.096 & 7.016 \\ & & 146.104 & 59.958 & 24.605 & 10.097 \\ & & & 68.375 & 28.060 & 11.515 \\ & & & & 29.801 & 12.230 \\ & & & & & 12.649 \end{bmatrix}$$

El menor valor de D_m corresponde al calculado para el generador Random3, por lo que concluimos que éste reproduce el modelo de decaimiento radiactivo con mayor precisión. Aunque el generador Random2 presentó un valor mayor de D_m , el valor de probabilidad asociado nos indica que aproximadamente un 85 % de las veces en que se simule este proceso bajo las mismas condiciones, podríamos esperar un valor mayor o igual a 2.634, por lo que el valor obtenido es suficientemente pequeño para concluir que la simulación es muy aceptable. Por el contrario, Ran0 y Random1 son los peores generadores según nuestro criterio, el resultado producido por el generador Random1, nos indica que la probabilidad de obtener un valor tan grande como $D_m=14.500$ es del orden de 1 en 40, significando con esto que un valor tan improbable nos lleva necesariamente a concluir que este generador no simula satisfactoriamente nuestro proceso. Ran0 es aún peor con un valor de D_m casi un orden de magnitud más grande que el de Random1 y con probabilidad cero.

Las Figs. 2 a 6 muestran los histogramas de los valores D bajo las mismas condiciones de simulación para cada uno de los generadores bajo estudio, recordando que para cada generador, éstos corresponden a 2000 simulaciones del decaimiento radiactivo con $\lambda = 0.035$, un número de 1000 partículas iniciales e intervalo de tiempo de $\Delta t = 0.0025$ unidades de tiempo. En ellos se aprecia una congruencia del valor obtenido de D_m con el ajuste $\chi^2_{(6)}$ ilustrado por la línea continua en cada una de ellas.

TABLA II. Valores calculados de la estadística D_m para cada uno de los generadores examinados.

Generador	D_m	Probabilidad
drand48	2.914	0.820
ran0	132.79	0.000
Random1	14.500	0.025
Random2	2.634	0.850
Random3	1.223	0.976

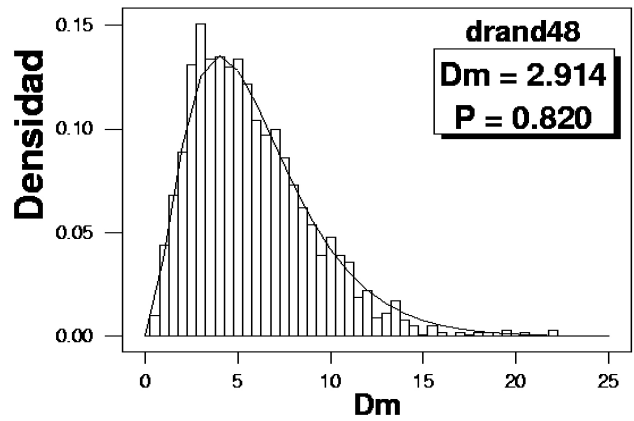


FIGURA 2. Histograma de los valores de D para drand48. Se muestra el resultado de 2000 simulaciones. La línea sólida representa la densidad $\chi^2_{(6)}$, que teóricamente tiene D . En esta prueba obtenemos los valores $D_m = 2.914$ con probabilidad $p = 0.820$.

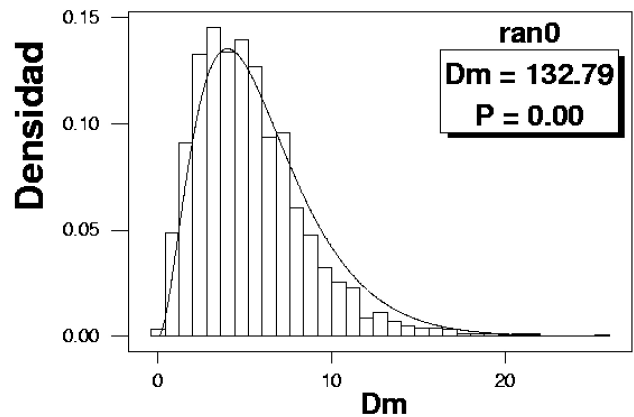


FIGURA 3. Histograma de D para ran0. La línea sólida representa la densidad $\chi^2_{(6)}$, que teóricamente tiene D . Note que el ajuste no parece satisfactorio, lo cual es claro de los valores obtenidos de $D_m = 132.79$ y probabilidad $p = 0.0$ en esta prueba.

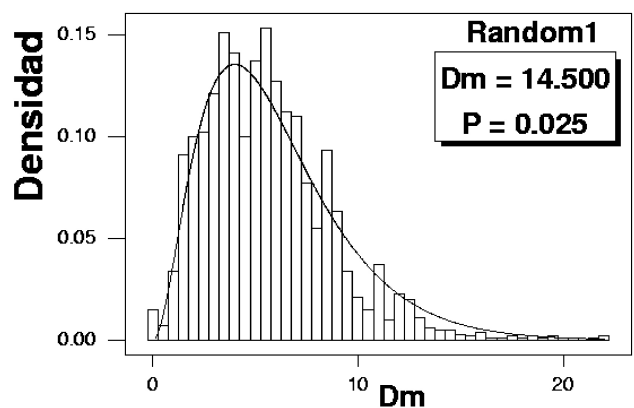


FIGURA 4. Histograma de D para el generador Random1. Como antes, la línea sólida representa la densidad $\chi^2_{(6)}$ teórica de D . Note que el ajuste no es satisfactorio, lo cual puede verse de los valores obtenidos en esta prueba, los cuales son $D_m = 14.500$ y una probabilidad $p = 0.025$.

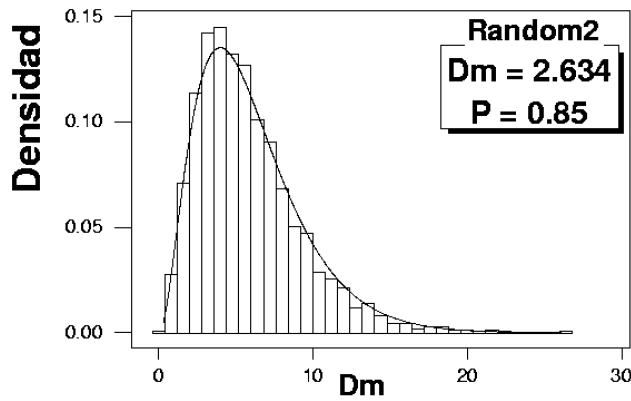


FIGURA 5. Histograma de D para el el generador Random2 y su comparación con la línea sólida que representa la densidad de $\chi^2_{(6)}$, que teóricamente tiene D ; en este caso, el ajuste parece adecuado y obtuvimos los valores de $D_m = 2.634$ y $p = 0.85$.

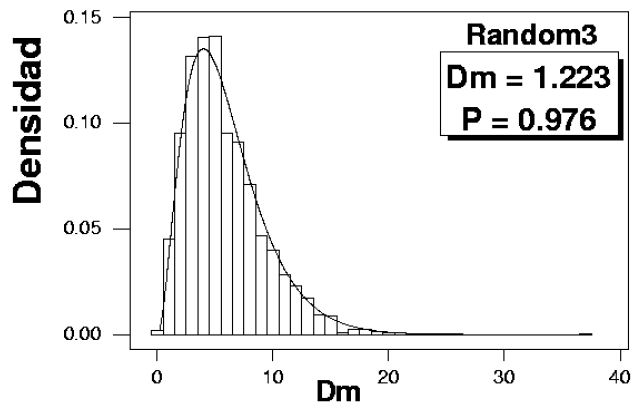


FIGURA 6. Histograma de D correspondiente al generador Random3, con $\lambda = 0.035$ y $n = 6$, que teóricamente tiene D ; para este generador, el ajuste de $\chi^2_{(6)}$ es el mejor de todos, lo cual se ve numéricamente de los valores $D_m = 1.223$ y $p = 0.976$.

Es importante destacar que, por el teorema central del límite, la distribución de D_m es poco sensible a desviaciones de la normalidad de los vectores $\mathbf{N}^*(t)$ y consecuentemente a la distribución original de D ; sin embargo, a partir de nuestros resultados, podría percibirse incorrectamente que D_m es sensible a desviaciones en la distribución teórica de los valores D , por lo que es necesario aclarar que la congruencia observada entre nuestra medida de ajuste D_m y las variaciones en la distribución de D con respecto a la distribución $\chi^2_{(6)}$, como la que se aprecia en la Fig. 4 son consecuencia de diferencias entre los valores teóricos de los parámetros y los producidos por las simulaciones, mismas que se ven reflejadas como irregularidades en el histograma respectivo. Es, en este sentido específico, que es posible concluir que D_m mide indirectamente esas irregularidades en la distribución de D , condensando el comportamiento general en una sola cantidad.

5.1. Ventajas y desventajas del método propuesto y su corrección

Entre las ventajas que posee nuestra prueba, mencionaremos las siguientes:

1. Aun si la simulación del decaimiento radioactivo tiene sus puntos finos (véase Ref. 46, p. 43), nuestro método es de fácil implementación.
2. Aunque no presentamos un estudio al respecto; por la construcción de nuestra prueba, ésta debe de ser sensible a los generadores que producen series de números aleatorios con correlaciones entre sí. Esto es debido a que tales correlaciones afectarían la frecuencia relativa del decaimiento de las partículas debido a la dependencia de los decaimientos introducida por las correlaciones.

Ahora bien, es claro que no existe un generador de números aleatorios perfecto; un generador que sirve para simular correctamente algún proceso, puede pasar cierto tipo de pruebas y no pasar otras; esto es ilustrado observando en la Tabla I que aunque Random2 parece ser un ligeramente mejor generador que Random3, Random3 es un mejor generador para simular el decaimiento radioactivo que Random2. Por lo dicho anteriormente, se ha llegado a sugerir que el mejor generador de números aleatorios es aquel que simula mejor el proceso que se quiere estudiar.

Sobre las desventajas que presenta nuestra prueba, a continuación mencionaremos sus dos principales debilidades y la forma de corregirlas:

1. La primera desventaja de nuestro método es consecuencia del carácter binomial del decaimiento radioactivo, que ilustraremos para un caso particular: dada una razón de decaimiento λ , al realizar la simulación del decaimiento de un material radioactivo con N_o partículas al tiempo inicial, si el número aleatorio del generador a probar es menor que λ la partícula decaerá, en caso contrario no decaerá. Ahora bien, si la constante de decaimiento toma un valor, por decir de $\lambda = 0.5$, y si el generador fuera tan malo que no nos diera como salida números pseudoaleatorios dentro de cualquier conjunto $S \subset [0, 1]$ de la forma $S = (a, b) \cup (1 - b, 1 - a)$ con $0 < a < 0.5$, $0 < b < 0.5$ y $a < b$; es claro que este generador pasaría nuestra prueba de aleatoriedad. Nótese que este ejemplo se puede generalizar, en teoría, al caso de un mal generador que omite en su salida números en la unión de cualquier número de conjuntos disjuntos de la forma del conjunto S y para cualquier valor de λ , multiplicando a los valores dentro de los intervalos por correspondientes constantes de proporcionalidad que conserven la probabilidad del decaimiento de las partículas.
2. Para aplicar nuestro método, calculamos la matriz de covarianzas Γ del vector aleatorio \mathbf{N}^* , con dimensión

n , misma que se invierte para obtener el estadístico D_m . Puede verse que refinando la partición del intervalo temporal de nuestra simulación, es decir haciendo crecerⁱⁱ a n nos puede llevar a invertir una matriz de gran tamaño. Para darse una idea de lo que esto podría significar en cuanto a tiempo de CPU, realizamos un pequeño estudio y calculamosⁱⁱⁱ que el tiempo de máquina necesario para invertir matrices hasta de tamaño 100×100 es despreciable, pero de allí en adelante la complejidad de los cálculos y, por tanto el tiempo de máquina, crece grandemente según la Tabla III.

Sobre la primera desventaja mencionada, es claro que cuando se desea probar un generador de números aleatorios, no podemos conformarnos con que pase una sola prueba, sino que debe pasar consistentemente gran cantidad de pruebas. Por lo tanto, podemos decir que nuestra prueba se debería de combinar con pruebas de uniformidad para complementar su confiabilidad. Otra manera de corregir este problema sería haciendo variar λ en incrementos de tamaño adecuado y realizar nuestra prueba para los distintos valores de λ . Aunque esta forma de corrección de nuestra prueba requeriría por supuesto de un gran incremento de la complejidad de los cálculos a realizar. Sobre esto último, y en relación con la segunda desventaja de nuestro método, debemos mencionar que la generación, así como la prueba de números aleatorios, es tanto un arte como una ciencia y, en este caso, la selección del número de instantes de tiempo a muestrear es obvia y se aplica a la simulación de cualquier proceso: debe de hacerse razonablemente de tal forma que no se requiera tanta CPU para hacer inviable la prueba. En este artículo hemos utilizado exitosamente 6 instantes de tiempo para ilustrar nuestra prueba.

6. Conclusiones

En este trabajo se propone y se fundamenta teóricamente una prueba empírica de generadores de números pseudoaleatorios basada en el proceso estocástico de decaimiento exponencial. Específicamente hacemos uso de la distribución del número de partículas sobrevivientes y su normalidad en concordancia con el teorema central del límite. Nuestra prueba, tal como debiera esperarse para este tipo de criterios de aleatoriedad,

TABLA III.

Tamaño	Tiempo de CPU (s)
100 × 100	0.01
200 × 200	0.06
400 × 400	1.12
800 × 800	12.20
1600 × 1600	206.31

posee la característica de ser sensible tanto a desviaciones distribucionales como a las de los parámetros del modelo usados en la simulación.

Se ilustra este método mediante su aplicación a cinco generadores conocidos: tres generadores congruenciales lineales con distintos parámetros, un Tausworthe y un Mersenne Twister, siendo el mejor de todos, según los criterios de nuestra prueba, este último. También se discutieron las desventajas de nuestro método y cómo corregirlas.

Nuestro método no resulta de difícil implementación, sólo hay que tener cuidado al realizar las simulaciones completas del proceso de decaimiento radioactivo, con las dificultades computacionales que esto implica, especialmente con lo que respecta al tamaño del intervalo de tiempo elegido (véase la Ref. 46, p. 53). Por otro lado, es posible, considerando la longitud del periodo de los generadores modernos, usar una muestra relativamente pequeña de sus valores. En este trabajo, para obtener la muestra a analizar con nuestro método, se hicieron 2000 simulaciones del decaimiento radioactivo para cada generador, esto es, en total se simularon 50 000 000 eventos y de estos seleccionamos 2000 valores para cada uno de los seis tiempos distintos (1,25,50,75,100 y 125 unidades recíprocas de λ). Es decir, en nuestra prueba y para cada generador, de los 50 000 000 eventos simulados, un total 12 000 eventos fueron analizados.

Agradecimientos

Agradecemos los comentarios hechos a este trabajo por N. Cruz, S. Jiménez-Castillo y M. Rodríguez-Achach, también agradecemos al Consejo Nacional de Ciencia y Tecnología (Conacyt) el apoyo brindado bajo el proyecto No SEP-2003-C02-44598, finalmente quisieramos agradecer las útiles sugerencias y comentarios del referee anónimo que llevaron grandemente a mejorar la calidad del presente trabajo.

ⁱ En la práctica, frecuentemente se requiere usar muchas veces una misma sucesión de números aleatorios, lo cual implica tener que almacenar la sucesión completa con todos los inconvenientes que esto acarrea. Es en este sentido que usamos el término reproducibilidad.

ⁱⁱ Aunque claramente no tiene ningún caso, n se puede hacer crecer hasta su límite práctico de $n = T/\Delta t$, el cual es un número

muy grande (en nuestro caso y con la simulación realizada en el presente trabajo, implicaría trabajar con un vector de dimensión $n = 10^5$ y por lo tanto el invertir una matriz de tamaño $10^5 \times 10^5$.

ⁱⁱⁱ Utilizamos para realizar este cálculo una vez más ROOT que incluye un intérprete C++ en una máquina Pentium 4 a 1.8 GHz.

1. N. Metropolis y S. Ulam, *J. Amer. Statist.* **44** (1949) 335.
2. N. Metropolis, A. Rosenbluth, A.H. Teller y E. Teller, *J. Chem. Phys.* **21** (1953) 1087.
3. B. Ya, Ryakbo, V.S. Stognienko y Yu.I. Shokin, *Journal of statistical planning and inference* **123** (2004) 365.
4. M. Blum y M. Shub, *SIAM J. Computing* **13** (4) (1984) 850.
5. J. Bossert, M. Feindt y U. Kerzel, *Nuclear Instruments and Methods in Physics Research Section A* **559** (2006) 232.
6. B. Vilenkin, *Ecological Modelling* **195** (2006) 385.
7. L. Nanni y A. Lumini, *Neurocomputing* **69** (2006) 858.
8. D. Maio y L. Nanni, *Neurocomputing* **69** (2005) 242.
9. Navindra Persaud, *Medical Hypotheses* **65** (2005) 211.
10. D.Davis, R. Ihaka y P. Fenstermacher, Cryptographic Randomness from air turbulence in disk airs. Proceedings of Crypto 94, Springer Verlag Lecture Notes in Computer Science, No. 839, (1994).
11. Nicola Bruti-Liberatía, Filippo Martini, Massimo Piccardi y Eckhard Platena. A hardware generator of multi-point distributed random numbers next term for Monte Carlo simulation. Mathematics and Computers in Simulation, in Press (2007).
12. Salih Ergün y Serdar Ozogdcu, *International Journal of Electronics and Communications* **61** (2007) 235.
13. J. Szczepanski *et al.*, *Computers & Security* **23** (2004) 77.
14. D.E. Knuth, *The Art of Computer Programming*, Second edition (Addison-Wesley Professional, (1998).
15. Numerical Recipes in C: The Art of Scientific Computing by William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, Second edition (Cambridge University Press, 1992).
16. S. Wolfram, *Advances in Applied Mathematics* **7** (1986) 123.
17. Sheng-Uei Guan y Shu Zhang, *FGCS* **20** (2004) 627.
18. Syn Kiat Tan, *International Journal of Modern Physics C* **18** (2007) 217.
19. Lap-Piu Lee y Kwok-Wo Wong, *An International Journal of Computers & Mathematics with applications* **47** (2004) 217.
20. J.R. Heringa, H.W. Blote y A. Compagner, *J. Computat. Phys.* **10** (1983) 250.
21. A.M. Ferrenberg, L.D. Landau y Y.J. Wong, *Phys. Rev. Lett.* **69** (1992) 3382.
22. P. Hellekalek, *Mathematics and Computers in Simulation* **46** (1998) 485.
23. B.Ya. Ryakbo y V.A. Monarev, *Journal of Statistical Planning and Inference* **133** (2005) 95.
24. Makoto Matsumoto y Takuji Nishimura, *Mathematics and computers in simulation* **62** (2003) 431.
25. Nezh C. Geclinli y Murat A. Apohan, *Signal Processing* **81** (2001) 1389.
26. Hui-Chin Tan, *European Journal of Operational Research* **182** (2007) 820.
27. Stefan Wegenkittl, *Mathematics and Computers in Simulation* **55** 281-288 (2001).
28. I. Stauffer, *International Journal of Modern Physics C* **5** (1999) 807.
29. Mihyun Kang, *Journal of Computational and Applied Mathematics* **174** (2005) 1.
30. P. L'Ecuyer, *Entropy Tests for Random Number Generators GERAD report* 1996.
31. A.L. Rukhin, *J. Appl. Probab.* **37** (2000) 88.
32. I. Vattulainen, T. Ala-Nissila y K. Kankaala, *Physical Review E* **52** (1995) 3205.
33. I. Vattulainen, K. Kankaala y T. Ala-Nissila, *Physical Review Letters* **73** (1994) 2513.
34. I. Vattulainen, *New tests of random numbers for simulations in physical systems*. Tesis, Universidad de Helsinki, Helsinki, Finlandia, (1994).
35. E. Fermi, *Nuclear Physics* (University of Chicago Press. Chicago USA, 1949).
36. Rene Brun y Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996; *Nucl. Inst. Meth. in Phys. Res. A* **389** (1997) 81; Ver también <http://root.cern.ch/>
37. S. Park y K. Miller, *Comm. ACM* **31** (1988) 1192.
38. S. Park y K. Miller, *Comm. ACM* **36** (1993) 105.
39. CERN Program Library. CERNLIB. Short Writeups. Application Software and Databases. Computing and Network Division. Edition june 1996. CERN Geneva, Switzerland.
40. P. L'Ecuyer, *Mathematics of Computation* **65** (1996) 213.
41. P. L'Ecuyer, *Mathematics of Computation* **68** (1999) 225.
42. M. Matsumoto y T. Nishimura, *ACM Transactions on Modeling and Computer Simulations: Special Issue on Uniform Random Number Generators* **8** (1998) 3.
43. R.B. D'Agostino y M.A. Stephens, *Goodness of fit Techniques Marcel Dekker* (New York, 1986).
44. J. Ziv y A. Lempel, *IEEE Transactions on Information Theory* **23** (1977) 337.
45. La muestra fue descargada del sitio www.random.org
46. D.P. Landau y K. Binder, *A guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, 2000).