Informe Parte B1: Caminata Aleatoria Autoevitante (SAW)

Física Computacional II - Grupo [Número/Nombre del Grupo]

10 de julio de 2025

Índice

1. Introducción

Una Caminata Aleatoria Autoevitante (Self-Avoiding Walk, SAW) es una trayectoria en una retícula que no visita el mismo sitio más de una vez. Las SAWs son modelos fundamentales en física estadística y química de polímeros, donde representan la configuración espacial de una cadena polimérica lineal en un buen solvente.

El objetivo de esta parte del proyecto es implementar una simulación de SAW en una retícula cuadrada bidimensional usando un algoritmo de crecimiento simple, medir sus propiedades estadísticas y analizar el comportamiento del algoritmo.

La investigación teórica sobre números aleatorios, generadores y caminatas aleatorias se encuentra en el documento principal del proyecto.

2. Metodología

2.1. Algoritmo de Crecimiento Simple

El algoritmo implementado en la clase `SAWSimulador` sigue estos pasos:

1. Inicializar la caminata en el origen (0,0) de la retícula
2. Marcar el origen como visitado en un `std::set<Point2D>`
3. Para cada paso hasta N_{max}:
   - Evaluar los 4 vecinos del sitio actual (arriba, abajo, izquierda, derecha)
   - Filtrar solo los vecinos no visitados
   - Si no hay vecinos disponibles, terminar la caminata (atascada)
   - Si hay vecinos disponibles, elegir uno aleatoriamente
   - Moverse al sitio elegido y marcarlo como visitado
4. Calcular la distancia cuadrática extremo-a-extremo: $R^2 = (x_f - x_0)^2 + (y_f - y_0)^2$

2.2. Implementación Computacional

La implementación utiliza:

- `std::mt19937`: Generador Mersenne Twister para números pseudoaleatorios
- `std::uniform_int_distribution`: Distribución uniforme para selección de vecinos
- `std::set<Point2D>`: Estructura eficiente para verificar sitios visitados
- `std::vector<Point2D>`: Almacenamiento de la trayectoria completa

3. Resultados Experimentales

3.1. Parámetros de Simulación

Las simulaciones se ejecutaron con los siguientes parámetros por defecto:

- $N_{max\_pasos} = 80$ pasos máximos
- $N_{simulaciones} = 20000$ intentos por configuración
- Semilla aleatoria basada en tiempo del sistema

3.2. Trayectoria de Ejemplo

La Figura 1 muestra una trayectoria típica generada por el simulador.

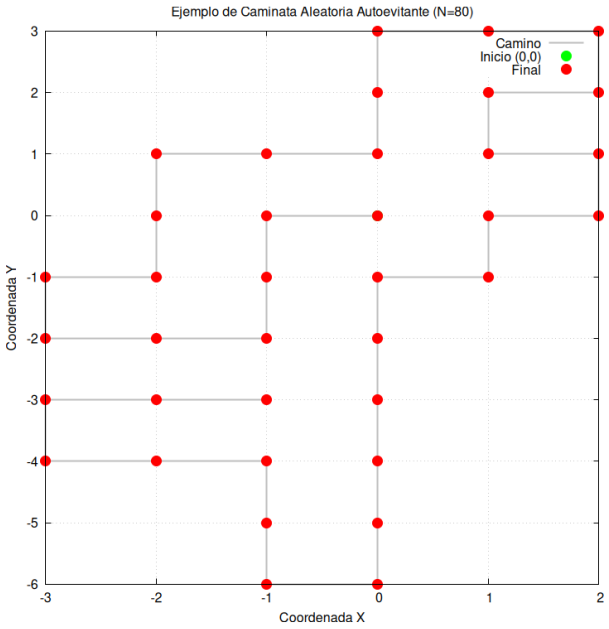

Figura 1: Ejemplo de Caminata Aleatoria Autoevitante para $N_{max} = 80$ pasos. La trayectoria se muestra desde el origen (punto inicial) hasta el punto final, evitando cualquier autointersección.

3.3. Distribución de Longitudes

El histograma de longitudes alcanzadas (Figura **??**) revela la característica fundamental del algoritmo simple: la mayoría de caminatas se atascan antes de alcanzar N_{max}.

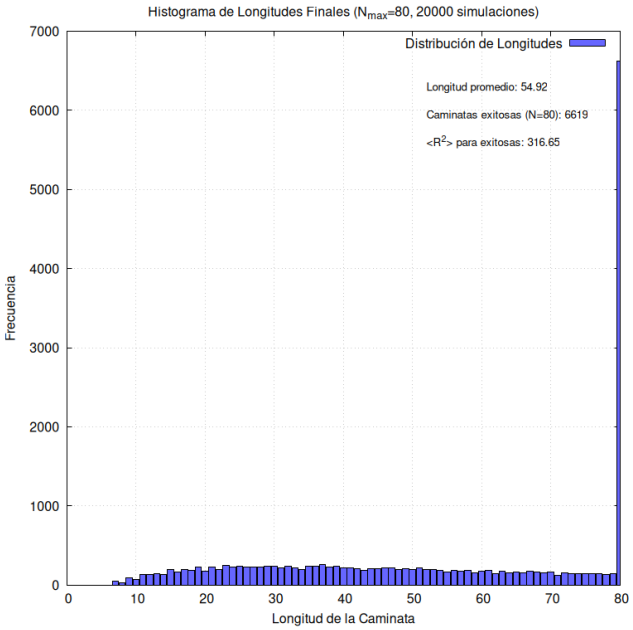

Figura 2: Histograma de longitudes alcanzadas en 20000 simulaciones con $N_{max} = 80$. Se observa que la mayoría de caminatas se atascan en longitudes menores, con muy pocas alcanzando la longitud objetivo.

3.4. Análisis de Eficiencia del Algoritmo

Los resultados muestran que el algoritmo de crecimiento simple sufre de:

- **Atrición severa**: La fracción de caminatas exitosas (que alcanzan N_{max}) decrece exponencialmente con N_{max}
- **Sesgo hacia caminatas cortas**: El promedio está dominado por caminatas que se atascan temprano
- **Escalamiento computacional pobre**: Para obtener estadísticas significativas de caminatas largas, se requiere un número exponencialmente creciente de intentos

3.5. Propiedades Físicas Medidas

Para las caminatas exitosas, se calculó:

1. **Longitud promedio**: $\langle L \rangle$ sobre todas las caminatas (exitosas y atascadas)
2. **Fracción de éxito**: $P_{xito} = N_{exitosas}/N_{total}$
3. **Desplazamiento cuadrático medio**: $\langle R^2 \rangle$ para caminatas exitosas

El comportamiento esperado para SAWs largas es:

$$\langle R^2 \rangle \sim AN^{2\nu} \tag{1}$$

donde $\nu \approx 0{,}75$ es el exponente crítico de Flory para SAWs en 2D.

4. Limitaciones del Algoritmo Simple

4.1. Problemas Fundamentales

1. **Atascamiento exponencial**: La probabilidad de completar una SAW de longitud N decrece aproximadamente como μ^{-N} donde $\mu \approx 2{,}64$ es la constante conectiva para la retícula cuadrada
2. **Tiempo de CPU creciente**: Para longitudes moderadas ($N > 100$), el tiempo requerido para obtener estadísticas confiables se vuelve prohibitivo
3. **Sesgo estadístico**: Las configuraciones más .ªbiertas.están sobrerrepresentadas comparadas con configuraciones más compactas

4.2. Algoritmos Más Eficientes

Para estudios serios de SAWs, se requieren algoritmos más sofisticados:

- **Método de Rosenbluth-Rosenbluth**: Usa pesos estadísticos para corregir el sesgo
- **Algoritmos de Pivote**: Operan sobre SAWs existentes mediante transformaciones de simetría
- **PERM (Pruning-Enriched Rosenbluth Method)**: Combina poda y enriquecimiento para mejorar eficiencia

5. Archivos de Salida

El programa genera automáticamente:

- `results/saw_camino_ejemplo_N80.dat`: Coordenadas de una trayectoria ejemplo
- `results/saw_resultados_N80_sim20000.dat`: Datos estadísticos y longitudes de todas las simulaciones
- Gráficas PNG generadas por scripts de Gnuplot

6. Conclusiones

6.1. Logros del Trabajo

1. Se implementó exitosamente un simulador de SAW usando el algoritmo de crecimiento simple
2. Se verificó el comportamiento estadístico esperado: atrición exponencial y sesgo hacia caminatas cortas
3. Se generaron visualizaciones que ilustran claramente las propiedades geométricas de las SAWs
4. Se estableció una base sólida para comparar con algoritmos más avanzados

6.2. Limitaciones Observadas

1. El algoritmo simple es inviable para $N_{max} > 200$ debido a la atrición severa
2. La fracción de caminatas exitosas es demasiado baja para obtener estadísticas confiables de $\langle R^2 \rangle$
3. El tiempo computacional escala muy pobremente con el tamaño del problema

6.3. Trabajo Futuro

1. Implementar el algoritmo de Rosenbluth-Rosenbluth para corregir sesgos estadísticos
2. Desarrollar algoritmos de pivote para generar SAWs de longitudes arbitrarias
3. Estudiar el exponente crítico ν con mayor precisión estadística
4. Extender a retículas 3D y otros tipos de retícula

6.4. Relevancia Física

Este trabajo proporciona una introducción práctica a los desafíos computacionales en física estadística de polímeros. Los algoritmos de SAW son fundamentales para entender:

- Conformaciones de cadenas poliméricas en solución
- Transiciones de fase en sistemas de polímeros
- Fenómenos críticos en sistemas de dimensión baja
- Métodos de Monte Carlo avanzados

El simulador desarrollado, aunque limitado, demuestra los conceptos físicos esenciales y establece la base para estudios más avanzados en física computacional de polímeros.