

FUNDAMENTOS DE MONTE CARLO - I

CURSO ESFM-IPN

Arturo Delfín Loya

<https://arturodelfinloya.github.io/MCNP>

Agosto, 2016



instituto nacional de
investigaciones nucleares



***** Simulación de transporte de partículas

***** Muestreo aleatorio

”La simulación es mejor que la realidad”

Richard W. Hamming, 1991

Von Neumann inventó la computación científica en la década de los 1940s.

- ▶ Almacenamiento de programas, "software", algoritmos y diagramas de flujo
- ▶ Computadoras "ordinarias" hoy son llamados "máquinas de Von Neumann" (no en México)

Von Neumann inventó métodos de Monte Carlo para el transporte de partículas en la década de 1940 (con Ulam, Fermi, Metropolis, y otros en el LANL).

- ▶ Primer publicación sobre el transporte de partículas de Monte Carlo: J. von Neumann, carta a R. D. Richtmyer, en "Métodos Estadísticos de difusión neutrones" informe de LANL LAMS-551 (1947).
- ▶ Alta precisión - no hay aproximaciones esenciales
- ▶ Caro - "método como último recurso" bastante tiempo de CPU

Hoy en día, con las computadoras de alta velocidad y clusters Linux, Monte Carlo es el método de elección para muchos cálculos.

- ▶ Hoy se emplean hasta PC's = Ayer eran los superordenadores
- ▶ Modelado preciso de la geometría y la física

Ecuación lineal de Transporte de Boltzmann - independiente del tiempo

$$\Psi(\vec{r}, \vec{v}) = \int \left[\int \Psi(\vec{r}', \vec{v}') C(\vec{v}' \rightarrow \vec{v}, \vec{r}') d\vec{v}' + Q(\vec{r}', \vec{v}) \right] T(\vec{r}' \rightarrow \vec{r}, \vec{v}) d\vec{r}' \quad (1)$$

donde:

- ▶ $\Psi(\vec{r}', \vec{v}')$ = densidad de colisión de partículas
- ▶ $C(\vec{v}' \rightarrow \vec{v}, \vec{r}') =$ Kernel de colisión, cambio de velocidad en una posición constante
- ▶ $Q(\vec{r}', \vec{v}) =$ Término fuente
- ▶ $T(\vec{r}' \rightarrow \vec{r}, \vec{v}) =$ Kernel de transporte, cambio de posición en una velocidad constante

Flujo angular

$$\psi(\vec{r}, \vec{v}) = \frac{\Psi(\vec{r}, \vec{v})}{\Sigma(\vec{r}, |\vec{v}|)} \quad (2)$$

Flujo escalar

$$\Phi(\vec{r}, |\vec{v}|) = \int_{\hat{\Omega}} \frac{\Psi(\vec{r}, \vec{v})}{\Sigma(\vec{r}, |\vec{v}|)} d\hat{\Omega} \quad \vec{v} = |\vec{v}| \hat{\Omega} \quad (3)$$

Término fuente para la ecuación de Boltzmann

$$Q(\vec{r}', \vec{v}) =$$

- ▶ $S(\vec{r}, \vec{v}) =$ \Leftarrow Fuente fija
- ▶ $S(\vec{r}, \vec{v}) = + \int \Psi(\vec{r}, \vec{v}') F(\vec{v}' \rightarrow \vec{v}, \vec{r}) d\vec{v}'$ \Leftarrow Fuente fija + Fisión
- ▶ $\frac{1}{K} \int \Psi(\vec{r}, \vec{v}') F(\vec{v}' \rightarrow \vec{v}, \vec{r}) d\vec{v}'$ \Leftarrow Eigenvalor

donde:

- ▶ $S(\vec{r}, \vec{v}) =$ Fuente fija
- ▶ $F(\vec{v}' \rightarrow \vec{v}, \vec{r}) =$ Operador de creación (debido a fisión), partículas en (\vec{r}, \vec{v}') crea partículas en (\vec{r}, \vec{v})
- ▶ $K =$ Eigenvalor

$$\Psi(\vec{r}, \vec{v}) = \int \left[\int \Psi(\vec{r}', \vec{v}') C(\vec{v}' \rightarrow \vec{v}, \vec{r}') d\vec{v}' + Q(\vec{r}', \vec{v}) \right] T(\vec{r}' \rightarrow \vec{r}, \vec{v}) d\vec{r}' \quad (4)$$

Suposiciones

- ▶ medio estático y homogéneo
- ▶ Las propiedades de los materiales no se ven afectados por las reacciones de partículas
- ▶ No hay interacción de las partículas entre sí
- ▶ Cadenas de Markov - el resultado en cualquier etapa es independiente de todos los resultados previos (el próximo evento depende sólo del actual (\vec{r}, \vec{v}, E) , no de los eventos anteriores)
- ▶ Puede ignorarse los efectos cuánticos y de onda (\vec{v} no es demasiado pequeño)

⇒ Puede usar el principio de superposición

Por conveniencia: * independiente del tiempo, * efectos relativísticos despreciables, * las partículas viajan en línea recta durante el evento, etc., etc.

Sea $p = (\vec{r}, \vec{v})$ y $R(p' \rightarrow p) = C(\vec{v}' \rightarrow \vec{v}, \vec{r}').T(\vec{r}' \rightarrow \vec{r}, \vec{v})$

Desarrollando Ψ en componentes que tienen $0, 1, 2, \dots, k$ colisiones

$$\Psi(p) = \sum_{k=0}^{\infty} \Psi_k(p) \quad \text{con } \Psi_0(p) = \int Q(\vec{r}', \vec{v}) T(\vec{r}' \rightarrow \vec{r}, \vec{v}) d\vec{r}'$$

Por definición

$$\Psi_k(p) = \int \Psi_{k-1}(p').R(p' \rightarrow p) dp'$$

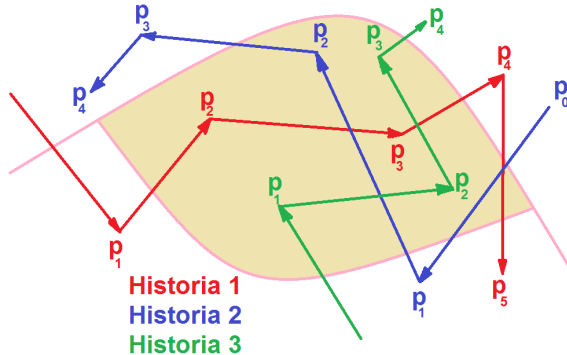
Note que la colisión k depende únicamente de los resultados de la colisión $k-1$ y no de colisiones anteriores $k-2, k-3, \dots$

Historias

Después de la repetida sustitución de Ψ_k

$$\Psi_k(p) = \int \Psi_{k-1}(p').R(p' \rightarrow p) dp' = \int \dots \int \Psi_0(p_0).R(p_0 \rightarrow p_1) \dots R(p_{k-1} \rightarrow p) dp_0 \dots dp_{k-1}$$

Una historia es una secuencia de estados $p_0, p_1, p_2, p_3, \dots$,



Para las evaluaciones en una región determinada, el **Tally** estima todos los eventos de cada colisión de cada "historia" dentro de cada región

$$\Psi_k(p) = \int \dots \int \Psi_0(p_0) \cdot R(p_0 \rightarrow p_1) R(p_1 \rightarrow p_2) \dots R(p_{k-1} \rightarrow p) dp_0 \dots dp_{k-1} \quad (5)$$

Aproximación de Monte Carlo

Genera una secuencia de estados $(P_0, p_1, p_2, p_3, \dots)$ [pe. una historia] por:

- ▶ Muestra aleatoria desde PDF para la fuente $\Psi_0(p_0)$
- ▶ Muestra aleatoria desde PDF para el k^{th} paso, $R(p_{k-1} \rightarrow p_k)$

Generar las evaluaciones de resultados promediando sobre los estados de M historias :

$$A = \int A(p) \cdot \Psi(p) dp \approx \frac{1}{M} \sum_{m=1}^M \left[\sum_{k=1}^{\infty} A(p_{k,m}) \right] \quad (6)$$

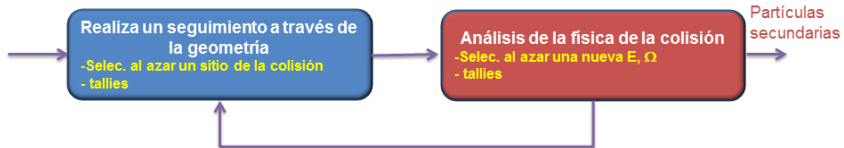
Aproximación en la simulación para el transporte de partículas:

Simula fielmente la historia de una sola partícula desde el nacimiento hasta la muerte.

Trayectoria aleatoria para una sola partícula

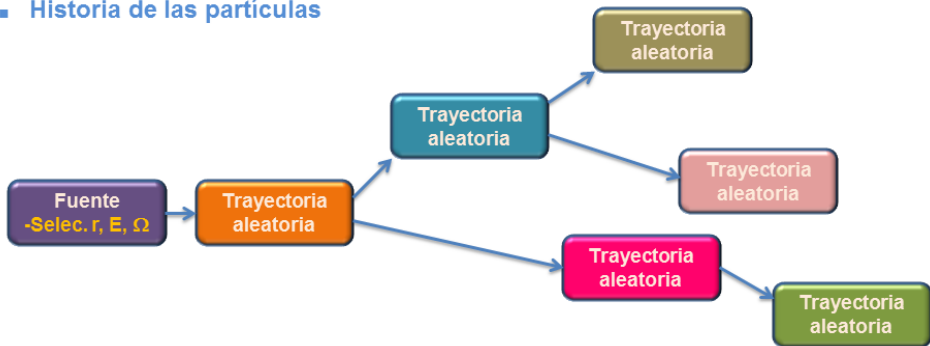
- ▶ Modelo de colisiones utilizando ecuaciones físicas y datos de secciones eficaces
- ▶ Modelo de trayectoria libre entre colisiones utilizando geometría computacional
- ▶ Las ocurrencias de eventos en cada región mediante Tally's
- ▶ Almacena datos de todas las partículas secundarias, para analizarlos en su momento

■ **Trayectoria aleatoria para partículas**



Una "Historia" es la simulación de la partícula original y de toda su descendencia

- Historia de las partículas



Se repiten para muchas historias, y se acumulan en los tallies

Regla fundamental: Piense como una partícula!!!

Monte Carlo con Fuente Fija (1)

T-02 Básico MCNP

13/45

Historia 1



Historia 2



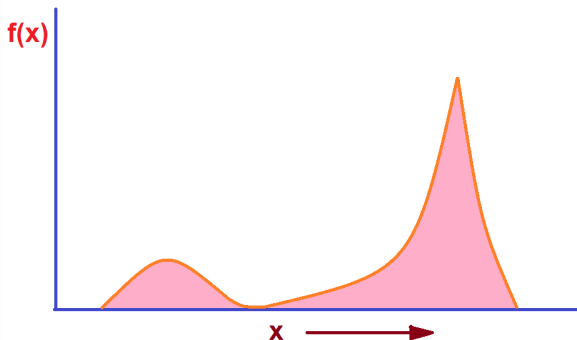
Historia 3



El éxito del Método de MC, es la noción del muestreo aleatorio

El problema puede ser establecido de la siguiente forma:

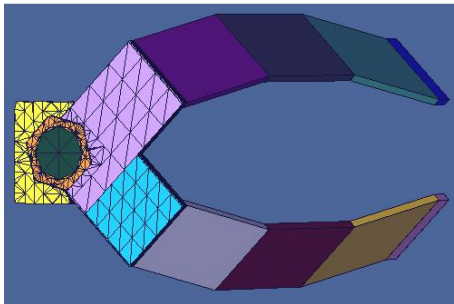
- Dada una densidad de probabilidad $f(x)$'s , se produce una secuencia de \hat{X} 's. Las \hat{X} 's deben ser distribuidas de la misma manera que $f(x)$



El uso del muestreo aleatorio distingue a MC de otros métodos

Cuando MC se emplea para resolver la integral de la ecuación de transporte de Boltzmann:

- ▶ Modelos de muestreo aleatorio, dan resultados de acontecimientos físicos.
(pe. colisión de neutrones, procesos de fisión, fuentes, ...)
- ▶ Modelos de geometría computacional de acuerdo a la disposición de materiales.



Fuente

- ▶ Muestreo aleatorio: E, Ω - analítica, discreta o tabulación espaciada PDF's
- ▶ Geometría computacional: r - muestreo desde una región en el espacio 3D, o discreta PDF

Seguimiento

- ▶ Muestreo aleatorio: $d_{colisión}$ - distancia de colisión, desde mfp y exponencial PDF
- ▶ Geometría computacional: d_{geom} - distancia a la frontera, trayectoria de rayos, siguiente región,...

Colisiones

- ▶ Muestreo aleatorio: E, Ω - analítica, discreta o tabulación espaciada PDF's
- ▶ Físicas: $\Sigma, f(\mu)$ - datos de secciones eficaces, angular PDF's, cinemática,...

Tallies: Estadística

Reducción de Varianza: Muestreo aleatorio

Códigos de Monte Carlo

Categoría de muestreo aleatorio

- ▶ Generador de número aleatorio \rightarrow PDF uniforme en $(0, 1)$
- ▶ Muestreo desde PDF's analíticos \rightarrow normal, exponencial, Maxwelliano, ...
- ▶ Muestreo desde PDF's tabulados \rightarrow PDF's angulares, espectros, ...

Desde códigos de Monte Carlo

- ▶ Números aleatorios $[\xi]$, son producidos por el generador RN en $(0, 1)$
- ▶ Variables aleatorias no uniformes, se producen a partir de las $[\xi's]$ por:
 - ▶ inversión directa
 - ▶ métodos de rechazo
 - ▶ Transformaciones
 - ▶ composición (mezclas)
 - ▶ sumas, productos, relaciones, ...
 - ▶ Búsqueda en una tabla + interpolación
 - ▶ otros trucos (!)

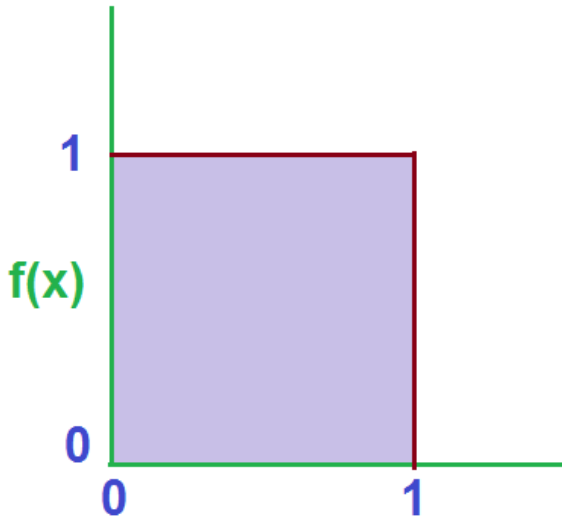
**Típicamente $<10\%$ del
tiempo total de CPU**

"La aleatoriedad es una propiedad negativa, es la ausencia de cualquier patrón." *Richard W. Hamming, 1991*

Los números no son aleatorios; una secuencia de números puede ser.
Secuencias aleatorias generalmente no son deseables en un ordenador.
RNG.

- ▶ Una función que genera una secuencia de números, que parecen haber sido por muestra al azar de una distribución uniforme en $(0, 1)$
- ▶ Repetible (determinista)
- ▶ Pasar las pruebas estadísticas de aleatoriedad
- ▶ En los códigos aparece como: $r = \text{ranf}()$
- ▶ También llamados "generadores de números pseudo-aleatorios"

Todos los muestreos aleatorios se realizan utilizando la declaración básica *RNG*



La mayoría de los códigos de nivel de producción de Monte Carlo para el transporte de partículas, utilizan generadores de números aleatorios de congruencia lineal:

$$S_{i+1} = S_i * g + c \mod 2^m \quad (7)$$

s_1 =semilla, g = multiplicador, c = sumador, 2^m = módulo

- ▶ Simple, rápido, robusto, más de 40 años de uso pesado
- ▶ La teoría es bien comprendida (pe., DE Knuth, Vol. 2, p.177)
- ▶ No es de los "mejores" generadores, pero lo suficientemente bueno - RN´s se utilizan de forma impredecible durante la simulación de partículas
- ▶ Para lograr la reproducibilidad del vector o del cálculo en paralelo, debe haber un método rápido y directo para saltar hacia delante (o hacia atrás) en la secuencia aleatoria

RNG Simple - Ejemplo No. 1

T-02 Básico MCNP

21/45

$$S_{k+1} \leftarrow [g * s_k + c] \bmod p \quad g = 47, c = 1, S_0 = 1, p = 100$$

s(0)	=	1
s(1)	=	(47 x 1 + 1) mod 100 = 48 mod 100 = 48
s(2)	=	(47 x 48 + 1) mod 100 = 2257 mod 100 = 57
s(3)	=	(47 x 57 + 1) mod 100 = 2680 mod 100 = 80
s(4)	=	(47 x 80 + 1) mod 100 = 3761 mod 100 = 61
s(5)	=	(47 x 61 + 1) mod 100 = 2868 mod 100 = 68
s(6)	=	(47 x 68 + 1) mod 100 = 3197 mod 100 = 97
s(7)	=	(47 x 97 + 1) mod 100 = 4560 mod 100 = 60
s(8)	=	(47 x 60 + 1) mod 100 = 2821 mod 100 = 21
s(9)	=	(47 x 21 + 1) mod 100 = 988 mod 100 = 88
s(10)	=	(47 x 88 + 1) mod 100 = 4137 mod 100 = 37
s(11)	=	(47 x 37 + 1) mod 100 = 1740 mod 100 = 40
s(12)	=	(47 x 40 + 1) mod 100 = 1881 mod 100 = 81
s(13)	=	(47 x 81 + 1) mod 100 = 3808 mod 100 = 8
s(14)	=	(47 x 8 + 1) mod 100 = 377 mod 100 = 77
s(15)	=	(47 x 77 + 1) mod 100 = 3620 mod 100 = 20
s(16)	=	(47 x 20 + 1) mod 100 = 941 mod 100 = 41
s(17)	=	(47 x 41 + 1) mod 100 = 1928 mod 100 = 28
s(18)	=	(47 x 28 + 1) mod 100 = 1317 mod 100 = 17
s(19)	=	(47 x 17 + 1) mod 100 = 800 mod 100 = 0
s(20)	=	(47 x 0 + 1) mod 100 = 1 mod 100 = 1
s(21)	=	(47 x 1 + 1) mod 100 = 48 mod 100 = 48
s(22)	=	(47 x 48 + 1) mod 100 = 2257 mod 100 = 57
etc . . .		

Método de congruencia multiplicativo - Lehmer

$$S_{i+1} = S_i * g + c \mod 2^m \quad 0 < S_i < 2^m$$

$$\xi_i = \frac{S_i}{2^m} \quad 0 < \xi < 1$$

Parámetros típicos

		2^m	Período	g	c
RACER	(KAPL)	2^{47}	2^{45}	84, 000, 335, 758, 957	0
RCP	(BAPL)	2^{48}	2^{48}	$2^9 + 1$	59, 482, 192, 516, 946
MORSE	(ORNL)	2^{47}	2^{45}	5^{15}	0
MCNP	(LANL)	2^{48}	2^{46}	5^{19}	0
VIM	(ANL)	2^{48}	2^{46}	5^{19}	0
RAND	(CRAY)	2^{48}	2^{46}	44, 485, 709,377, 909	0
—	(G. Marsaglia)	2^{32}	2^{32}	69069	1
MCNP5	(LANL)	2^{63}	2^{63}	Varios	1

Generador de congruencia lineal (LCG)

$$S_{n+1} = S_n * g + c \mod 2^m \quad (8)$$

- ▶ Están disponibles 7 LCG's - Se eligen en base a la prueba espectral, pruebas de Knuth y pruebas de Marsaglia DIEHARD
- ▶ LCG ($g, c, 2^m$):
 - ▶ MCNP Tradicional período = $2^{46} \approx 7 \times 10^{14}$
LCG ($5 \times 10^{19}, 0, 2^{48}$)
 - ▶ L'Ecuyer 63-bit LCG's mezclado período = $2^{63} \approx 9 \times 10^{18}$
LCG(9219741426499971445, 1, 2^{63}), [L'Ecuyer, Math. Comp., 68, 249-260 (1999)]
LCG(2806196910506780709, 1, 2^{63})
LCG(3249286849523012805, 1, 2^{63})
 - ▶ L'Ecuyer 63-bit LCG's multiplicativo período = $2^{61} \approx 2 \times 10^{18}$
LCG(3512401965023503517, 0, 2^{63}), [L'Ecuyer, Math. Comp., 68, 249-260 (1999)]
LCG(2444805353187672469, 0, 2^{63})
LCG(1987591058829310733, 0, 2^{63})

Naïve usa en varios códigos antiguos y códigos estudiantiles

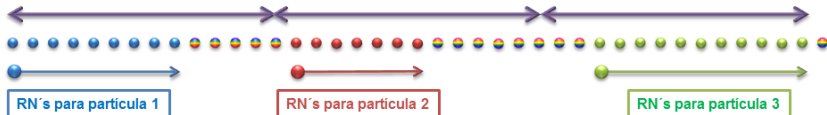


Problema:

- ▶ No se puede iniciar de partícula-2, hasta que termine la partícula-1, etc.
- ▶ No se puede realizar el procesamiento en paralelo de diferentes partículas

MCNP, VIM, RACER, y muchos otros códigos de producción

- ▶ Particionan la secuencia de RN en subsecuencias de igual longitud, una para cada partícula



- ▶ Puede procesar todas las partículas en paralelo
- ▶ La longitud de cada subsecuencia se llama paso "**stride**"
- ▶ Debe tener una forma rápida de **saltar** en la secuencia de RN

Historias vs Partículas

- ▶ Con la separación y/o creación de partículas secundarias, el número de partículas en una historia dada no se conoce de antemano
- ▶ Se necesita la partición en la secuencia RN por historia, no por partículas



Uso de RNG en Códigos de Transporte de Partículas MC (3)

T-02 Básico MCNP

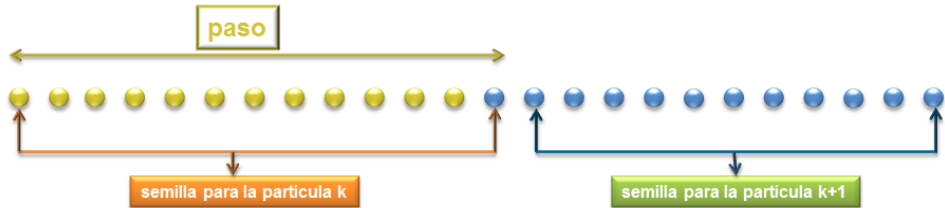
26/45



- ▶ Con este esquema, puede procesar historias en paralelo, pero no partículas en la misma historia
- ▶ Debe tener un esquema predecible por partículas banqueadas/no banqueadas, en una historia dada (por ejemplo, LIFO)

Reproducibilidad de la historia de una partícula

- ▶ Utiliza por separado una secuencia aleatoria para cada partícula
- ▶ Semillas de partida para las partículas separadas están separados por "pasos"

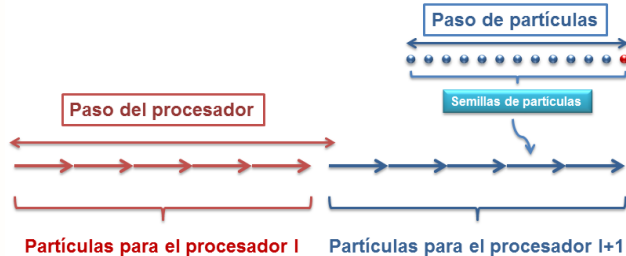


- ▶ El paso debe ser lo suficientemente grande, como para evitar el solapamiento
 - ▶ Reducción del desdoblamiento y la varianza que no sea necesaria para la física del núcleo
 - ▶ Reducción de uso total de números aleatorios

- ▶ 4,297 es el antiguo valor de default que usaba MCNP y VIM
- ▶ 152,917 es el valor por default usado por MCNP y VIM
 - ▶ preparado para lotes de reducción del desdoblamiento y la varianza
 - ▶ Potencial para lotes de partículas secundarias

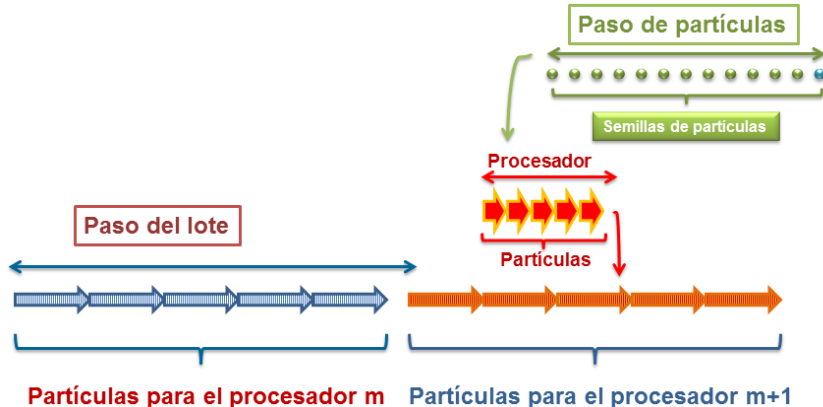
Procesamiento en paralelo

- ▶ Adquiere un súper paso en la secuencia aleatoria de partículas en cada procesador



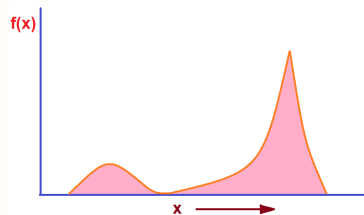
Problemas de eigenvalor

- ▶ Lotes de partículas, distribuidas entre procesadores paralelos
- ▶ Semillas para cada uno



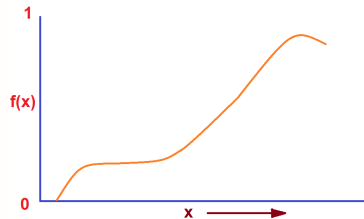
Función Densidad de Probabilidad (PDF)

- ▶ $f(x)$ función densidad de probabilidad (PDF)
 $f(x) \geq 0$
- ▶ Probabilidad $\{ a \leq x \leq b \} = \int_a^b f(x) dx$
- ▶ Normalización $\int_{-\infty}^{\infty} f(x) dx = 1$



Función de Distribución Acumulativa (CDF)

- ▶ $F(x) = \int_{-\infty}^x f(x') dx'$
 $0 \leq F(x) \leq 1$
- ▶ $\frac{dF(x)}{dx} \geq 0$
- ▶ $F(-\infty) = 0, \quad F(\infty) = 1$



Muestreo Directo

Solución directa $\hat{x} = F^{-1}(\xi)$

- ▶ Resolver para \hat{x} : $\xi = \int_{-\infty}^x f(x) dx$

Procedimiento de muestreo

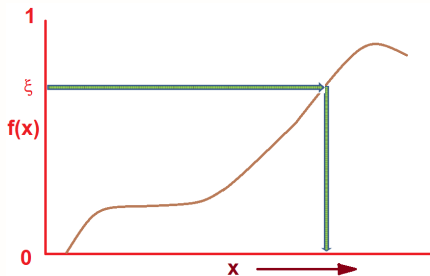
- ▶ Genera ξ
- ▶ Determina \hat{x} tal que $F(\hat{x}) = \xi$

Ventajas

- ▶ Matemáticas y codificación sencillas
- ▶ Alto nivel de aproximación

Desventajas

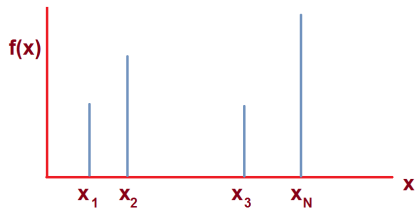
- ▶ A menudo involucra funciones complicadas
- ▶ En algunos casos, $F(x)$ no puede ser invertida (pe. Klein-Nishina)



(Es decir, los números son diferentes, pero las distribuciones son las mismas)

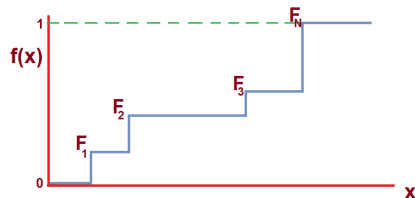
Discreta (PDF)

- ▶ $\{f_k\}$, donde $f_k = f(x_k)$ con $k = 1, 2, \dots, N$
 $f_k \geq 0$
- ▶ $\sum_{j=1}^N f_j = 1$



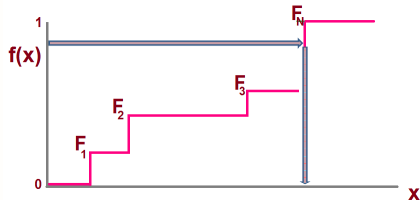
Discreta (CDF)

- ▶ $\{F_k\}$, donde $F_k = \sum_{j=1}^k f_j$ con $k = 1, 2, \dots, N-1$
- ▶ $F_0 = 0$
- ▶ $F_N = 1$



Muestreo desde PDF's discretos

- ▶ Solución directa de $\hat{x} \leftarrow F^{-1}(\xi)$
 - ▶ (1) Genera ξ
 - ▶ (2) Determina k tal que $F_{k-1} \leq \xi \leq F_k$
 - ▶ (3) Regresa $\hat{x} = x_k$
- ▶ Requiere de búsqueda en la tabla
 - ▶ Búsqueda **Lineal**, requiere de tabla $O(N)$;
Cuando N es pequeño
 - ▶ Búsqueda **Binaria**, requiere de tabla $O(\ln_2 N)$;
Cuando N es grande
- ▶ Algunos PDF's y F'_k s discretos, no están pre-calculados
 - ▶ Búsqueda **lineal**, Calcula con F'_k s en el momento lo que se necesita.



Dispersión Multigrupos

⊙ Dispersión del grupo g al grupo g'

$$f_{g'} = \frac{\sigma_{g \rightarrow g'}}{\sum_{i=1}^n \sigma_{g \rightarrow i}} \quad (11)$$

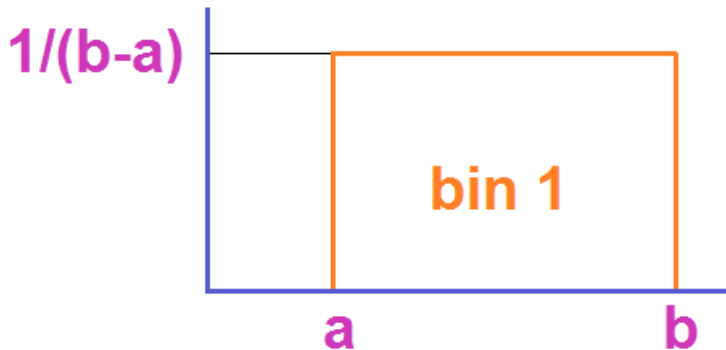
Selección del núclido de dispersión para una colisión

⊙ k = número de núclidos en la composición

$$f_k = \frac{N^{(k)} \sigma_s^{(k)}}{\sum_{j=1}^k N^{(j)} \sigma_s^{(j)}} \quad (12)$$

Ejemplo

- Muestreo desde un PDF uniforme en el intervalo (a, b) , Histograma con 1 bin



$$x \leftarrow a + \xi \cdot (b - a)$$

(13)

Ejemplo

- Muestreo desde un Histograma con 2 bin

▶ $A_1 = (x_1 - x_0) \cdot f_1$

▶ $A_2 = (x_2 - x_1) \cdot f_2$

▶ $p_1 = \text{prob}\{x_0 < x < x_1\} = \frac{A_1}{A_1 + A_2}$

▶ $p_2 = \text{prob}\{x_1 < x < x_2\} = \frac{A_2}{A_1 + A_2}$

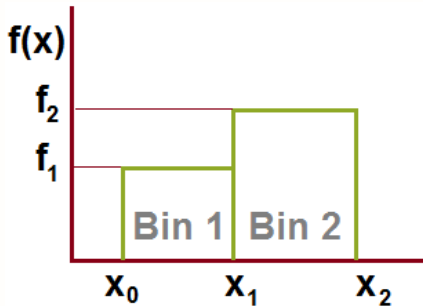
- Procedimiento de muestreo en dos etapas:

- ▶ 1. Seleccione un bin: b

Si $\xi_1 < p_1$, seleccione b=bin 1
de otra manera, seleccione b=bin 2

- ▶ Muestrea x dentro de bin:

$$x \leftarrow x_{b-1} + \xi_2 \cdot (x_b - x_{b-1})$$



Ejemplo

- Muestreo desde un Histograma PDF

Muestreo en dos etapas:

- ▶ 1. Muestreo desde PDF discreto, para seleccionar un bin
- ▶ 2. Muestreo desde PDF uniforme, dentro de un bin

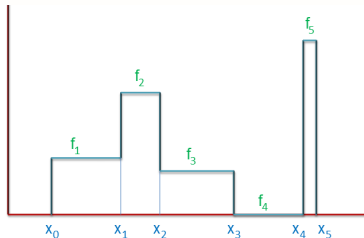
PDF discreto

$$p_k = f_k \cdot (x_k - x_{k-1}), \quad k = 1, 2, \dots, N \quad \sum p_k = 1$$

- ▶ Genera ξ_1
- ▶ Usa búsqueda en tabla para seleccionar k

Muestreo uniforme, dentro del bin k

- ▶ Genera ξ_2
- ▶ Entonces $x \leftarrow x_{k-1} + \xi_2 \cdot (x_k - x_{k-1})$



Ejemplo

- Muestreo desde tramos lineales PDF

Muestreo en dos etapas:

- ▶ 1. Muestreo desde un PDF discreta, para seleccionar un bin
- ▶ 2. Muestreo desde un PDF lineal, dentro de un bin

PDF discreto

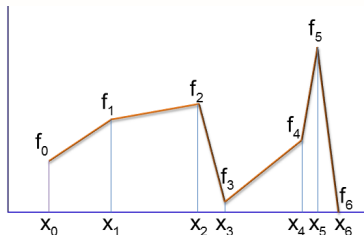
$$p_k = \frac{f_k - f_{k-1}}{2} \cdot (x_k - x_{k-1}), \quad k = 1, 2, \dots, N$$

- ▶ Genera ξ
- ▶ Usa búsqueda en tabla o método "alias" para seleccionar k

Muestreo lineal, dentro del bin k

- ▶ Genera ξ
- ▶ Entonces si:

- ▶ $\xi_1 < \frac{f_{k-1}}{f_k - f_{k-1}},$
 $\hat{x} \leftarrow x_k - (x_k - x_{k-1})\sqrt{\xi_2}$
- ▶ De otra manera,
 $\hat{x} \leftarrow x_{k-1} + (x_k - x_{k-1})\sqrt{\xi_2}$



Muestreo de Rechazo

Von Neumann " *...parece objetable calcular una función trascendental de un número aleatorio*"

- ▶ Seleccione una función de frontera $g(x)$, tal que:
 $c \cdot g(x) \geq f(x)$, para toda x
 $g(x)$ es un PDF fácil de muestrear

Procedimiento de Muestreo

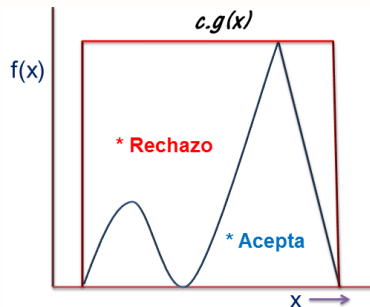
- ▶ Muestrea \hat{x} desde $g(x)$, $\hat{x} \leftarrow G^{-1}(\xi_1)$
- ▶ Prueba $\xi_2 \cdot c g(\hat{x}) \leq f(\hat{x})$
- ▶ Si es verdadero \rightarrow acepta \hat{x} , echo
- ▶ Si es falso \rightarrow rechaza \hat{x} , vuelve a intentarlo

Ventajas

- ▶ Operaciones simples de computadora

Desventajas

- ▶ Bajo nivel de aproximación, a veces difícil de entender

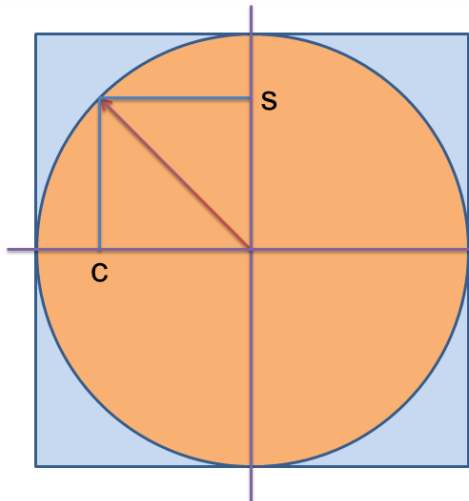


Ejemplo: -Isotrópico 2D

$$f(\vec{\rho}) = \frac{1}{2\pi}, \quad \vec{\rho} = (u, v) \quad (14)$$

Rechazo (anterior VIM)

```
SUBROUTINE AZIRN_VIM( S, C )  
IMPLICIT DOUBLE PRECISION (A-H, O-Z)  
100  R1=2.*RENF() - 1.  
    R1SQ=R1*R1  
    R2=RANF()  
    R2SQ=R2*R2  
    RSQ=R1SQ+R2SQ  
    IF(1.-RSQ)100, 105, 105  
105  S=2.*R1*R2/RSQ  
    C=(R2SQ-R1SQ)/RSQ  
    RETURN  
END
```



Ejemplo: -Isotrópico 2D

Directo (RACER, NEW, VIM)

```
subroutine azirn( S, C )  
implicit double precision(A-H, O-Z)  
parameter ( twopi = 2.*3.14159265 )  
phi=twopi*ranf()  
c = cos(phi)  
s = sen(phi)  
return  
end
```

Tipo	Función Densidad de Probabilidad	Método de Muestreo Directo
Lineal:	$f(x) = 2x, \quad 0 < x < 1$	$x \leftarrow \sqrt{\xi}$
Exponencial:	$f(x) = e^{-x}, \quad 0 < x$	$x \leftarrow -\log \xi$
Isotrópico 2D:	$f(\vec{\rho}) = \frac{1}{2\pi}, \quad \vec{\rho} = (u, v)$	$u \leftarrow \cos 2\pi \xi_1$ $v \leftarrow \sin 2\pi \xi_1$
Isotrópico 3D:	$f(\vec{\Omega}) = \frac{1}{4\pi}, \quad \vec{\Omega} = (u, v, w)$	$u \leftarrow 2\xi_1 - 1$ $v \leftarrow \sqrt{1 - u^2} \cos 2\pi \xi_2$ $w \leftarrow \sqrt{1 - u^2} \sin 2\pi \xi_2$
Maxwelliano:	$f(x) = \frac{2}{T\sqrt{\pi}} \sqrt{\frac{x}{T}} e^{-\frac{x}{T}}, \quad x$	$x \leftarrow T(-\log \xi_1 - \log \xi_2 \cos^2 \frac{\pi}{2} \xi_3)$
Espectro de Watt:	$f(x) = \frac{2e^{\frac{-ab}{4}}}{\sqrt{\pi a^3 b}} e^{\frac{-x}{a}} \sinh \sqrt{bx}, \quad x < 0$	$w \leftarrow a(-\log \xi_1 - \log \xi_2 \cos^2 \frac{\pi}{2} \xi_3)$ $x \leftarrow w + \frac{a^2 b}{4} + (2\xi_4 - 1)\sqrt{a^2 b w}$
Normal:	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$	$x \leftarrow \mu + \sigma \sqrt{-2\log \xi_1} \cos 2\pi \xi_2$

Reglas de oro para el diseño de algoritmos que han cambiado

- ▶ Nunca tomar la raíz cuadrada de un número aleatorio
- ▶ Evite usar \sin , \cos , \log , \exp ,
- ▶ Use *IF...GOTO...* para evitar la aritmética
- ▶ Números aleatorios son baratos, la aritmética es cara

Los métodos de muestreo directo tienen ventajas

- ▶ Codificación clara y sucinta - fácil de verificar y mantener
- ▶ El tiempo de CPU es comparable al rechazo (???)
- ▶ Los métodos directos, vectorizan eficientemente

Todos los desarrolladores de código de Monte Carlo que trabajan con un muestreo aleatorio deben poseer y leer las siguientes referencias:

- ▶ D. E. Knuth, The Art of Computer Programming, Vol. 2: Semi-numerical Algorithms, 3rd Edition, Addison-Wesley, Reading, MA (1998).
- ▶ L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, NY (1986).
- ▶ J. von Neumann, "Various Techniques Used in Conjunction with Random Digits", J. Res. Nat. Bur. Stand. Appl. Math Series 3, 36-38 (1951).
- ▶ C. J. Everett and E. D. Cashwell, "A Third Monte Carlo Sampler", LA9721-MS, Los Alamos National Laboratory, Los Alamos, NM (1983).
- ▶ H. Kahn, "Applications of Monte Carlo", AECU-3259, Rand Corporation, Santa Monica, CA (1954).