Informe de Laboratorio N°3

Alumno: Méndez Martín
Docentes: Dra. Marconi Verónica I.; Dr. Banchio Adolfo
Universidad Nacional de Córdoba (UNC)
Facultad de Matemática, Astronomía, Física y Computación (FaMAF)
Curso de Física Computacional: Problema N°2

I. Introducción

I-1. Problema 2 - Caminatas al azar

En una red cuadrada bidimensional implementar un algoritmo que realice una caminata al azar de n pasos. Iniciar la caminata en el sitio central de la red $R(0) = (0, 0)$.

- a) Hallar el desplazamiento cuadrático medio $\langle [R(n) - R(0)]^2 \rangle$ en función del número de pasos n, promediando ($\langle \ldots \rangle$) sobe $N = 10^6$ realizaciones de la caminata. ¿Se verifica la ley $\langle R^2 \rangle \sim n$?
- b) Subdividir la red en cuatro cuadrantes y contabilizar la cantidad de veces que el caminante termina en un dado cuadrante, comparar con el valor esperado $N/4$. Grafique estos resultados en función de n, comparando los resultados de distintos generadores (emplear como generadores de números al azar, el **ran2** del Numerical Recipes, el **mzran** y el **mt199337** Mersenne Twister).
- c) Ídem a) pero para una partícula real en 3 dimensiones (por ejemplo una molécula aromática que se difunde en una sala).

I-2. Generadores de números pseudo-aleatorios

El algoritmo **Random number generator Mersenne Twister (MT19937)** tiene un periodo primo tremendamente largo, $219937 - 1$. La implementación del algoritmo genera un número aleatorio de 32 bits de longitud (precisión simple). Las ventajas de este algoritmo es que ha pasado numerosas pruebas de aleatoriedad estadística y, al momento de realizar las optimizaciones del compilador, tiene mayor capacidad de mejorar su performance que la función **ran0** o **ran2** del Numerical Recipes. Su mayor desventaja, es que, al trabajar con arreglos, donde se almacenan los valores aleatorios generados, se requiere un mayor uso de memoria, por lo que al aumentar la cantidad de pasos del caminante se deberá requerir mayor tiempo de ejecución del programa. Por otro lado, **mzran** ofrece muy buenos resultados, aunque se tuvo la desventaja de que no fue posible modificar las semillas preestablecidas en la subrutina generadora, cosa que limita la repetibilidad de los resultados. Finalmente, los métodos congruenciales como **ran0** o **ran2** poseen la desventaja de que tienen correlaciones que pueden ser significativas, si uno observa en detalle la distribución de números es posible detectar hiper-planos paralelos donde los números se sitúan.

Cabe aclarar que existen procesos físicos capaces de generar variables estocásticas, como por ejemplo la radiación térmica, emisión de radioisótopos α o β, ruletas de casinos, dados, etc., sin embargo, estos procesos no cuentan con suficiente velocidad computacional que se requiere en los problemas de física. Por ello, fue necesario utilizar generadores de números pseudo-aleatorios (y quasi-aleatorios, que son menos aleatorios que los pseudo y están acotados en un dominio particular, y que son útiles en problemas específicos). La denominación de números "pseudo"se debe a que estos son representados por procesos no estocásticos, sino deterministas (como es el caso de las computadoras), entonces, es necesario utilizar generadores con un gran periodo para obtener ventajas computacionales. También, se buscan tres características fundamentales en los generadores, portabilidad (para no depender de las arquitecturas de las computadoras), velocidad computacional (para obtener mayor performance) y repetibilidad (para comparar resultados entre distintas simulaciones y testear los programas).

II. Resultados y Discusiones

II-A. Random Walk

Se recolecataron las posiciones del caminante para tres caminatas aleatorias con un mismo generador de números aleatorios (mzran), para una longitud de paso fija de 1, y una cantidad de pasos fija de 1 millón, los resultados se muestran en la figura (ver 1). Aquí se puede observar la forma típica de las

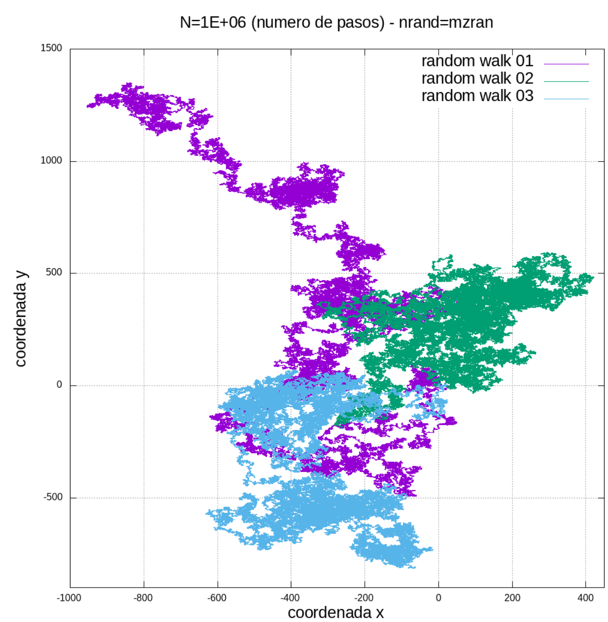

Fig. 1. caminatas aleatorias (muestreo del espacio de configuraciones)

caminatas aleatorias, y además nos permite corroborar que, a pesar de haber simulado un millón de pasos por cada caminata el espacio de configuraciones no se abarca completamente, es decir, que existen correlaciones o tendencias del caminante a moverse en determinados lugares del espacio. Es por ello, que se plantea, resolver el problema con un cambio de enfoque, es decir, no realizar una única caminata muy larga, sino muchas caminatas cortas para barrer todo el espacio de fases y tener una mejor estadística. Estos dos enfoques son totalmente equivalentes por tratarse de sistemas ergódicos, es decir, que el valor medio temporal de cualquier variable dinámica del sistema es igual al promedio de esa variable sobre el ensamble.

II-B. Desplazamiento cuadrático medio (dcm)

Para computar el desplazamiento cuadrático medio se utilizó la siguiente expresión

$$dcm = \left\langle |\overrightarrow{x}(t) - \overrightarrow{x}(0)|^2 \right\rangle = \frac{1}{N}\sum_{i=1}^{N} |\overrightarrow{x}_i(t) - \overrightarrow{x}_i(0)| \quad (1)$$

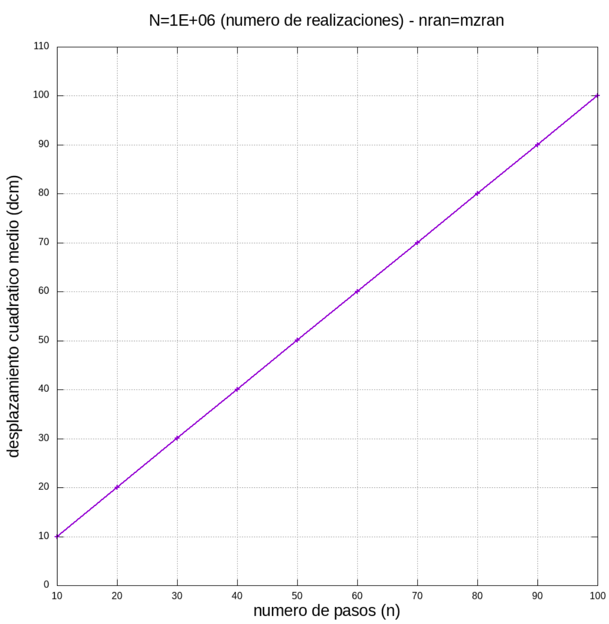

Fig. 2. desplazamiento cuadrático medio vs número de realizaciones

Notemos que se cumple la dependencia lineal (ver 2) entre dcm y el número de realizaciones N, específicamente, debería cumplirse que $\left\langle (\Delta \overrightarrow{x})^2 \right\rangle = N \left\langle (\Delta \overrightarrow{s})^2 \right\rangle$, donde $\left\langle (\Delta \overrightarrow{s})^2 \right\rangle$ sería la dispersión del desplazamiento por paso (en nuestro caso sería uno, pues cada longitud de paso es fija) y $\left\langle (\Delta \overrightarrow{x})^2 \right\rangle$ sería una medida del cuadrado del ancho de la distribución del desplazamiento neto, respecto de la media. Al ser lineal esta dependencia vemos que a medida que aumenta el número de realizaciones N, la distribución de probabilidades se va ensanchando, análogo a un problema de difusión de una partícula en un medio, a medida que transcurre el tiempo.

II-C. Contabilidad de cuadrantes

Para determinar en qué cuadrante el "borracho"termina luego de una caminata aleatoria, con condicionales se evaluó la posición en el plano y en función de está se pesó el contador de cada cuadrante específico es decir, si el caminante estaba un cuadrante sin ambiguedad el contador de ese cuadrante se incrementaba en 1, sin embargo, en los casos en que el caminante estaba en un semieje los contadores de los dos cuadrantes adyacentes se incrementaban en $1/2$ y si el caminante estaba en el origen los contadores de los cuatros cuadrante se incrementaban en $1/4$, el bloque de código sería de la forma,

```
! Determinamos el cuadrante de la part cula al
    final de la caminata
cond2: if (x>0._dp.and.y>0._dp) then
            count_cuad_01=count_cuad_01+1._dp
            exit cond2
    else if (x<0._dp.and.y>0._dp) then
        count_cuad_02=count_cuad_02+1._dp
        exit cond2
    else if (x<0._dp.and.y<0._dp) then
        count_cuad_03=count_cuad_03+1._dp
        exit cond2
    else if (x>0._dp.and.y<0._dp) then
        count_cuad_04=count_cuad_04+1._dp
        exit cond2
    else if (x==0._dp.and.y==0._dp) then !
        origen de coordenadas
        count_cuad_01=count_cuad_01+0.25_dp
        count_cuad_02=count_cuad_02+0.25_dp
        count_cuad_03=count_cuad_03+0.25_dp
        count_cuad_04=count_cuad_04+0.25_dp
        exit cond2
    else if (x>0._dp.and.y==0._dp) then ! semi-
        eje x positivo
        count_cuad_01=count_cuad_01+0.5_dp
        count_cuad_04=count_cuad_04+0.5_dp
        exit cond2
    else if (x<0._dp.and.y==0._dp) then ! semi-
        eje x negativo
        count_cuad_02=count_cuad_02+0.5_dp
        count_cuad_03=count_cuad_03+0.5_dp
        exit cond2
    else if (x==0._dp.and.y>0._dp) then ! semi-
        eje y positivo
        count_cuad_01=count_cuad_01+0.5_dp
        count_cuad_02=count_cuad_02+0.5_dp
        exit cond2
    else if (x==0._dp.and.y<0._dp) then ! semi-
        eje y negativo
        count_cuad_03=count_cuad_03+0.5_dp
        count_cuad_04=count_cuad_04+0.5_dp
        exit cond2
end if cond2
```

Los resultados (ver 3,4,5,6) muestran que la cantidad de veces que el caminante termina en un determinado cuadrante oscila alrededor de la media correcta $N/4$, los cual nos dice que los resultados se podrían considerar çorrectos", sin embargo, se obtienen variaciones significativas con todos los generadores, lo cual podría deberse a la correlación que existen entre los números generados, o poca cantidad de caminatas, es decir, a se esperaría que, a media que aumenta el número de realizaciones se mejora la estadística y se obtiene una oscilación entorno a la media menos significativa.

III. CÓDIGOS

Repositorio de GitHub

- https://github.com/mendzmartin/fiscomp2022.git

Repositorio GitHub del problema

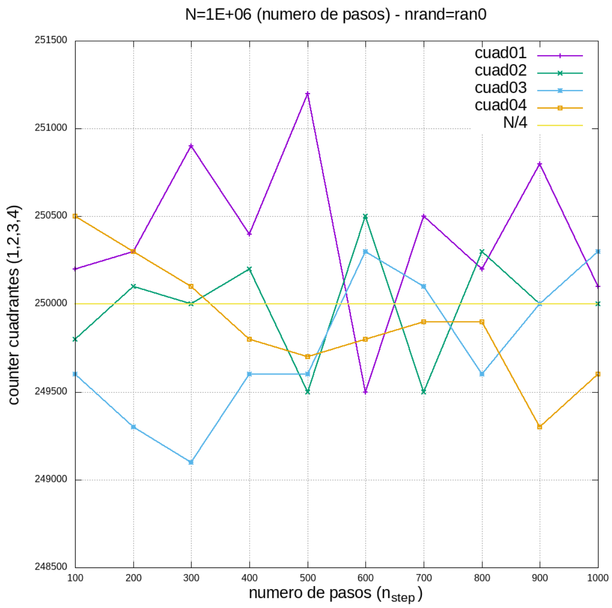

Fig. 3. contador cuadrante vs número de pasos

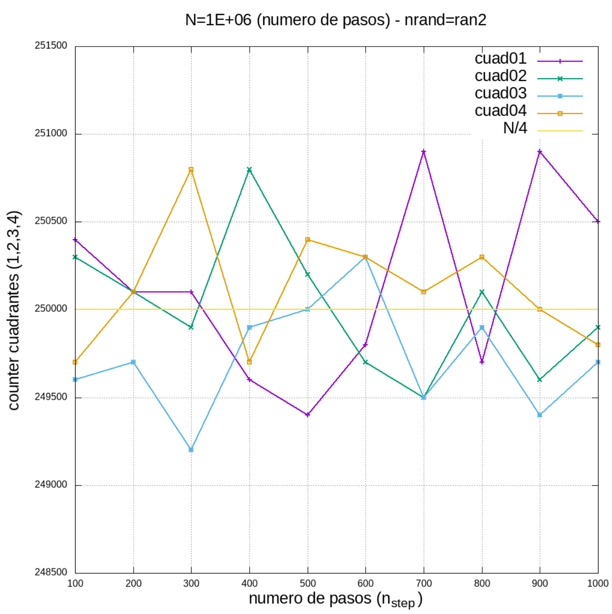

Fig. 4. contador cuadrante vs número de pasos

- https://github.com/mendzmartin/fiscomp2022/tree/main/lab03/prob02

Códigos principales y Makefile

- https://github.com/mendzmartin/fiscomp2022/blob/main/lab03/prob02/code/random_walk_2d.f90
- https://github.com/mendzmartin/fiscomp2022/blob/main/lab03/prob02/code/Makefile

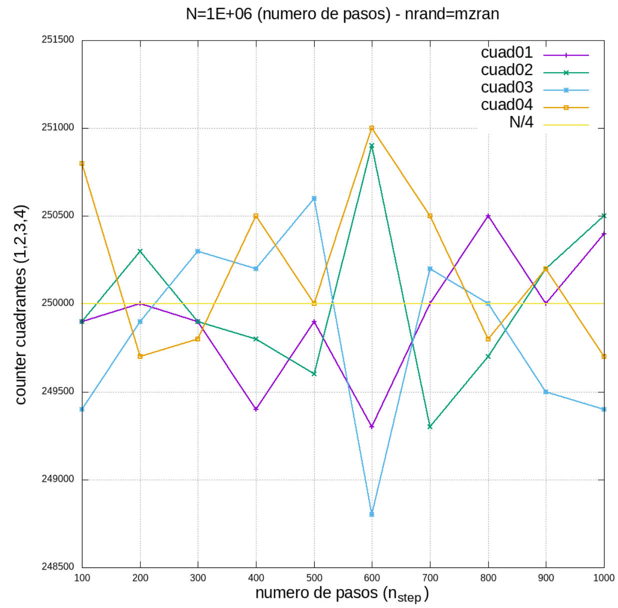

Fig. 5. contador cuadrante vs número de pasos

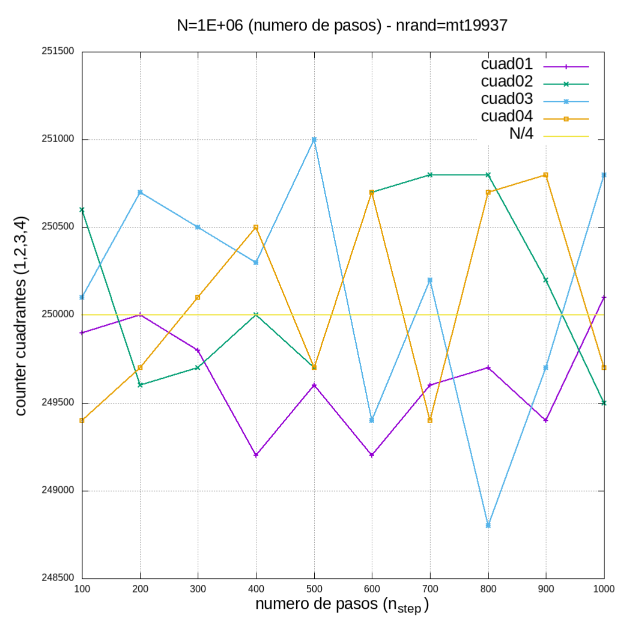

Fig. 6. contador cuadrante vs número de pasos