

CAPÍTULO 2

EL MÉTODO DE MONTE CARLO Y EL PROGRAMA DE CÓMPUTO MCNPX.

Para poder realizar el análisis del combustible del GT-MHR se utilizará uno de los más poderosos métodos de física de reactores nucleares, el método de Monte Carlo, debido a que los problemas del tipo de transporte de partículas nucleares analizados en este trabajo de tesis son resueltos de una manera más precisa con este método que con los métodos de tipo determinísticos.

En este capítulo haremos una breve descripción de los fundamentos teóricos del método de Monte Carlo y su aplicación al transporte de partículas. Posteriormente veremos el uso de este método como herramienta computacional en el código MCNPX, y cómo este código es empleado para el análisis del combustible en el reactor nuclear GT- MHR.

2.1 MÉTODO DE MONTECARLO

El método de Monte Carlo es un método estadístico (no determinístico) que proporciona soluciones aproximadas a una gran variedad de problemas matemáticos, haciendo factible la realización de experimentos con muestreos de números pseudo-aleatorios en una computadora. El método es aplicable a cualquier tipo de problema, ya sea estocástico o determinístico

2.1.1 Historia del método de Monte Carlo

El método de Monte Carlo fue nombrado así por la ciudad de Montecarlo en Mónaco donde se juega “la ruleta”, el juego de azar que genera resultados aleatorios. Este método surge formalmente en el año 1944, sin embargo, ya existían prototipos y procesos anteriores que se basaban en los mismos principios.

El empleo del método de Monte Carlo para fines de investigación comenzó con el desarrollo de la bomba atómica en la Segunda Guerra Mundial en el Laboratorio Nacional de Los Álamos. Durante el desarrollo de este proyecto, los científicos Von Neumann y Ulam perfeccionaron la técnica y la aplicaron a problemas de cálculo de difusión de neutrones en un material. Alrededor de 1970, los desarrollos teóricos en complejidad computacional comienzan a proveer mayor precisión y relación para el empleo del método Monte Carlo [1]

Actualmente el método Monte Carlo a veces es usado para analizar problemas que no tienen un componente aleatorio explícito; en estos casos un parámetro determinista del problema se expresa como una distribución aleatoria y se simula dicha distribución. La simulación de Monte Carlo también fue creada para resolver integrales

que no se pueden resolver por métodos analíticos, para solucionar estas integrales se usaron números aleatorios. Posteriormente fue utilizado para cualquier esquema que emplee números aleatorios, usando variables aleatorias con distribuciones de probabilidad conocidas. En el caso del comportamiento de neutrones en un medio dado, el método de Monte Carlo es una alternativa más práctica que resolver las ecuaciones diferenciales que definen los fenómenos a los que los neutrones están sujetos.

2.1.2 Conceptos básicos del Método Monte Carlo

En el método Monte Carlo se combinan conceptos estadísticos como lo es el muestreo aleatorio, con la generación de números aleatorios y la automatización de los cálculos. Es un procedimiento matemático que consiste en la generación numérica de series mediante un muestreo aleatorio de las distribuciones de probabilidad.

2.1.2.1 Generación de números pseudo-aleatorios

La generación de una buena secuencia de números aleatorios es la base probabilística del método de Monte Carlo. Cada número aleatorio debe ser totalmente independiente de los otros números de la secuencia. Además, dos generadores aleatorios independientes deben proporcionar estadísticamente el mismo valor promedio de salida.

Comúnmente las secuencias de números se obtienen de algún algoritmo y se denominan números pseudo-aleatorios, mostrando así el origen determinístico. Este algoritmo ha de generar números de una manera realmente estocástica si se quieren simular correctamente los sucesos de interacción que sufren las partículas en la materia.

Esto hace que los generadores tengan que cumplir una serie de características [2]:

a) Buena distribución; se entiende que los números obtenidos estén uniformemente distribuidos en el intervalo en el que se obtienen $[0, 1]$. Si tomamos un subintervalo cualquiera, la fracción de números aleatorios que aparece respecto del total tiene que ser la misma para todo subintervalo de la misma amplitud.

b) Al ser generados mediante un algoritmo, siempre tienen un ciclo más o menos largo. En el caso de simulaciones en que se usa una gran cantidad de números aleatorios es importante que éstos no se repitan para que evitar las correlaciones.

c) Es importante que se pueda reproducir la sucesión de números usados. Si se repite la simulación en las mismas condiciones el resultado ha de ser el mismo.

2.1.2.2 Teorema del límite central [2]

El fundamento del método de Montecarlo hay que buscarlo en el teorema del límite central de la teoría de probabilidades, el cual afirma que si se consideran N variables aleatorias $\Psi_1, \Psi_2, \dots, \Psi_N$ independientes, cuyas leyes de probabilidad

coinciden y, por lo tanto, sus valores medios y su variancia cualquiera que sea el intervalo $[a', b']$, para grandes valores de N se cumple

$$p(a' \leq \rho_N \leq b') \approx \int_{a'}^{b'} p(x) dx$$

siendo $p(x)$ la densidad de probabilidad y

$$\rho_N = \Psi_1 + \Psi_2 + \dots + \Psi_N$$

Tomando este teorema como base, si se ha de calcular una magnitud m desconocida, tal que su valor medio y variancia sean respectivamente

$$\langle \Psi \rangle = m$$

$$V_\Psi = b^2$$

Considerando las N variables aleatorias independientes $\Psi_1, \Psi_2, \dots, \Psi_N$, con la misma distribución que Ψ . La ley de la variable suma ρ_N será aproximadamente normal, con un valor medio Nm y una variancia Nb^2 . Para un intervalo de confianza $\pm 3\sigma$ tenemos que;

$$P\left(m - \frac{3b}{\sqrt{N}} \leq \frac{\rho_N}{N} \leq m + \frac{3b}{\sqrt{N}}\right) = 0.997$$

$$P\left\{\left|\frac{1}{N} \sum_{i=1}^N \Psi_i - m\right| \leq \frac{3b}{\sqrt{N}}\right\}$$

Este resultado indica que el valor medio de los N valores resultantes del sorteo de la variable Ψ es una estimación del valor m , al tiempo que **este error es inversamente proporcional a N , lo cual refiere escoger una muestra de gran tamaño para conseguir un error pequeño [2].**

2.1.2.3 Técnicas de muestreo

El análisis elemental de la teoría de probabilidades nos va a permitir conocer más a fondo el funcionamiento interno de la técnica de Monte Carlo y poder interpretar los resultados obtenidos.

Función de densidad de probabilidad

En el método de Monte Carlo para análisis numéricos, se obtienen resultados estadísticos de determinadas variables físicas (energía, posición, etc.) sacando una muestra apropiada de la distribución de probabilidad. Para ello escogemos un conjunto de muestras aleatorias x_i distribuidas de acuerdo con la función de densidad de probabilidad (pdf por sus siglas en inglés: *probability density function*) denominada $p(x)$, tal como se muestra en la figura 2.1. Así, $p(x)dx$ nos indica la probabilidad de que

cualquiera x_i quede incluida entre x y $x+dx$. En general, $p(x)$ debe cumplir algunos requisitos:

- Se define positiva ($p(x) \geq 0$).
- Es integrable y normalizada

$$\int_a^b p(x)dx = 1.$$

a y b son números reales que cumplen

$$-\infty < a < b < \infty.$$

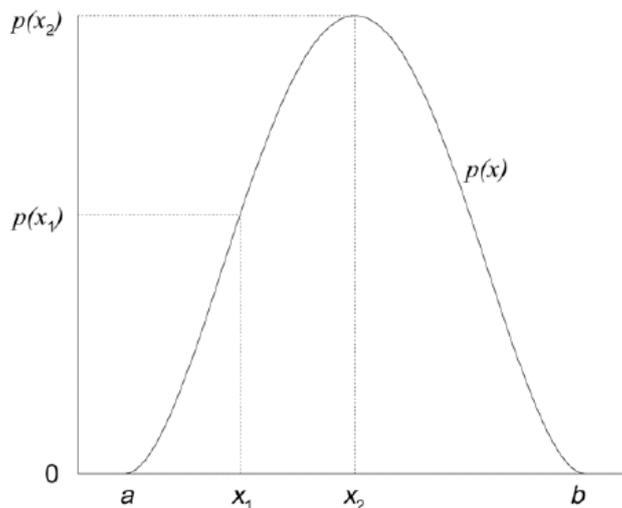


Figura 2.1. Ejemplo de función de probabilidad, $p(x)$. [2]

Función de distribución

Asociada a cada función densidad de probabilidad, $p(x)$, podemos definir la función de distribución, denominada $c(x)$, como la suma de las probabilidades de cada x_i perteneciente al interior de cada intervalo infinitesimal entre a y x ,

$$c(x) = \int_a^x p(x')dx'$$

Como la probabilidad de eventos excluyentes es aditiva, $c(x)$ se interpreta como la probabilidad que cualquier x_i dado, sea menor o igual que x . La función $c(x)$ es monótona creciente en x , ya que $p(x) \geq 0$ para todo x . Como se muestra en la figura 2.2, la probabilidad integrada, a lo largo de todos los posibles resultados es la unidad, $c(b)=1$,

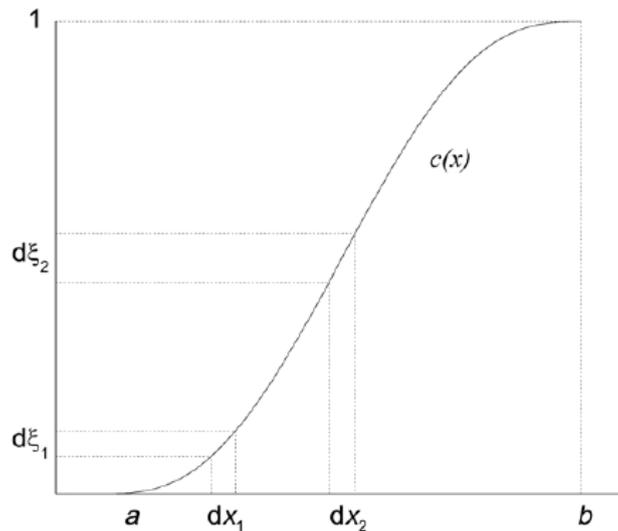


Figura 2.2. Función de distribución, $c(x)$, obtenida de la integración de la función densidad de probabilidad. [2]

2.1.3 Descripción del algoritmo.

El algoritmo de Simulación Monte Carlo fundamentado en la generación de números aleatorios, se basa en las distribuciones acumuladas de frecuencias:

- Determinar la o las variables aleatorias y sus funciones de distribución de probabilidad.
- Generar un número aleatorio uniformemente distribuido en $(0,1)$.
- Determinar el valor de la variable aleatoria para el número aleatorio generado de acuerdo a las pdf que tengamos.
- Iterar los dos pasos anteriores tantas veces como muestras necesitamos.
- Calcular la media, desviación estándar, error y realizar el histograma.
- Analizar resultados para distintos tamaños de muestra.

Otra opción para trabajar con el método Monte Carlo, cuando la variable aleatoria no es directamente el resultado de la simulación o tenemos relaciones entre variables es la siguiente:

- Diseñar el modelo lógico de decisión.
- Especificar las distribuciones de probabilidad para las variables aleatorias relevantes.
- Incluir posibles dependencias entre variables.

- Muestrear valores de las variables aleatorias.
- Calcular el resultado del modelo según los valores del muestreo (iteración) y registrar el resultado.
- Repetir el proceso hasta tener una muestra estadísticamente representativa.
- Obtener la distribución de frecuencias del resultado de las iteraciones.
- Calcular media, desviación.
- Analizar los resultados.

Las principales características a tener en cuenta para la implementación o utilización del algoritmo son:

- El sistema debe ser descrito por una o más funciones de distribución de probabilidad (pdf).
- El proceso de generación de los números aleatorios es importante para evitar que se produzca correlación entre los valores muestrales.
- Establecer límites y reglas de muestreo para las pdf; conocemos qué valores pueden adoptar las variables.
- Definir cuándo un valor aleatorio tiene o no sentido para el modelo a simular.
- Estimar con qué error trabajamos, cuánto error podemos aceptar para que una corrida sea válida.
- Paralelización y vectorización: En aplicaciones con muchas variables se estudia trabajar con varios procesadores paralelos para realizar la simulación. [3]

2.1.4 Método de Monte Carlo para transporte de partículas.

Todos los procesos que involucran el transporte de partículas tienen naturaleza estocástica, 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. Sin embargo, las distribuciones de probabilidad que describen el comportamiento del sistema que queremos estudiar son conocidas. El método de Monte Carlo construye un modelo estocástico, que basándose en las funciones de densidad de probabilidad modela secuencialmente eventos individuales de una variable aleatoria. Teóricamente se siguen todos los eventos o interacciones que sufre cada partícula desde su origen hasta que alcanza una condición terminal, ya sea por absorción, escape, energía de corte, etc. Lo mismo se aplica para todas las partículas creadas en el proceso.

El método de Monte Carlo es considerado el más preciso para simular el transporte de partículas y en particular de neutrones en un medio, por el alto grado de detalle con el que considera la distribución energética de los neutrones, las interacciones de los neutrones con los diferentes materiales, la geometría y la trayectoria que siguen los neutrones en el medio.

Para cualquier evento, el método de Monte Carlo genera un número aleatorio fundamentándose en las funciones de distribución de probabilidad, que definirá el tipo de interacción y otros parámetros. Posteriormente, se calcula el valor esperado de todos los eventos simulados. El valor esperado de una o varias variables aleatorias es equivalente al valor de una cantidad física del sistema estudiado.

En cada etapa de la vida de un neutrón, su comportamiento futuro se puede determinar con ayuda de varias pdf's. Por ejemplo, la localización de una interacción se determina con la función $p(x) = \sum_i \exp(-\Sigma_i x)$; el tipo de interacción se determina por tasas de secciones eficaces (por ejemplo σ_s / σ_t es la probabilidad de que una interacción sea una dispersión elástica); el ángulo de dispersión del neutrón (si es dispersado) se obtiene de la sección eficaz diferencial ($\sigma_s(\mathcal{G})$); y así sucesivamente [4].

2.1.4.1 Trayectoria de las partículas

La historia de una partícula comienza calculando mediante procesos aleatorios la probabilidad de que se cree un neutrón, de manera análoga se obtiene la energía, la posición y la dirección inicial de la trayectoria de dicho neutrón, tomando en cuenta que éstas son independientes entre sí. Posteriormente, se simula la distancia libre que recorrerá antes de interactuar y el tipo de colisión que sufrirá. La energía y dirección de las partículas dispersadas y de los neutrones secundarios son variables aleatorias que también se calculan en la simulación. Este proceso continúa para cada uno de los neutrones que se van generando hasta que se alcanza una condición terminal. Siendo posible obtener con este método, paso por paso, el “trayecto aleatorio” de las partículas a través del medio, obteniendo así la distribución exacta de los neutrones.

Considérese la trayectoria de un caso particular de partícula que sea un neutrón que viaja a través de un medio homogéneo, dado que la partícula es dispersada frecuentemente, su trayectoria será en zig-zag, tal como se muestra en la Figura 2.3 [4]. Suponiendo que la partícula se origina en la posición A, con dirección y energía conocidas, al inicio tendrá un “vuelo libre” hasta que tiene una colisión con algún núcleo atómico del medio, que podría resultar en la absorción de la partícula y la inmediata terminación de su historia. Pero, suponiendo que la interacción es de dispersión, la partícula sufre un cambio en energía y en dirección. Tanto el cambio en energía como el cambio en dirección son procesos estadísticos (es decir, no hay una única energía y dirección después de la dispersión), para los cuales hay una distribución de probabilidad. En la Figura 2.3 se observa cómo después de la primera dispersión, la

misma partícula tiene otro “vuelo libre” y posteriormente colisiona, y así sucesivamente.

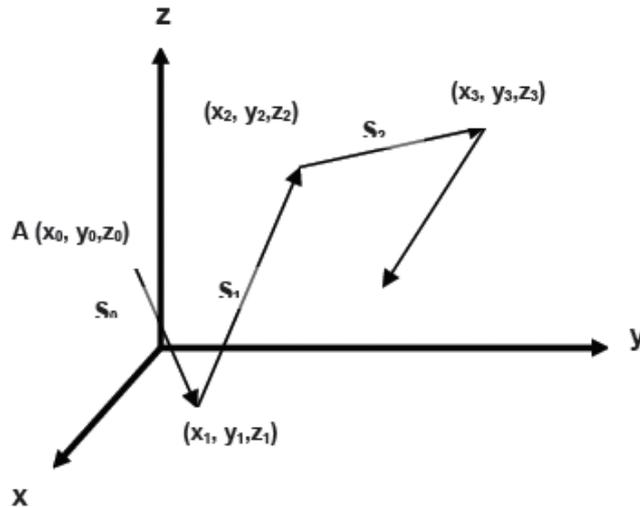
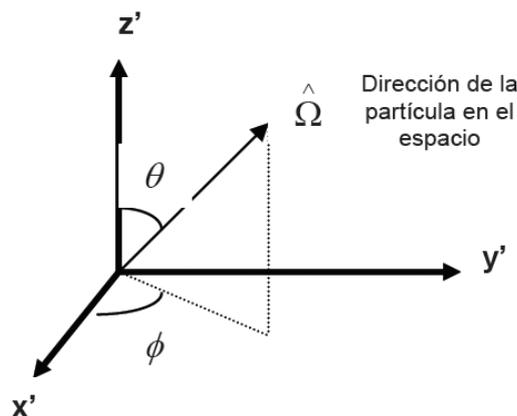


Figura 2.3. Trayectoria aleatoria de una partícula a través de un medio. [4]

Para seguir la trayectoria del viaje de la partícula es necesario conocer sus coordenadas espaciales (x_0, y_0, z_0), las coordenadas esféricas de su dirección (θ, ϕ) y su energía. Estas variables son suficientes para definir el estado α , de la partícula, donde:

$$\alpha \equiv \alpha(x, y, z; E; \theta, \phi) \tag{1}$$

El sistema de coordenadas esféricas que definen la dirección de la partícula se ilustra en la Figura 2.4



Nota: el sistema de coordenadas ortogonal (x', y', z') es paralelo al sistema (x, y, z) de la figura 2.3 básico de referencia.

Figura 2.4. Dirección de la Partícula en Coordenadas Esféricas (θ, ϕ). [4]

La trayectoria de una partícula se puede construir de colisión a colisión como una sucesión de estados $\alpha_0, \alpha_1, \dots, \alpha_n$, donde el i -ésimo estado es:

$$\alpha_i \equiv \alpha_i(x_i, y_i, z_i; E_i; \theta_i, \phi_i) \quad (2)$$

Por lo tanto, en el i -ésimo estado, una partícula tiene las coordenadas espaciales del punto en donde ocurre la colisión y tanto la energía y dirección de la partícula después de dicha colisión. A excepción del estado inicial, cada estado sucesivo está en función del estado previo y de las leyes de dispersión, obedecidas por la partícula en el material de interés. Por ejemplo, el estado α_0 comienza con condiciones iniciales y por muestreo aleatorio de las distribuciones de probabilidad correspondientes a cada tipo de interacción se determina el estado α_1 , y así sucesivamente, construyendo de esta forma la historia de la vida de cada partícula.

Obviamente, se requiere de procedimientos matemáticos para seleccionar la posición de la siguiente colisión, así como la nueva energía y dirección de la partícula, si es que sobrevive a una colisión.

Considérese un partícula que ha sufrido su i -ésima colisión (una dispersión), las coordenadas del punto de su siguiente colisión se obtienen de la siguiente manera: si s es la longitud de la trayectoria que la partícula viaja hasta su próxima colisión, la probabilidad de que la partícula viaje la distancia s sin tener una interacción es $e^{-\Sigma_t s}$.

La probabilidad de que la partícula tendrá una interacción en el intervalo ds es $\Sigma_t ds$, por lo tanto, la probabilidad de que la partícula tendrá una interacción entre s y $s+ds$ es:

$$\Sigma_t e^{-\Sigma_t s} ds \quad (3)$$

donde Σ_t es la sección eficaz macroscópica total.

Para completar la simulación de todo sistema físico, es necesario saber la orientación de una partícula después de que sufre una interacción. Esto resulta de establecer un procedimiento para seleccionar aleatoriamente un valor de s (s_i) de la función de probabilidad dada por la ecuación (3). Una vez determinado el valor de s_i , las coordenadas de la siguiente colisión se obtienen de:

$$\begin{aligned} x_{i+1} &= x_i + s_i (\cos \theta_i \cos \phi_i) \\ y_{i+1} &= y_i + s_i (\cos \theta_i \sin \phi_i) \\ z_{i+1} &= z_i + s_i (\sin \theta_i) \end{aligned} \quad (4)$$

De forma similar, la energía de la partícula después de la dispersión se obtiene muestreando la función de probabilidad apropiada.

Por ejemplo en el caso de una interacción que sea una dispersión, los ángulos "locales" después de la dispersión (θ_0, ϕ_0) pueden determinarse. θ_0 es el ángulo de

deflección y ϕ es el ángulo azimutal (ver Figura 2.5). Se obtiene un número aleatorio ϕ que esté uniformemente distribuido entre $[0, 2\pi]$. Finalmente, con los ángulos “locales” (θ_0, ϕ) se puede determinar la nueva dirección de la partícula, (θ_i, ϕ_i) .

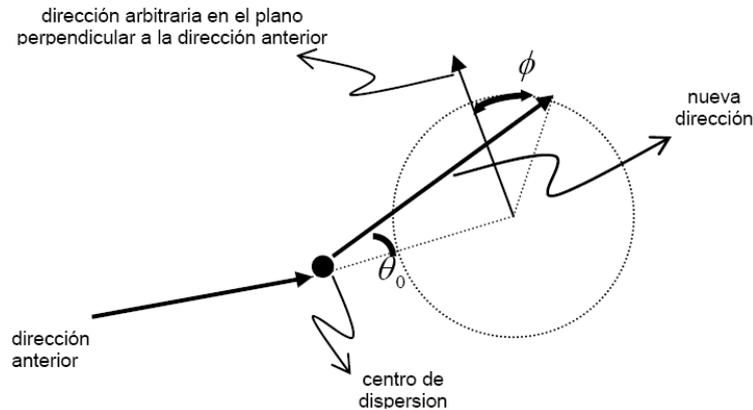


Figura 2.5. Ángulos locales de dispersión de la partícula. [4]

Otro ejemplo es la función de distribución de Maxwell-Boltzmann, para la distribución de energía de los neutrones en equilibrio térmico con los átomos del medio, la cual está dada por:

$$n(E) = \frac{2\pi N}{(\pi kT)^{3/2}} E^{1/2} e^{(-E/kT)}$$

La función $p(x)$ puede ser continua como la definida anteriormente, o puede ser discreta cuando hay un número finito de posibles resultados. Por ejemplo, el espectro de energía de los neutrones de fisión $\chi(E)$, es una función de densidad de probabilidad continua. Un ejemplo de una función discreta se puede construir notando que σ_a / σ_t y σ_s / σ_t son las probabilidades de absorción y dispersión en un material, respectivamente, con σ_t la sección eficaz microscópica total.

2.1.4.2 Cálculo de la criticidad

Para los cálculos de criticidad se estima el valor del factor de multiplicación efectiva de neutrones (k_{eff}). En estos cálculos, un grupo de historias de neutrones se refiere a un ciclo de k_{eff} (en teoría de reactores se define como una generación de neutrones), donde el factor de multiplicación está dado por la razón del número de neutrones generados en los eventos de fisión que se presentan en el ciclo, entre el número de neutrones cuyas historias son evaluadas en este ciclo (es decir, el número de neutrones al inicio de la generación). El valor esperado del factor de multiplicación se estima promediando sobre los eventos en el ciclo de k_{eff} . De la misma forma, también se puede obtener el valor esperado de la probabilidad de fuga o la fracción de eventos

que llevan a captura. El error relativo en la estimación de k_{eff} usualmente disminuye conforme el número de ciclos de k_{eff} aumenta, por lo que se necesita un gran número de ciclos para obtener una buena estimación. El factor de multiplicación se estima mediante la siguiente expresión:

$$\bar{k} = \frac{1}{N} \sum_{i=1}^N k_i$$

donde \bar{k} es el factor de multiplicación estimado para el sistema de interés, y k_i es el factor de multiplicación estimado en el i -ésimo ciclo.

La aproximación de Monte Carlo se puede resumir como sigue: una secuencia de números aleatorios r_i ($0 < r_i < 1$) se usa para producir una distribución aleatoria de cantidades que simulan el problema de interés. Un ejemplo de cómo funciona el método de Monte Carlo para obtener el factor de multiplicación en un medio multiplicativo es el siguiente:

1. Para el primer ciclo del cálculo del factor de multiplicación de neutrones (k_{eff}), determinar la posición inicial del neutrón.
2. Utilizar un número aleatorio para seleccionar la energía del neutrón.
3. Usar el siguiente número aleatorio para determinar la dirección coseno del neutrón.
4. Determinar la localización de la siguiente colisión con el siguiente número aleatorio (la distancia que recorra el neutrón depende de la sección eficaz del material).
5. Verificar la nueva posición del neutrón para determinar si ha escapado del sistema; si esto ocurre, agregar un uno a los escapes totales y regresar al paso 1 empezando otra historia o ciclo; de otra forma, continuar.
6. Determinar qué tipo de interacción ocurrió en la nueva posición basado en el siguiente número aleatorio. Cada tipo de interacción tiene asociada una sección eficaz (bibliotecas del código) que determina su probabilidad de ocurrencia:
 - a) Si la interacción es una dispersión, determinar la energía del neutrón después de la dispersión utilizando el siguiente número aleatorio. Continuar en el paso 3 para saber la trayectoria del neutrón dispersado.
 - b) Si la interacción es una absorción, regresar al paso 1 y empezar otro ciclo con un nuevo neutrón.
 - c) Si la interacción es una fisión, determinar cuántos neutrones se producen en este evento de fisión (utilizando las bibliotecas del código) y determinar el número total de neutrones que se han producido en el ciclo. También determinar la posición de los neutrones que se produjeron en la fisión para que sirvan como inicio de otro ciclo (esto reemplaza al paso 1 en futuros ciclos de k_{eff}).
7. Cuando se han completado las historias que se han solicitado (las suficientes para una estadística adecuada), evaluar la k_{eff} dividiendo el número de nuevos neutrones creados en este ciclo por el número de historias evaluadas en el ciclo.

El proceso se repite para tantos ciclos como sean requeridos para obtener una estadística apropiada.

Error Relativo.

Existe una diferencia entre el valor esperado verdadero de una función f y el valor de la misma función cuando sólo se toma un muestreo de N variables estadísticas independientes (número de partículas a simular) [5]. Para saber si la simulación arroja un resultado confiable, es importante tomar en cuenta el error relativo R . Este dato representa la precisión estadística y es calculado por el programa a partir cada historia aleatoria.

El error esperado en $\langle f \rangle_N$ se estima con la desviación estándar

$$\delta_N^2 = \left\langle (f(x_1, x_2, \dots, x_N) - f(x_1, x_2, \dots, \infty))^2 \right\rangle$$

Sin embargo como no se tiene un número infinito de variables, el error se estima considerando cada selección aleatoria como un estimador independiente de $\langle f \rangle_N$, así,

$$\delta_N^2 = \frac{1}{N} \sum_{i=1}^N [f(x_i) - \langle f \rangle_N]^2$$

La suma de las desviaciones estándar, ε_{sum} , es N veces el valor de δ_N . Por consiguiente:

$$\varepsilon_{sum} = \sqrt{N} \delta_N$$

El error relativo es la media de ε_{sum} , de tal manera que:

$$R = \frac{\varepsilon_{sum}}{N} = \frac{\delta_N}{\sqrt{N}}$$

Es importante recalcar que R es proporcional a $1/\sqrt{N}$, es decir, mientras mayor sea el número de partículas simuladas se podrá obtener un resultado más preciso. Para reducir el error relativo en una simulación se puede aumentar N o reducir el valor de δ_N . Sin embargo, el presupuesto limita el incremento que se puede tener en N por el tiempo de computadora que requiere. Por esta razón, existen técnicas de reducción de varianza en el código MCNP que se basan en disminuir el valor de δ_N [5]. Al realizar un gran número de historias al azar, aumentará la precisión del valor promedio o de otras cantidades de interés.

2.2 EL PROGRAMA DE CÓMPUTO MCNPX

Actualmente el código MCNPX, desarrollado por el Laboratorio Nacional Los Álamos, es el programa de computadora más completo basado en el método de Monte Carlo.

Para realizar una simulación se debe crear un archivo de entrada. Este archivo contiene de manera estructurada información sobre la geometría, los materiales utilizados, las secciones eficaces a utilizar, la localización, características y tipo de la fuente (electrones, fotones o neutrones), los *tallies* (conteos) y cualquier técnica de reducción de varianza.

Los *tallies* son instrucciones que le dicen al programa el tipo de datos que se desea calcular; por ejemplo, la corriente y el flujo de algún tipo de partículas resultantes, o la deposición y distribución de energía.

El programa MCNPX leerá las instrucciones del archivo de entrada, realizará la simulación y creará un nuevo archivo de salida. El archivo de salida incluirá los resultados generados por cada *tally*, los errores producidos y algunas tablas que resumen el proceso de simulación.

2.2.1 Formato del archivo de entrada

El archivo de entrada es el medio por el cual el usuario introduce el problema que desea resolver. Este archivo consta de tres tarjetas separadas entre sí por una línea en blanco como se observa en la siguiente figura.

```
Nombre de la tarjeta
Tarjeta de celdas
.....
.....
Línea en blanco
Tarjeta de superficies
.....
.....
Línea en blanco
Tarjeta de datos
.....
.....
Línea en blanco terminal (opcional)
```

Figura 2.6. Formato de archivo de entrada de MCNPX. [6]

2.2.1.1 Tarjeta de definición de celdas (*Cell Cards*):

En esta primera parte del archivo se definen las celdas. Una celda es un volumen del espacio definido por las superficies que se implementarán en la segunda parte del archivo. Para definir las se usa la lógica booleana. Todo el espacio debe estar definido por una única celda. Las celdas son usadas para definir la forma y el material contenido en un espacio físico. El formato específico de la tarjeta de celdas es:

j m d geom params

j = Número de celda

m = Número de material (0 si es un espacio vacío)

d = Densidad del material de la celda:

 No introducir si la celda es un espacio vacío

 Positiva = densidad atómica (atom/cm³)

 Negativa = densidad másica (g/cm³)

geom = Lista de todas las superficies y operaciones booleanas que limitan y definen la celda.

params = Especificaciones opcionales de los parámetros de la celda.

Por ejemplo se tiene la siguiente tarjeta de definición de celdas:

```
C    Cell Card
3    2    1.23e-3    -2
```

Donde:

- El número 3 indica el número de la celda, éste es único y debe ser consecutivo para todas las celdas.
- El número 2 indica el material del que está formada la celda, aunque el material se definirá en la tarjeta de datos. Varias celdas pueden tener este número igual al tratarse de celdas con el mismo material.
- El tercer parámetro indica que la densidad del material es de 1.23E-3 [átomos/barn-cm], dichas unidades se deben a que la cantidad 1.23E-3 es positiva. Este parámetro se coloca aquí y no en la definición de materiales para poder representar diferentes estados de un mismo material.
- El parámetro -2 indica que la celda está delimitada por la superficie 2, misma que en la tarjeta de superficies será descrita. El signo indica de qué lado de la superficie está la celda.

2.2.1.2· Tarjeta de definición de las superficies. (*Surface Cards*):

En esta tarjeta se definen las superficies que se usarán para crear las celdas. Se usan sus ecuaciones cartesianas para definir las. El formato de la tarjeta de superficie es:

j a list

- j = Número de superficie
- a = Nemónico de la superficie (plano, esfera, cilindro, etc)
- list = Números que describen la superficie (dimensiones, radios, etc)

Por ejemplo se tiene la siguiente tarjeta de superficies:

2 CZ 20.00

Donde:

- 2 es el número de la superficie
- CZ indica que el tipo de superficie es un cilindro infinito centrado en el eje Z
- 20.00 indica el radio del cilindro con unidades en cm.

2.2.1.3 Tarjeta de datos (*Data Cards*):

Esta es la zona más importante. En ella se definen los parámetros de la fuente de partículas, materiales de las celdas, modo del problema (tipo de partículas de las que se desea realizar el transporte), duración del problema definido por tiempo de cálculo o por número de partículas simuladas y modo en que MCNPX mostrará los resultados.

Para realizar cálculos de criticidad es necesario que la tarjeta de datos contenga la línea de comando kcode. El formato de la línea de comando Kcode es el siguiente:

kcode nsrck rkk ikz kct

kcode = Nombre de la tarjeta para calcular la criticidad.

nsrck = Número de neutrones por ciclo

rkk = Aproximación inicial para K_{eff}

ikz = Número de ciclos ignorados antes de la acumulación de datos

kct = Número total de ciclos a correr

Por ejemplo para indicar que un problema se correrá con 1000 neutrones por ciclo, un factor de criticidad efectivo igual a 1, con 15 ciclos ignorados, y un total de 115 ciclos; se tendrá la siguiente línea del comando Kcode de la tarjeta de datos:

Kcode 1000 1.0 15 115

También es necesario definir las coordenadas del punto de ubicación de la fuente para efectuar la fisión, esto es ejecutado a través de la línea de comando `ksrc`, cuyo formato es el siguiente:

```
ksrc  x1 y1 z1  x2 y2 z2  ...  xn yn zn
```

`ksrc` = Nombre de la tarjeta que posiciona la fuente inicial de fisión

`xk yk zk` =Coordenadas de localización del punto de la fuente.

Así, para indicar la ubicación de dos fuentes de fisión, una en el origen (0,0,0) y otra en el punto (2,2,3) se tiene:

```
ksrc 0 0 0    2 2 3
```

Para definir los materiales utilizados se emplea la siguiente línea de comando

```
mn zaid1 fraction1 zaid2 fraction2 ...
```

Donde:

mn= Nombre de la tarjeta de material, seguida del número de material citado en la tarjeta de celdas.

Zaid= Número y masa atómica del isótopo.

Fraction= Fracción del isótopo.

(+) Fracción atómica

(-) Fracción en peso.

Por ejemplo, se tiene la siguiente tarjeta de datos con el comando para definir el material:

```
m2 8016 1
```

La cual indica que el material 2 es oxígeno, ya que el `zaid` del oxígeno es 8016. El número 1 hace referencia a que la fracción atómica del oxígeno en el total del combustible es la unidad.

2.2.2 Descripción del archivo de salida.

El archivo de salida será un archivo que tendrá el nombre que se le haya indicado en la línea de comandos de ejecución de MCNPX. En primer lugar aparece una copia exacta del archivo de entrada, intercalando entre líneas los diferentes *warnings* y errores que se hayan podido ocasionar.

Posteriormente aparecen los cálculos de geometría (volúmenes de celdas, densidades, etc.). Estos datos se realizan antes de simular la primera partícula. A continuación aparecen los balances de partículas, la forma de su creación y su pérdida y, al final de éstos, MCNPX muestra el tiempo que ha tardado en realizar la simulación, y el número de simulaciones que ha realizado por minuto. Este dato es muy útil para tener una idea de la magnitud del problema que se va a realizar, de si se necesitará mucho tiempo para obtener un error relativo aceptable, o bien, se ha de hacer uso de algún tipo de reducción de varianza.

Consecutivamente aparece la actividad de cada tipo de partícula por celda, el número de esas partículas que han entrado, que se han generado y que han colisionado en cada celda. Y, al final del archivo, se muestra el resumen de los cálculos realizados, como los *tallies*, con sus respectivos chequeos estadísticos.

Esquema de uso recomendado.

El uso del programa de cómputo MCNPX sugiere una metodología para ahorrar tiempo para la solución de un problema. El siguiente es un esquema de posible uso.

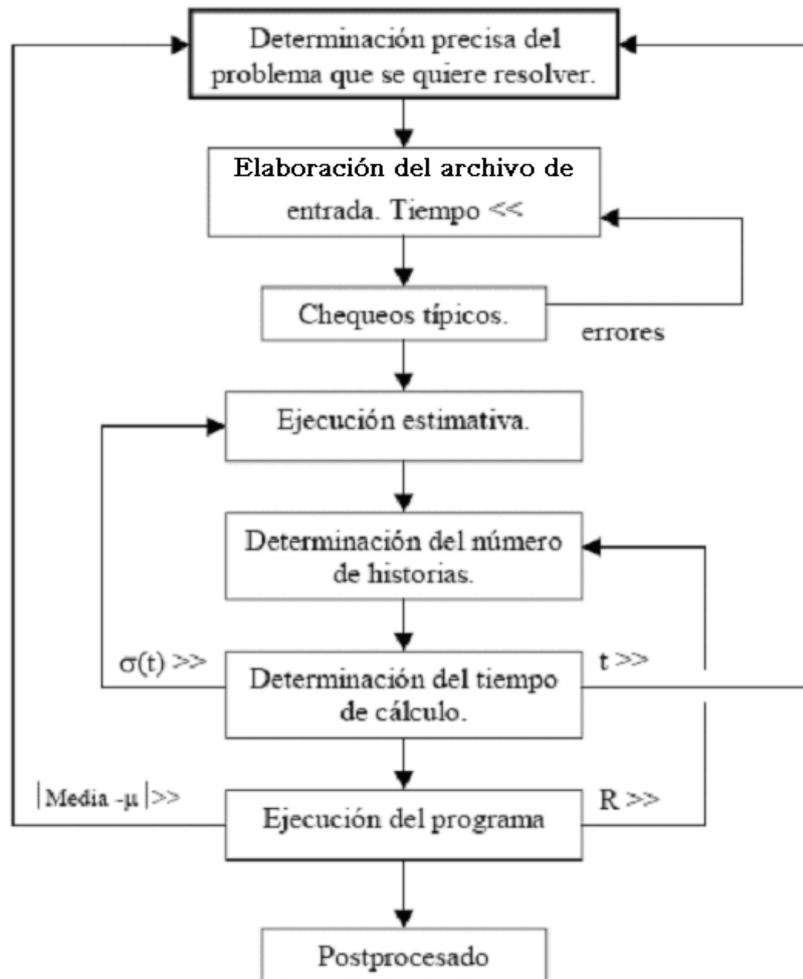


Figura2.7. Esquema de uso recomendado. [7]

Donde el número de iteraciones y los criterios usados para tomar las distintas decisiones dependerán del tipo de problema y del nivel de exigencia a implementar.