

Métodos de Montecarlo

Curso de Física Computacional
Facultad de Ciencias

M. en C. Gustavo Contreras Mayén

Secuencia aleatoria

Se define una secuencia de números r_1, r_2, \dots como aleatoria si no existe una correlación entre ellos.

Aunque sean aleatorios, no implica que todos los números en la secuencia tengan la misma probabilidad de ocurrir. Si todos los números en la secuencia tienen la misma probabilidad de ocurrir, se dice que la secuencia es uniforme y los números son aleatorios.

Por ejemplo, $1, 2, 3, 4, \dots$ es uniforme pero probablemente no es aleatoria.

También es posible tener una secuencia de números que de alguna forma son aleatorios, pero tienen correlación dentro de un intervalo pequeño:

$$r_1, (1 - r_1), r_2, (1 - r_2), r_3, (1 - r_3), \dots$$

Matemáticamente, la probabilidad de que un número ocurra, está descrita por una función de distribución $P(r)$, donde $P(r) dr$, es la probabilidad de encontrar r en un intervalo $[r, r + dr]$.

Una distribución uniforme significa que $P(r)=\textit{constante}$.

El generador estándar de números aleatorios en las computadoras, genera distribuciones uniformes entre 0 y 1.

El generador estándar de números aleatorios, proporciona números en éste intervalo, y cada uno de ellos tiene la misma probabilidad de ocurrir, y además es independiente del número anterior.

Veremos que los números pueden generarse de manera no uniforme y ser aún así, aleatorios.

Las computadoras por naturaleza, son determinísticas y no pueden crear una secuencia de números aleatorios.

Pueden crear secuencias que contengan correlaciones y por tanto no ser totalmente aleatorias; si conocemos r_m y su elemento precedente, es posible estimar r_{m+1} .

Por ésta razón, se dice que las computadoras son generadores de números *pseudo-aleatorios*.

Algoritmo para generar números aleatorios.

El método de congruencia lineal es la manera más común que se utiliza para generar una secuencia de números pseudo-aleatorios

$0 \leq r_i \leq M - 1$ en el intervalo $[0, M - 1]$.

Podemos multiplicar el número aleatorio previo r_{i-1} por una constante a , sumar otra constante c , operar con el módulo M , manteniendo sólo la parte fraccional del resultado como el siguiente número aleatorio r_{i+1}

$$r_{i+1} = (ar_i + c) \bmod M$$

El valor de r_1 (se le llama *semilla*) lo proporciona normalmente el usuario.

Ejemplo

Veamos la secuencia que se genera si $c = 1$, $a = 4$, $M = 9$, y la semilla es $r_1 = 3$:

$$r_{i+1} = (ar_i + c) \bmod M$$

$$r_1 = 3$$

$$r_2 = (4 \times 3 + 1) \bmod 9 = 13 \bmod 9 = 4$$

$$r_3 = (4 \times 4 + 1) \bmod 9 = 17 \bmod 9 = 8$$

$$r_4 = (4 \times 8 + 1) \bmod 9 = 33 \bmod 9 = 6$$

$$r_{5-10} = 7, 2, 0, 1, 5, 3$$

Hemos obtenido una secuencia de longitud $M = 9$, antes de que la secuencia se repitiera.

Si queremos números en el rango $[0, 1]$, basta dividir los números r por $M = 9$:

0.333,	0.222,
0.444,	0.000,
0.889,	0.111,
0.667,	0.555,
0.778,	0.333.

Aún sigue siendo una secuencia de longitud 9 pero ya no es una secuencia de enteros.

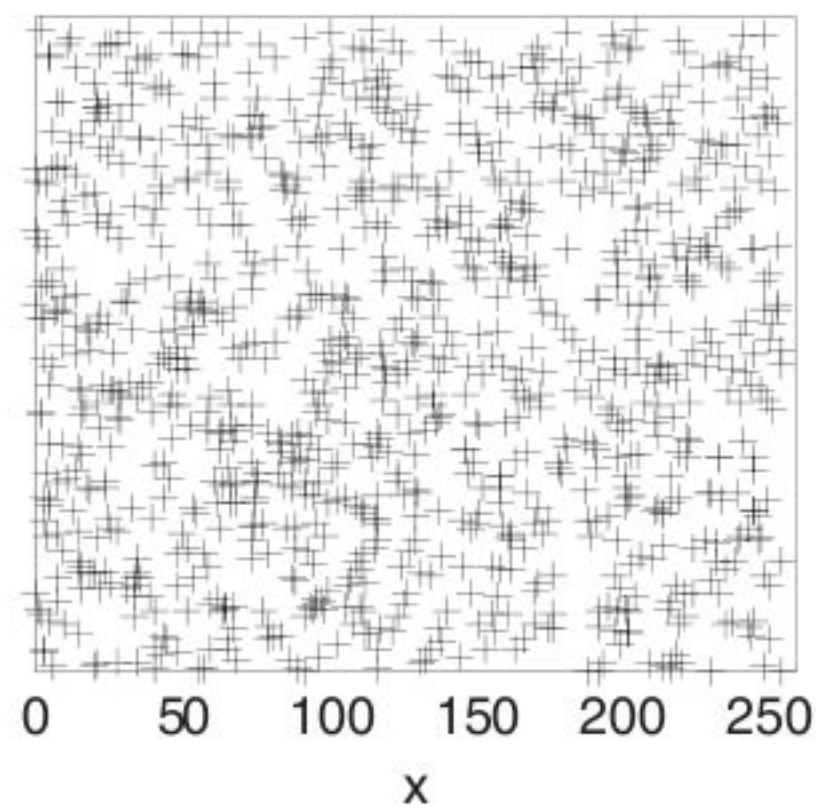
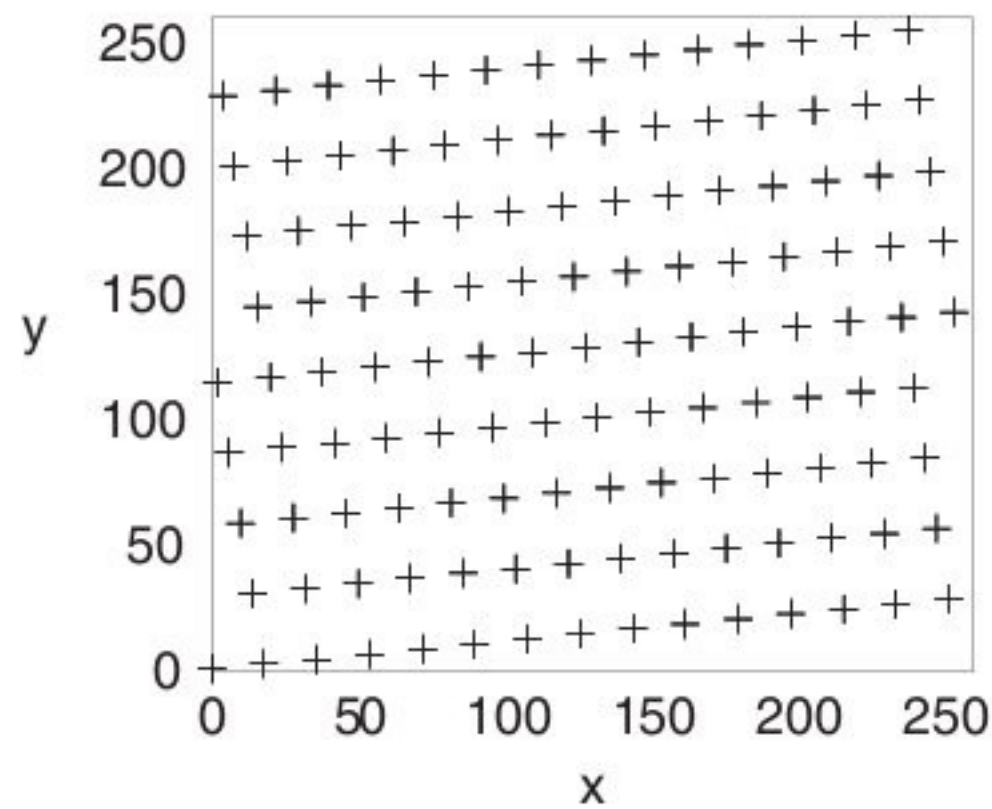
Si queremos que los números aleatorios estén en el rango $[A, B]$, se requiere aplicar el factor de escala:

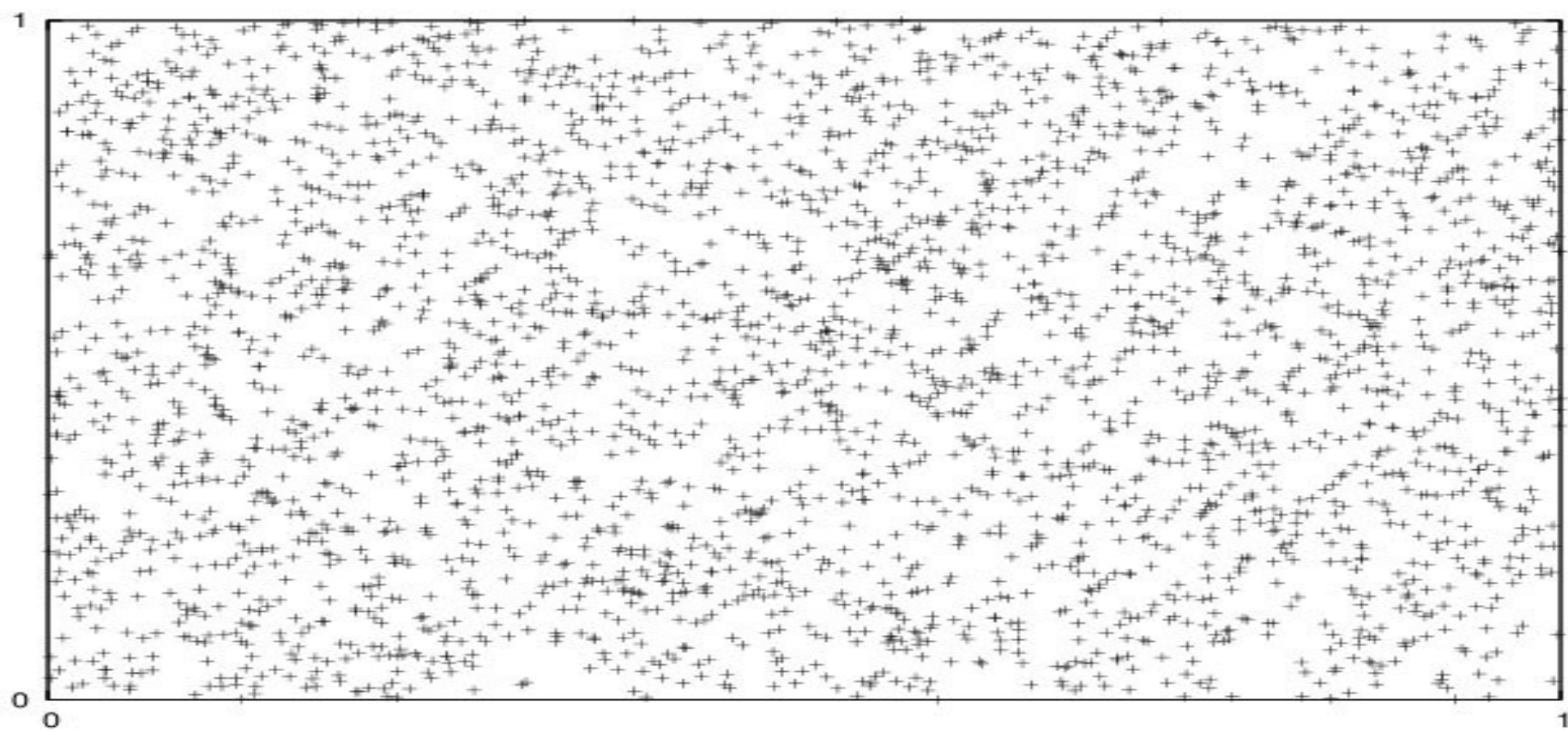
$$x_i = A + (B - A)r_i \quad 0 \leq r_i \leq 1 \rightarrow A \leq x_i \leq B$$

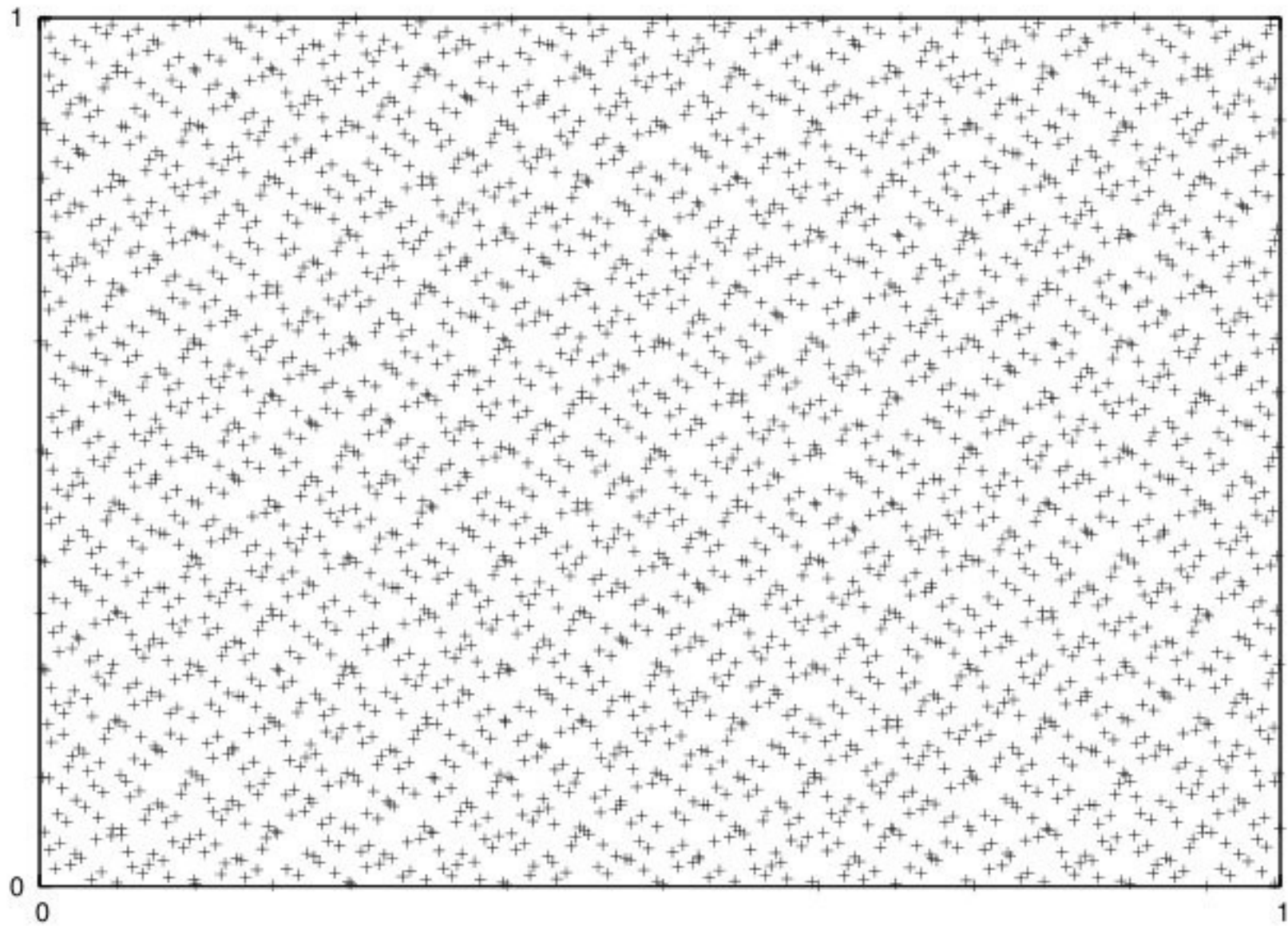
Sugerencia

Antes de utilizar un generador de números aleatorios en nuestros programas, debemos de revisar que el rango que produce, tiene apariencia de aleatorios.

Propiamente no es un prueba matemática, pero al graficar la distribución de números aleatorios, podemos reconocer ciertos patrones, con lo que nos diría si tenemos o no, números aleatorios.







Ejercicio

Compila, ejecuta y grafica los resultados del archivo de datos que se crea.

```
PROGRAM aleatorio
```

```
    imax = 100
```

```
    Open(1, FILE="Aleatorio.dat")
```

```
    DO i = 1, imax
```

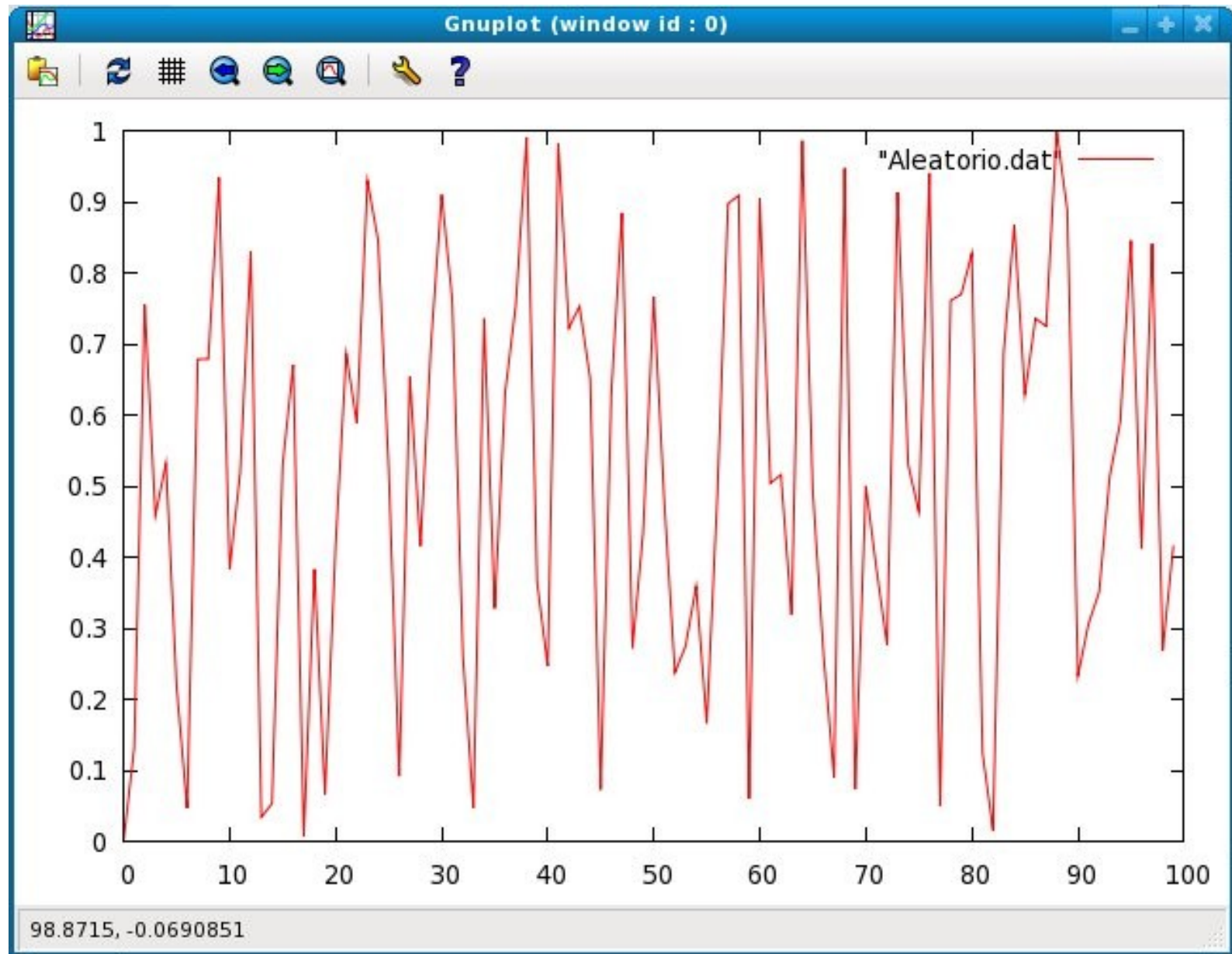
```
        WRITE(1,*) rand(0)
```

```
    END DO
```

```
    Close(1)
```

```
END PROGRAM aleatorio
```


Gráfica de la secuencia obtenida



Camino aleatorio

Queremos calcular un camino aleatorio, es decir, imaginamos una persona caminando en direcciones aleatorias dando pasos de una longitud dada por distribuciones aleatorias, pero de un tipo dado: gaussianas y uniformes.

Camino aleatorio

Para ello debemos obtener dos datos al azar, que son la longitud del paso y su dirección.

La longitud del paso la obtenemos mediante una de las dos fórmulas siguientes, suponiendo que z_1 , z_2 , z_3 y z_4 son dos números aleatorios entre 0 y 1

Distribución gaussiana: $r = x_{media} + \sigma \sqrt{-2 \log z_1} \cos(2 \pi z_2)$

Distribución uniforme: $r = x_{media} + (x_{máx} - x_{min}) z_3$

Para hallar la dirección del paso usaremos un ángulo, que podemos obtener como:

$$\theta = 2 \pi z_4$$

Por otro lado, una vez obtenida la longitud del paso y el ángulo que marca la dirección, determinaremos la coordenada en la que el caminante se encuentra, a través de las fórmulas:

$$x_i = x_{i-1} + r \cos \theta$$

$$y_i = y_{i-1} + r \sin \theta$$

Hacemos un número dado de simulaciones, dado por el parámetro *nsim* (por defecto, *nsim* = 100) , de un “paseo” aleatorio de 100000 (cien mil) pasos, definidos mediante el parámetro *nmax*.

Después hacemos simultáneamente el paseo para la distribución gaussiana y para la distribución uniforme.

En cada simulación, calculamos para cada valor del número de pasos la distancia al cuadrado recorrida, y vamos sumándola a un contador correspondiente a ese número de pasos.

Finalmente, cuando se han hecho todas las simulaciones, calculamos la media, y hallamos su raíz. Podemos hacerlo también al revés, hallar la raíz y después calcular la media, pero esto solo influirá en la pendiente de la teórica recta que debemos obtener.

El criterio que seguimos es el siguiente: hallamos la media de la distancia cuadrática, y de ella, hallamos la raíz, obteniendo así una estimación de la distancia a la que estará en cada paso.

Todos estos datos (para los parámetros seleccionados, se obtiene un total de 23 MB de datos) se almacenan en los siguientes archivos:

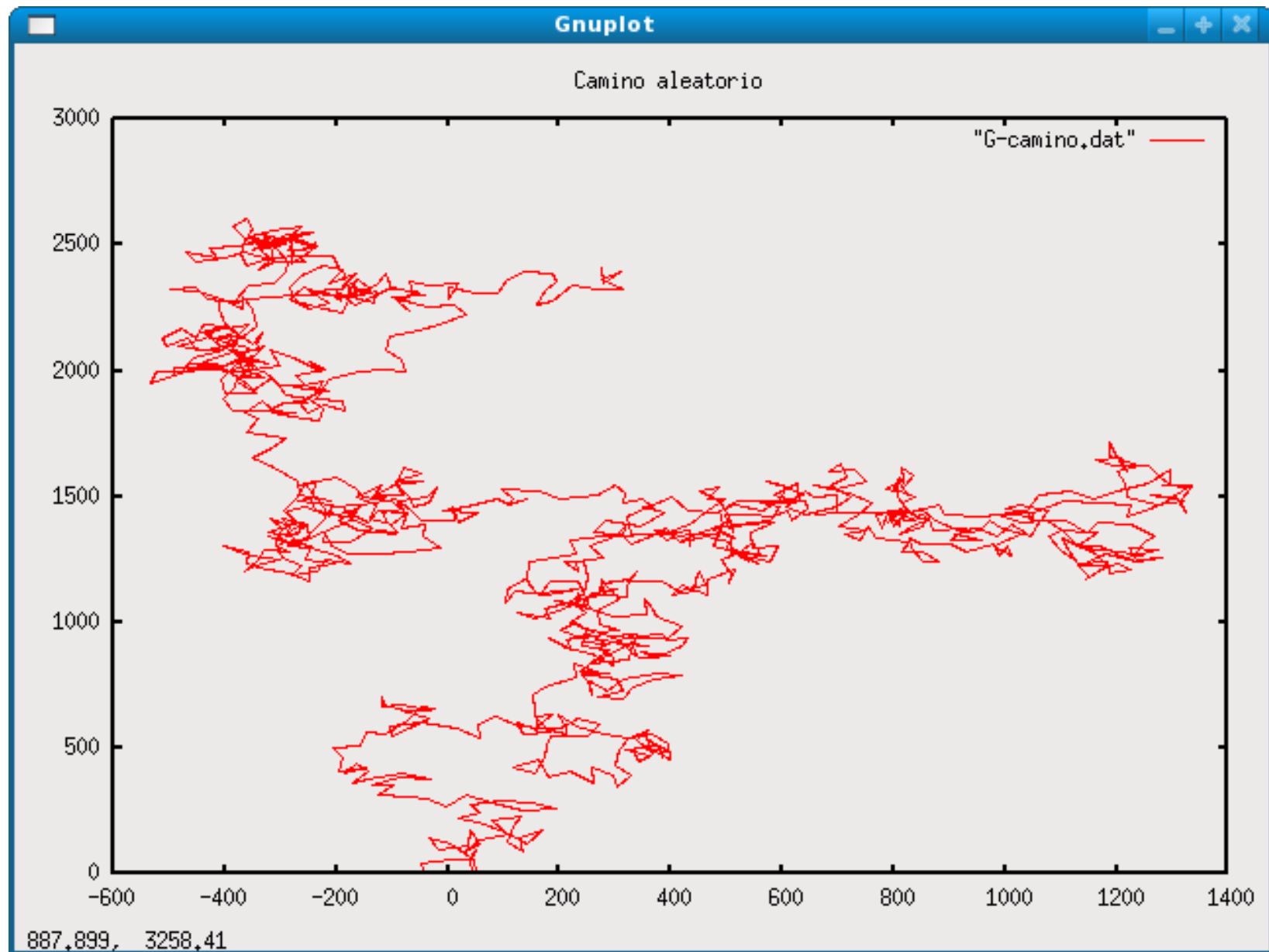
Distribución gaussiana:

- Dibujo del paseo: G-camino.dat
- Ajuste y datos de la distancia cuadrática media:
gauss.dat

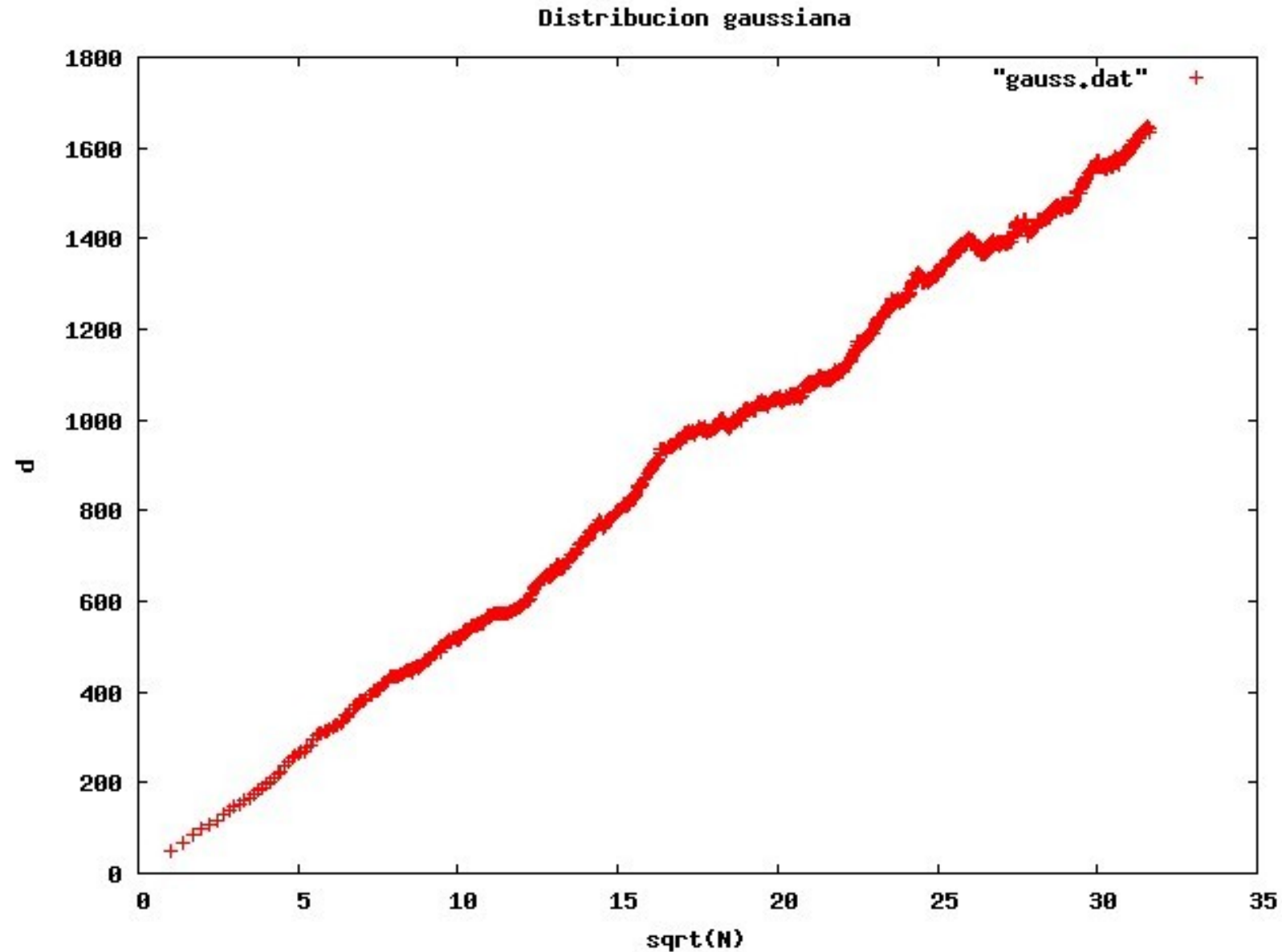
Distribución uniforme:

- Dibujo del paseo: U-camino.dat
 - Ajuste y datos de la distancia cuadrática media:
unif.dat

Camino aleatorio



Datos para la distribución gaussiana



Aguja de Buffon

El problema de la aguja de Buffon consiste en la aplicación de un método aleatorio para hacer una estimación del número π .

Según esta estimación, si dejamos caer agujas de longitud l en una loseta cuadrada de lado d , la probabilidad de que aquellas corten con la losa viene dada por:

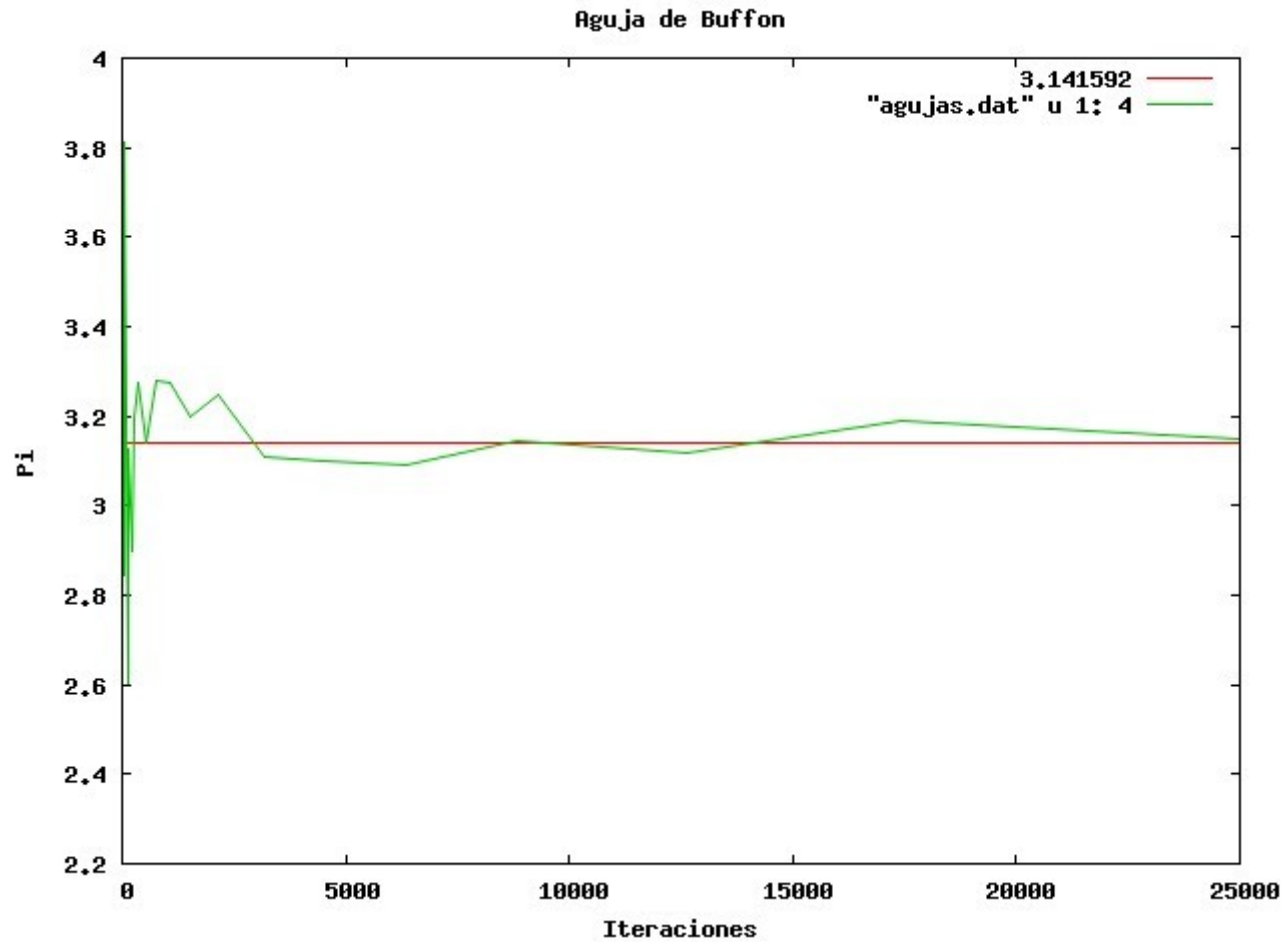
$$P = \frac{4l}{\pi d} \quad \rightarrow \quad \pi = \frac{4l}{Pd}$$

La probabilidad de que una aguja y alguno de los lados se corten podemos simularla tirando muchas agujas, y viendo cuántas de éstas cortan a la loseta respecto del número total de tiradas.

En esto consiste el programa que veremos.
Haremos varias simulaciones con un número mayor de agujas lanzadas, y calculando en cada uno de estos pasos el correspondiente valor de π .

El valor del número de datos con el que hay que realizar el ajuste es, evidentemente, el número de pasos que se da en la simulación, que viene dado por el parámetro n_{\max} , que por defecto vale $n_{\max} = 1000001$.

Representación de los valores



Uso del método Montecarlo

Esta técnica es aplicable en numerosos campos, tal como el de física de radiaciones, así como también en matemáticas (evaluaciones de integrales), ingeniería ambiental (crecimiento de bosques y estudios de contaminación), economía (análisis de mercado y crecimiento de PIB), biología molecular (interacción de moléculas de ADN) y muchas otras.

Uso en radiodiagnóstico

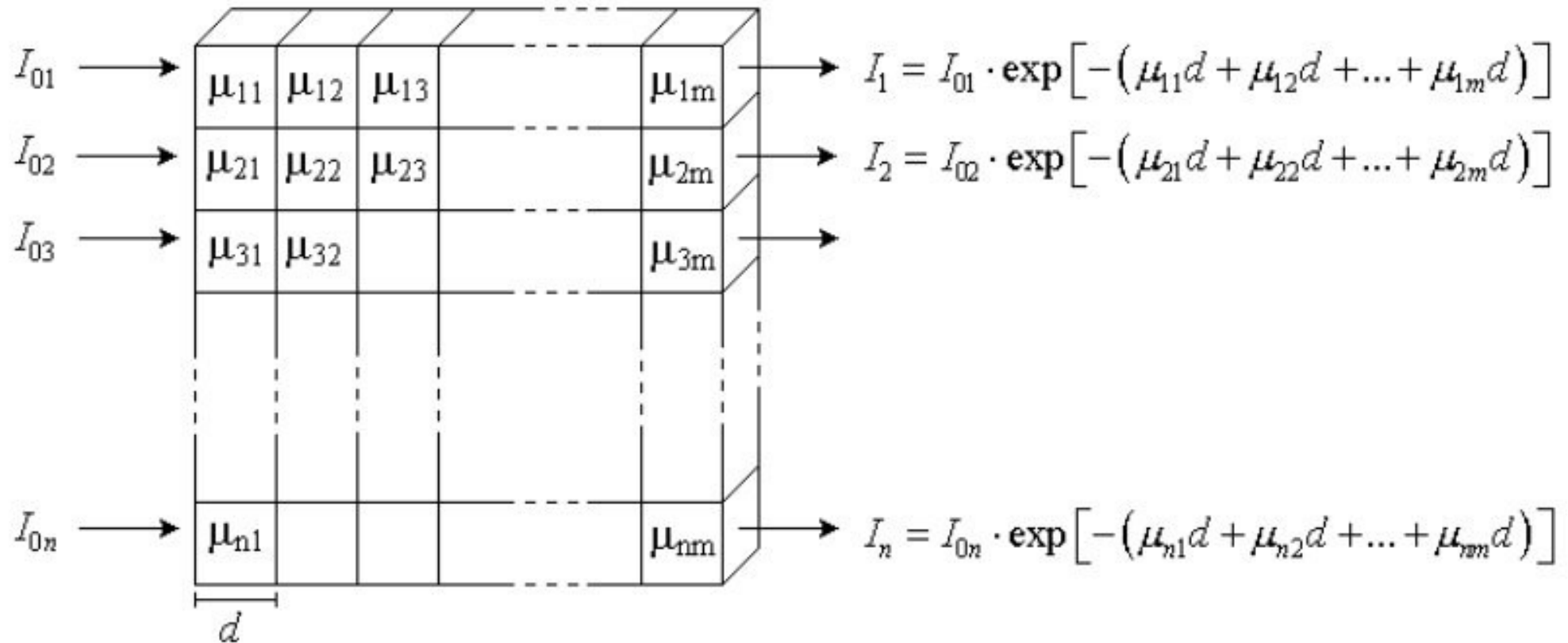
Si se trabaja en el campo de las radiaciones ionizantes, el método de Montecarlo hace uso de las distribuciones de probabilidad de las interacciones individuales para simular la trayectoria errática de las partículas.

Todos los datos físicos que van a determinar el transporte de las partículas estarán en los algoritmos del código usado, de modo que mediante secuencias de números aleatorios se pueda simular lo que realmente ocurre en la naturaleza.

Los fotones (o los neutrones) y las partículas cargadas (electrones y positrones), al atravesar la materia producen una serie de interacciones con los átomos que la forman.

Todos estos fenómenos de absorción, dispersión y producción de partículas secundarias siguen un proceso aleatorio, es decir, no se puede prever qué tipo de interacción se va a producir en cada momento y lugar, sino que solamente se puede asignar una probabilidad a cada uno de los posibles sucesos.

Imagen dividida en voxels

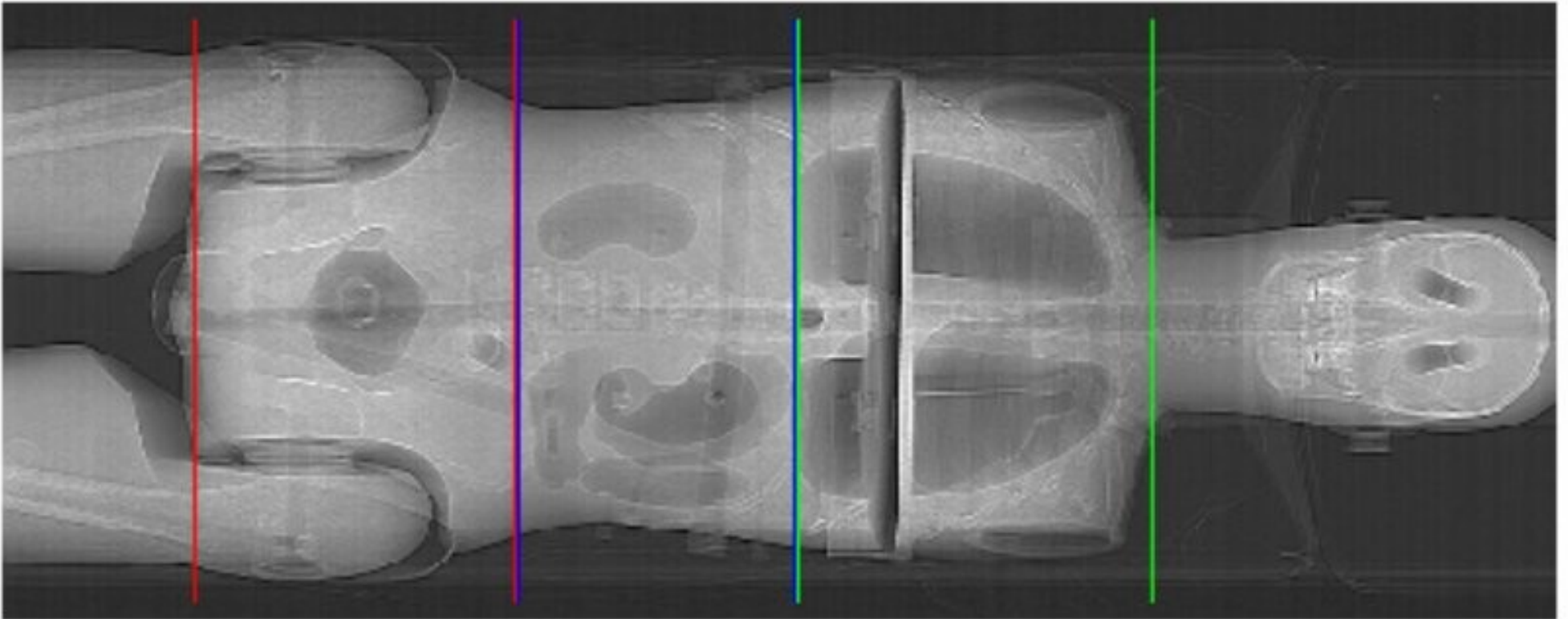


El método de Montecarlo escoge al azar cada una de las posibles variables, de acuerdo con su función de probabilidad, para reproducir los fenómenos que tienen lugar en el sistema a estudiar, y así, poder cuantificar finalmente el valor medio de aquellas magnitudes que interesan para la solución del problema.

La simulación mediante ordenadores y técnicas de Montecarlo, ha sido utilizada con frecuencia para estudiar problemas asociados con la utilización de radiaciones ionizantes en medicina:

- cálculo de dosis en radioterapia
- estudios de sistemas para la obtención de imágenes con radionucleidos o rayos X,
- caracterización de las fuentes y detectores,
- estudios específicos en mamografía,
- cálculo de radiación dispersa aplicables al diseño de salas de diagnóstico,
- optimización de técnicas en radiodiagnóstico convencional, etc.

Uso de maniqui



Se ha convertido en una de las mejores alternativas disponibles en la actualidad para resolver el problema del transporte de la radiación ionizante en la materia, ya que permite realizar cálculos, con suficiente precisión, de parámetros relacionados con dicho transporte en distintas aplicaciones médicas sobre geometrías tan complejas como el cuerpo humano.

Códigos de simulación

La utilización del método de Montecarlo requiere la elaboración de una serie de rutinas que controlan las interacciones de la radiación con la materia.

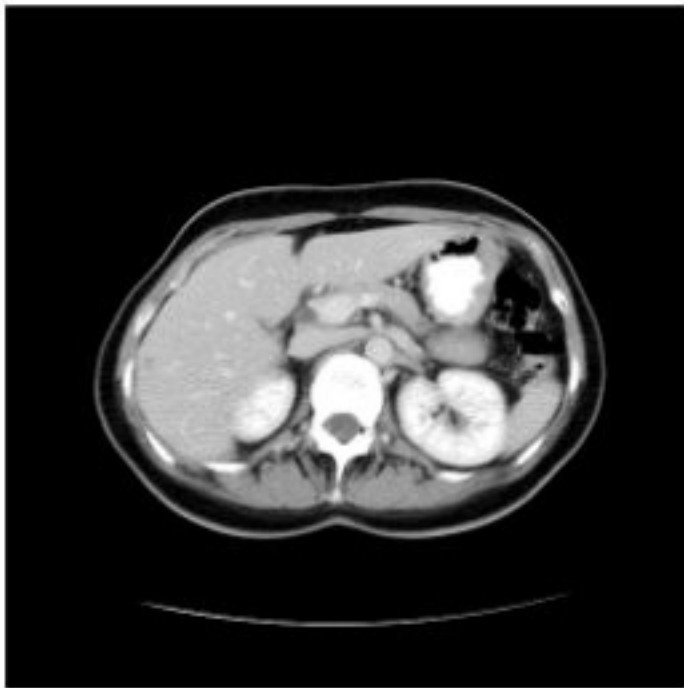
Esta es una tarea tediosa y complicada pero, afortunadamente, existen códigos ya elaborados que podemos utilizar a nuestra conveniencia siempre que definamos las rutinas específicas y las adaptemos a cada estudio concreto.

Entre los más usados, revisados y validados, se pueden destacar EGS4, MCNP, PENÉLOPE, ITS y GEANT.

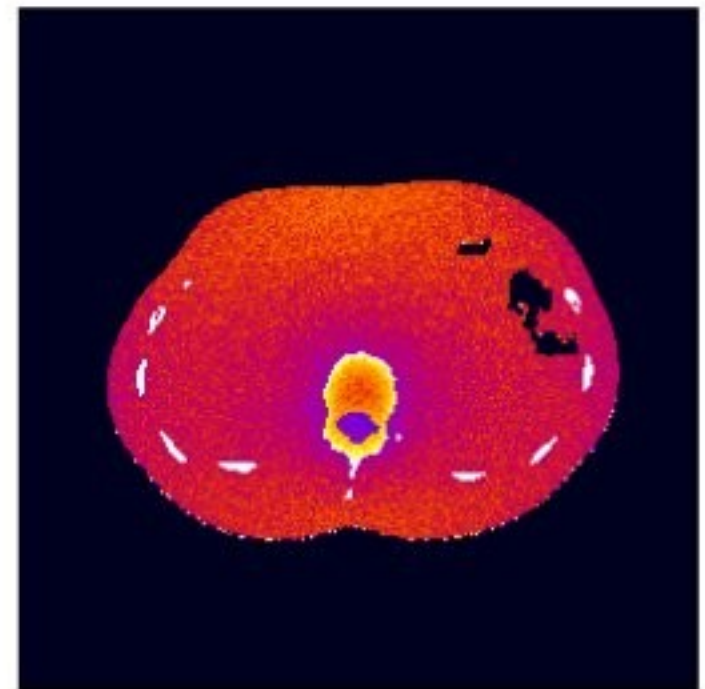
El esquema de simulación del transporte de partículas es, con diferencias menores, semejante en los distintos códigos desarrollados. Las diferencias entre éstos se centran en las aproximaciones o modelos usados para describir las interacciones y en su mayor o menor flexibilidad a la hora de interactuar con el usuario (datos de entrada, datos de salida, programación por parte del usuario, etc.).

Resultados de la simulación

Imagen de TC



Distribución de dosis



0 Dosis normalizada 1