



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE CIENCIAS EXACTAS
DEPARTAMENTO DE MATEMATICA

Trabajo de Iniciación a la Investigación

Nuevos métodos de Monte Carlo - Simulated Annealing con
aplicaciones en Ciencias de Materiales

Federico Costa

Director: Dr. Cristián Huck-Iriart

Veedora académica: Dra. Alejandra Vahnovan

2019

Índice

| | |
|--|-----------|
| Agradecimientos | 3 |
| 1. Resumen | 4 |
| 2. Introducción | 5 |
| 3. Objetivos | 8 |
| 4. Métodos de Monte Carlo | 9 |
| 4.1. Exploración independiente | 9 |
| 4.1.1. Búsqueda Aleatoria para todo el espacio de parámetros | 10 |
| 4.1.2. Búsqueda Aleatoria sólo para los parámetros no lineales | 10 |
| 4.2. Exploración dependiente: Simulated Annealing | 11 |
| 5. Resultados | 17 |
| 5.1. SAXS | 17 |
| 5.2. Datos experimentales | 19 |
| 5.3. Implementación de los métodos | 21 |
| 5.3.1. Implementación del método 1 | 21 |
| 5.3.2. Implementación del método 1_1 | 22 |
| 5.3.3. Implementación del método 2 | 22 |
| 5.3.4. Implementación del método 3 | 23 |
| 5.4. Comparación de los métodos | 24 |
| 5.5. Resultados del Simulated Annealing | 26 |
| 6. Conclusiones | 33 |
| 7. Perspectivas | 34 |
| 8. Anexo: definiciones | 35 |

Agradecimientos

- A Cristián y Alejandra, así como también a Leandro y Maximiliano por guiarme y por todo el tiempo y la paciencia que dedicaron para explicarme (que fue mucho).
- A Pedro y Victoria, por aceptar ser jurado de mi trabajo con todo lo que ello implica y aportarme comentarios y sugerencias. Particularmente a Pedro, por su inmensa generosidad para conmigo a lo largo de toda mi carrera.
- A mi familia, mi novia y mis amigos que siempre me apoyaron.
- A la Universidad Nacional de La Plata, por permitirme acceder a una educación pública de calidad.

1. Resumen

Nuestro entendimiento de los fenómenos naturales habitualmente se sobresimplifica en la capacidad de mensurar propiedades de un sistema de estudio, que luego se interpreta probando alguna hipótesis (usualmente formalizada en lenguaje matemático). Luego, esta hipótesis se verifica a través de modelos que capturan su esencia o bien muta en nuevas hipótesis.

En la ciencia de materiales, el uso de modelos paramétricos es de uso habitual para obtener información de mensurables físicos. La información del mismo se obtiene al maximizar la verosimilitud entre los datos experimentales y el modelo propuesto.

En el presente trabajo de tesis de grado para optar por el título de licenciado en matemática, se propusieron y programaron modificaciones a algoritmos de Monte Carlo para el ajuste datos experimentales empleando funciones paramétricas donde además se analizó el rendimiento numérico de cada una de ellas. Se pusieron a prueba varias estrategias como la búsqueda aleatoria simple, métodos híbridos basados en búsqueda aleatoria combinados con algoritmos de máximo gradiente, como así también la implementación de recocidos simulados (*Simulated Annealing*).

Entre los aportes originales de este trabajo de tesis, se destaca la propuesta de desacoplar los parámetros lineales de los no lineales. En esta metodología, la búsqueda aleatoria sólo se centra en los términos no lineales y el resto de los términos se computan en cada simulación mediante cuadrados mínimos lineales, a diferencia de metodologías convencionales de Monte Carlo donde la exploración se realiza sobre todos los parámetros indistintamente de su naturaleza. Esta propuesta disminuye el tiempo de máquina al reducir la dimensionalidad del espacio de muestreo.

2. Introducción

La interpretación de mensurables (o datos experimentales) por ajustes de modelos matemáticos es de uso habitual tanto en ciencias naturales como en ciencias sociales, sea en el ámbito académico como en el productivo, en donde los algoritmos de optimización juegan un rol central ([Pan, 2012], [Pelikan et al., 1999]). La naturaleza de los modelos y/o funciones a emplear suele ser multiparamétrica con variables lineales y no lineales dependiendo del caso.

Los procedimientos habituales para optimización de mensurables emplean figuras de mérito estadísticas. Las mismas son funciones reales definidas en \mathbb{R}^M y de clase C_2 ¹. En los métodos basados en optimizar la verosimilitud entre los mensurables y el modelo ([Godambe and Heyde, 2010]) la figura de mérito debe ser maximizada o minimizada, según la naturaleza del método. En el presente trabajo de iniciación a la investigación se emplearán métodos basados en minimización de cuadrados.

El método de cuadrados mínimos es muy utilizado para obtener las magnitudes físicas a través de funciones paramétricas por ajuste sobre medidas experimentales ([Björck, 1990]). Si se tiene un conjunto de N mediciones o puntos experimentales $\{(x_i, y_i)\}_{i=1}^N$ (donde la medición y_i depende de x_i) y una función $y(x|\theta)$ modelo que depende de ciertos parámetros $\theta = (\theta_1, \dots, \theta_M)$, en este método se minimiza la diferencia entre la función modelo (o curva teórica) y los puntos experimentales agregando una ponderación w_i para cada punto experimental:

$$Z(\theta) = \sum_{i=1}^N w_i (y(x_i|\theta) - y_i)^2 \quad (1)$$

En este trabajo consideraremos errores normales alrededor de cada punto y_i obtenido experimentalmente, por lo que $w_i = \frac{1}{\sigma_i^2}$, donde σ_i es la desviación estándar de la distribución normal correspondiente al error en el punto i -ésimo. En este caso, la función $Z(\theta)$ pasa a tener una distribución χ^2 con $N - M$ grados de libertad:

$$Z(\theta) = \chi^2(\theta) = \sum_{i=1}^N \left(\frac{y(x_i|\theta) - y_i}{\sigma_i} \right)^2 \quad (2)$$

Para el caso particular de cuadrados mínimos lineales en general se asume que la función de prueba $y(x|\theta)$ puede ser escrita como la suma de contribuciones lineales:

$$y(x|\theta) = \sum_{j=1}^M \theta_j f_j(x) \quad (3)$$

Para encontrar el mínimo de la figura de mérito, deberemos buscar θ tal que:

$$\frac{\partial \chi^2}{\partial \theta_1}(\theta) = \dots = \frac{\partial \chi^2}{\partial \theta_M}(\theta) = 0 \quad (4)$$

Este conjunto de M ecuaciones puede expresarse de forma matricial de la siguiente manera:

¹Una función $f : \mathbb{R}^M \rightarrow \mathbb{R}$ es de clase C_n en un punto $x_0 \in \mathbb{R}^M$ si en algún entorno de x_0 existen sus derivadas parciales de orden n y son continuas. f es de clase C_n si lo es todo punto de \mathbb{R}^M .

$$(A^T \cdot A) \cdot \theta = A^T \cdot b, \quad (5)$$

donde $A \in \mathbb{R}^{N \times M}$ tal que $A_{ij} = \frac{f_j(x_i)}{\sigma_i}$ y $b \in \mathbb{R}^{N \times 1}$ tal que $b_{i1} = \frac{y_i}{\sigma_i}$.

Llamamos a las ecuaciones de (4), o equivalentemente de (5), las *ecuaciones normales*.

El valor de la figura de mérito no es sólo una medida de la discrepancia entre los datos experimentales y el modelo $y(x|\theta)$, sino que también nos brinda información sobre la confiabilidad estadística del mismo (se llama *goodness of fit* a esta propiedad. [Bevington et al., 1993]; [Press et al., 2007]). Minimizar la figura de mérito $\chi^2(\theta)$ equivale a maximizar la función de máxima verosimilitud

$$L(\theta) = \left(\prod_{i=1}^N \frac{1}{\sigma_i \sqrt{2\pi}} \right) \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \left[\frac{y(x_i|\theta) - y_i}{\sigma_i} \right]^2 \right\}.$$

A su vez, esto equivale a maximizar la probabilidad de que los datos experimentales provengan de la distribución $y(x|\theta)$, si consideramos que cada punto (x_i, y_i) tiene asociado un error normal con desviación estándar σ_i . Bajo estas hipótesis, y teniendo en cuenta que se espera que la distancia entre el modelo y los puntos experimentales $|y(x_i|\theta) - y_i|$ sea estadísticamente el error de medición σ_i y que la esperanza de la distribución χ^2 con $\nu = N - M$ grados de libertad es $N - M$, consideraremos como solución óptima a aquel θ tal que $\chi^2(\theta) = N - M$. Es común buscar minimizar la función normalizada $\frac{\chi^2(\theta)}{N - M}$ en vez de $\chi^2(\theta)$, cuyo el valor óptimo será 1.

De esta manera, para realizar el ajuste de modelos consideraremos minimizar la figura de mérito estadística

$$\chi^2(\theta) = \frac{1}{N - M} \sum_{i=1}^N \left(\frac{y(x_i|\theta) - y_i}{\sigma_i} \right)^2, \quad (6)$$

considerando a 1 como valor óptimo.

Cuando la función a minimizar es convexa, los algoritmos convencionales basados en gradientes resultan, no solo suficientes, sino que además eficientes desde el punto de vista computacional. Sin embargo, habitualmente si una función posee diversos mínimos locales dentro del espacio de estudio, la convergencia al mínimo global depende de los valores iniciales. Además, en los algoritmos en los que se emplean gradientes se debe contar con la expresión analítica de las derivadas parciales de cada parámetro o bien recurrir a diferenciación numérica. En este último caso existen diversos algoritmos que permiten estimarlo, pero no siempre son estables.

Para sistemas no lineales (es decir, cuando no se puede asumir que $y(x|\theta)$ puede ser escrita como la suma de contribuciones lineales) y en el caso de que se pueda escribir y/o calcular la derivada parcial del modelo propuesto en función de los parámetros no lineales, existen varios métodos que resuelven las ecuaciones normales de manera iterativa. Sin embargo, en los entornos de máximos o mínimos, la inversión de la matriz $(A^T A)$ podría no ser factible o el algoritmo podría tornarse inestable. Esto es lo que se conoce como problemas mal planteados o *ill posed problems* ([Hansen, 1994]). Dentro de estos métodos uno de los más utilizados es el método de Levenberg-Marquardt ([Levenberg, 1944]; [Marquardt, 1963]). El mismo consiste en agregar un término que estabiliza la inversión de la matriz sumando un parámetro positivo, variable, que regulariza numéricamente la operación ([Madsen et al., 2004]).

En otra rama de la optimización numérica se encuentran los métodos de Monte Carlo, que son muy populares por su robustez y estabilidad (en comparación con métodos basados en gradientes) a pesar de ser computacionalmente más costosos ([Robert et al., 2010]; [Robert and Casella, 2013]). Los métodos de Monte Carlo tradicionales requieren un número muy alto de simulaciones para tener una evaluación estadística confiable. Sin embargo, cuando se trabaja con modelos fisicoquímicos, el espacio de búsqueda puede ser acotado a resultados viables reduciendo el tiempo de máquina. Los algoritmos de Monte Carlo, básicamente, determinan mediante alguna metodología la aceptación o rechazo de un evento generado aleatoriamente dentro del espacio de estudio ([Pollock et al., 2016]; [Wang et al., 2019]). Entre las rutinas de minimización por Monte Carlo, uno de los algoritmos más utilizados en optimización de sistemas físicos fue propuesto por Metrópolis et al. ([Metropolis et al., 1953]), donde la energía de un sistema (en este caso se tratará de la figura de mérito) se minimiza según la función de densidad de probabilidad de Boltzmann ([Hägström, 2002]). Este es un método fundamentado en el muestreo de estados ([Metropolis et al., 1953]; [Hastings, 1970]), y basado en distribuciones de probabilidad en procesos aleatorios, conocido como cadena de Markov ([Gamerman and Lopes, 2006]).

En las últimas dos décadas se han propuesto nuevos métodos, basados en el algoritmo de Metrópolis, denominados métodos de Recocido Simulado (*Simulated Annealing*) ([Tsallis and Stariolo, 1996]; [Van Laarhoven and Aarts, 1987]), donde al igual que en la industria metalúrgica, se “enfía” un sistema lentamente desde temperaturas altas hacia temperaturas menores para obtener cristales grandes y homogéneos. Este “enfriamiento” del sistema se realiza disminuyendo controladamente el valor de la temperatura T , hasta 0 Kelvin. O sea, la condición de mínima entropía. Desde el punto de vista numérico esto implica disminuir la probabilidad de aceptación de un nuevo valor. Estos algoritmos se han utilizado en optimización y se han formulado varios modelos que permiten el control de la temperatura ([Ingber, 1993]).

3. Objetivos

Sobre la base expuesta en la introducción, seguidamente se enuncian los objetivos propuestos para el desarrollo de este trabajo de iniciación a la investigación para optar al título de Licenciado en Matemática.

El objetivo fundamental de este trabajo consiste en resolver un problema de optimización: encontrar los parámetros óptimos θ asociados a un modelo no lineal $y(x|\theta)$ que minimicen la figura de mérito $\chi^2(\theta)$. Para lograr esto, los algoritmos propuestos serán una modificación a las técnicas de Monte Carlo-Simulated Annealing (MC-SA) convencionales ([Smith and Roberts, 1993]) y se utilizarán para resolver problemas provenientes de las metodologías de caracterización de materiales, en este caso de dispersión de rayos X a bajos ángulos, como modelo de estudio.

Se buscará, además de buenos ajustes de la curva, optimizar el tiempo de máquina disminuyendo los tiempos de convergencia de la función de prueba utilizando las siguientes estrategias:

- Separación de términos lineales respecto de términos no lineales:

Para favorecer la convergencia de los algoritmos se plantea la separación de las contribuciones lineales ($\theta^L = (\theta_1^L, \dots, \theta_r^L)$) de las contribuciones no lineales ($\theta^{NL} = (\theta_1^{NL}, \dots, \theta_s^{NL})$) obteniendo:

$$y(x|\theta) = \sum_{j=1}^r \theta_j^L f_j(x, \theta^{NL})$$

En cada paso del algoritmo se perturbarán solamente las variables no lineales (θ^{NL}) y se recalcularán los parámetros lineales mediante rutinas de cuadrados mínimos lineales.

- Autorregulación de la temperatura del sistema:

La temperatura se determinará en cada iteración en función del valor mínimo de la figura de mérito $\chi^2(\theta_{opt})$ hasta la iteración anterior. Como propuesta inicial se considerará el siguiente esquema:

- Si $\chi^2(\theta_{opt}) < 10$, entonces $t = \chi^2(\theta_{opt})$.
- Si $\chi^2(\theta_{opt}) \geq 10$, entonces $t = \frac{\chi^2(\theta_{opt})}{\log_{10}(\chi^2(\theta_{opt}))}$.

El lenguaje de programación que se propone usar en el desarrollo de este trabajo es Python ([Guzdial and Ericson, 2016]; [Pilgrim and Willison, 2009]), dada la versatilidad, una sintaxis de baja complejidad y por la alta aceptación que posee dentro de la comunidad científica. Python es un lenguaje interpretado, de alto nivel con bastas librerías libres y abiertas a la comunidad escritas en su mayoría en C y Fortran, que son lenguajes de menor complejidad, pero optimizados para cálculo numérico.

4. Métodos de Monte Carlo

Como mencionamos anteriormente, en este trabajo nos concentraremos principalmente en encontrar los parámetros óptimos θ asociados a un modelo no lineal $y(x|\theta)$ que minimicen la figura de mérito $\chi^2(\theta)$.

Los problemas de optimización pueden ser abordados de manera numérica o estocástica. El rendimiento de los métodos numéricos depende fuertemente de las propiedades analíticas de la función a optimizar, mientras que estas propiedades juegan un rol menos importante en los métodos estocásticos o de simulación. Así, si la función objetivo es demasiado compleja como para permitir un estudio analítico o si su dominio es muy irregular, conviene utilizar algún método de tipo estocástico ([Robert et al., 2010]).

Para el caso de modelos no lineales, la figura de mérito $\chi^2(\theta)$ suele ser una función con múltiples mínimos (tanto locales como globales). Por lo tanto, para minimizarla no podremos aplicar una solución de tipo numérica analítica ([Bevington et al., 1993]). Sin embargo, existen diversos métodos estocásticos basados en métodos de Monte Carlo para llevar a cabo la búsqueda de los parámetros que optimicen la figura de mérito, algunos de los cuales desarrollaremos en las próximas secciones.

El método de Monte Carlo es un método no determinista usado para aproximar expresiones matemáticas complejas y costosas de evaluar. Fue llamado así en referencia al Casino de Monte Carlo (Mónaco), al ser la ruleta un generador simple de números aleatorios. El nombre y el desarrollo sistemático del método datan aproximadamente de 1944 y su aplicabilidad mejoró enormemente con el desarrollo de las computadoras.

El método de simulación de Monte Carlo consiste en producir un número elevado de escenarios al azar, estudiar la ocurrencia de un suceso en dichos escenarios y finalmente obtener una aproximación del resultado que se está buscando. Es por esto que proporciona soluciones aproximadas a una gran variedad de problemas independientemente de su complejidad, aunque su precisión dependerá del número de escenarios aleatorios generados. Para poder simular estos escenarios será necesario poder crear números aleatorios. Sin embargo, en la práctica, es suficiente poder producir números pseudoaleatorios, es decir, números cuya generación sea determinística pero que simule tan bien al azar que una prueba estadística no pueda notar la diferencia ([Robert and Casella, 2013]). Actualmente, casi todos los lenguajes de programación permiten generar este tipo de números.

4.1. Exploración independiente

Una manera sencilla (y rudimentaria) de usar simulaciones para minimizar la función $\chi^2(\theta)$ es generar puntos de su dominio usando alguna distribución de probabilidad determinada, evaluar la función en dichos puntos y tomar como θ óptimo al que obtenga el menor valor de $\chi^2(\theta)$.

La ventaja de este método es la extrema sencillez en su implementación. Si el problema cuenta con ciertas condiciones de regularidad, se puede asegurar la convergencia del algoritmo².

Sin embargo, este tipo de método puede resultar muy ineficiente si la distribución que se elige no tiene ninguna relación con la función objetivo y volverse rápidamente impráctico cuando la dimensión o la complejidad del problema aumenta.

²Informalmente, si basándose en la generación de los puntos θ hay probabilidad no nula de alcanzar θ que minimice χ^2 , entonces la probabilidad de encontrar el mínimo tiende a 1 ([Spall, 2005]).

4.1.1. Búsqueda Aleatoria para todo el espacio de parámetros

El método de búsqueda aleatoria propone buscar los parámetros θ que minimicen $\chi^2(\theta)$ de la siguiente manera:

- Para cada $i = 1, \dots, M$ definimos un rango de valores razonables (según el contexto del experimento que estemos realizando) $I_i = [a_i, b_i] \subset \mathbb{R}$ que puede tomar cada parámetro θ_i .
- Definimos la cantidad n de puntos aleatorios que vamos a generar sobre el espacio de parámetros $\prod_{i=1}^M I_i \subset \mathbb{R}^M$.
- Generamos n puntos aleatorios $\theta^1, \dots, \theta^n$ utilizando una distribución uniforme sobre el espacio de parámetros $\prod_{i=1}^M I_i$.
- Calculamos $\chi^2(\theta^1), \dots, \chi^2(\theta^n)$.
- Por último, tomamos θ_{opt} como el primero de los $\theta^1, \dots, \theta^n$ tal que $\chi^2(\theta_{opt}) = \min_k \chi^2(\theta^k)$. Este θ_{opt} será nuestro vector de parámetros óptimos encontrado.

4.1.2. Búsqueda Aleatoria sólo para los parámetros no lineales

Teniendo en cuenta que $\theta = (\theta^L, \theta^{NL})$, donde $\theta^L = (\theta_1^L, \dots, \theta_r^L)$, $\theta^{NL} = (\theta_1^{NL}, \dots, \theta_s^{NL})$ y $r + s = M$ y que consideramos modelos no lineales de la forma $y(x|\theta) = \sum_{j=1}^r \theta_j^L f_j(x, \theta^{NL})$, se puede mejorar el método de búsqueda aleatoria reduciendo la dimensión del espacio de búsqueda realizándola sólo en el espacio de los parámetros no lineales (θ^{NL}) y calculando los parámetros lineales (θ^L) mediante ajuste por regresión lineal (método de cuadrados mínimos).

Este método de búsqueda aleatoria propone buscar los parámetros θ que minimicen $\chi^2(\theta)$ de la siguiente manera:

- Para cada $i = 1, \dots, M$ definimos un rango de valores razonables (según el contexto del experimento que estemos realizando) $I_i = [a_i, b_i] \subset \mathbb{R}$ que puede tomar cada parámetro θ_i .
- Definimos la cantidad n de puntos aleatorios que vamos a generar sobre el espacio de parámetros no lineales $\prod_{i=1}^s I_i \subset \mathbb{R}^s$.
- Generamos n puntos aleatorios $\gamma^1, \dots, \gamma^n$ utilizando una distribución uniforme sobre el espacio de parámetros no lineales $\prod_{i=1}^s I_i$.
- Para cada punto γ^k , consideramos $y(x|(\theta^L, \gamma^k)) = \sum_{j=1}^r \theta_j^L f_j(x, \gamma^k)$. Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β^k . Así, construimos $\theta^k = (\beta^k, \gamma^k)$.
- Calculamos $\chi^2(\theta^1), \dots, \chi^2(\theta^n)$.
- Por último, tomamos θ_{opt} como el primero de los $\theta^1, \dots, \theta^n$ tal que $\chi^2(\theta_{opt}) = \min_k \chi^2(\theta^k)$. Este θ_{opt} será nuestro vector de parámetros óptimos encontrado.

4.2. Exploración dependiente: Simulated Annealing

Un enfoque distinto a la búsqueda aleatoria para la optimización de $\chi^2(\theta)$ consiste en explorar la superficie de la función objetivo de forma local, es decir, definir una sucesión $\theta^1, \dots, \theta^n$ de puntos en su dominio de manera que la elección de θ^{j+1} dependa de θ^j . Esto contrasta con la búsqueda aleatoria, donde cada punto es independiente de los otros.

Para implementar este tipo de búsqueda estocástica, podríamos considerar elegir el punto θ^{j+1} con cualquier distribución que tenga en cuenta al punto θ^j (por ejemplo, una distribución normal centrada en θ^j). Sin embargo, dado que el objetivo es optimizar la figura de mérito, la búsqueda será más eficiente si logramos incluir información de dicha figura al momento de elegir el próximo punto.

El algoritmo de Recocido Simulado (*Simulated Annealing*) fue diseñado para resolver el problema de optimizar una función $f : \mathbb{R}^M \rightarrow \mathbb{R}$ que presente múltiples extremos locales. El nombre del método proviene de la industria metalúrgica, donde se enfría un sistema lentamente desde temperaturas altas hacia temperaturas menores para obtener cristales grandes y homogéneos. El método fue introducido por Kirkpatrick et al. ([Kirkpatrick et al., 1983]) y la teoría detrás de él puede encontrarse en ([Geman and Geman, 1984]; [Van Laarhoven and Aarts, 1987]) para el caso discreto y ([Haario and Saksman, 1991]; [Del Moral and Miclo, 1999]) para el caso continuo.

El método propone construir la sucesión de puntos $\theta^1, \dots, \theta^n$ para explorar el espacio de parámetros de una forma completamente distinta a los métodos mencionados anteriormente. Para esto, se define una sucesión de temperaturas positivas t_0, \dots, t_F que decrece acercándose a 0 y se construye una sucesión de densidades $\pi_{t_0}, \dots, \pi_{t_F}$ que concentren cada vez más su peso en los mínimos de f a medida que la temperatura decrezca, logrando que los valores simulados θ^k se concentren cada vez más cerca de los mínimos globales de f ([Robert et al., 2010]).

Para la implementación de este método surgen tres dificultades principales:

- La construcción de una sucesión de densidades $\pi_{t_0}, \dots, \pi_{t_F}$ que concentren cada vez más su peso en los mínimos de f a medida que la temperatura decrezca.
- La simulación de puntos a partir de las distribuciones π_t .
- La selección de temperaturas t_0, \dots, t_F .

Mientras que el tercer ítem es fuertemente dependiente del problema, el primer y el segundo ítem permiten una solución genérica.

La solución del segundo ítem se basa en el algoritmo de Metropolis-Hastings, propuesto por primera vez por Metropolis et al. ([Metropolis et al., 1953]), como veremos a continuación.

El algoritmo de Metropolis-Hastings permite samplear cualquier función $h(x)$ que represente una distribución de probabilidad, ya sea que esta esté normalizada (es decir, $\int_{\mathbb{R}^M} h(z) dz = 1$) o no. La idea del algoritmo es construir una cadena de Markov³ cuya única distribución estacionaria sea $\pi(x) = \frac{h(x)}{\int_{\mathbb{R}^M} h(z) dz}$. Es así que si simulamos puntos (x_0, x_1, \dots) de esa cadena, para k lo suficientemente grande, (x_k, x_{k+1}, \dots) tendrá distribución π .

El algoritmo construye la cadena de Markov de la siguiente manera:

1. Consideramos la distribución normalizada $\pi(x) = \frac{h(x)}{\int_{\mathbb{R}^M} h(z) dz}$ y $\{q(x, \cdot) : \mathbb{R}^M \rightarrow \mathbb{R}_{\geq 0} / x \in \mathbb{R}^M\}$ una familia de distribuciones que podemos samplear.

³Para teoría sobre cadenas de Markov ver ([Geman and Lopes, 2006]).

2. Elegimos un x_0 inicial al azar tal que $h(x_0) > 0$.
3. Partiendo del valor actual x_k , se propone un nuevo valor y sampleando la distribución conocida $q(x_k, \cdot)$.
4. Calculamos

$$\alpha(x_k, y) := \begin{cases} \min \left\{ \frac{\pi(y)q(y, x_k)}{\pi(x_k)q(x_k, y)}, 1 \right\} & \text{si } \pi(x_k)q(x_k, y) > 0 \\ 1 & \text{si } \pi(x_k)q(x_k, y) = 0 \end{cases} \quad (7)$$

$$= \begin{cases} \min \left\{ \frac{h(y)q(y, x_k)}{h(x_k)q(x_k, y)}, 1 \right\} & \text{si } h(x_k)q(x_k, y) > 0 \\ 1 & \text{si } h(x_k)q(x_k, y) = 0 \end{cases}$$

5. Aceptamos $x_{k+1} = y$ con probabilidad $\alpha(x_k, y)$. En caso de que se dé un rechazo, tomamos $x_{k+1} = x_k$.
6. Volvemos al paso 3.

Así, queda construida una cadena de Markov $C = (X^0, X^1, \dots)$, donde cada variable aleatoria $X^k : \Omega \rightarrow \mathbb{R}^M$ representa el valor x_k que se toma en el paso k .

De esta construcción resulta que π es la distribución estacionaria de C ([Chib and Greenberg, 1995]). Luego, para k lo suficientemente grande, $(x_k, x_{k+1} \dots)$ tendrá distribución π . Esta convergencia hacia la distribución π se da bajo ciertas condiciones de regularidad: se necesita que la cadena C sea irreducible y aperiódica ([Smith and Roberts, 1993]). En particular, estas condiciones se cumplen si $q(x, \cdot)$ tiene densidad positiva en el soporte de $\pi(\cdot)$ o incluso en un soporte restringido a una vecindad del punto x (como por ejemplo cualquier distribución normal centrada en x).

Notemos que los pasos anteriores no conforman un algoritmo, ya que para construir la sucesión deberíamos iterar infinitamente. Lo que se hace para generar una muestra de la distribución es:

1. Definimos n_I (iteración a partir de la cual consideraremos los valores generados - todos los valores anteriores no serán tenidos en cuenta) y n_F (cantidad de iteraciones que realizaremos en total). Consideramos $\{q(x, \cdot) : \mathbb{R}^M \rightarrow \mathbb{R}_{\geq 0} / x \in \mathbb{R}^M\}$ una familia de distribuciones que podemos samplear.
2. Elegimos un x_0 inicial al azar tal que $h(x_0) > 0$.
3. Partiendo del valor actual x_k , se propone un nuevo valor y sampleando la distribución conocida $q(x_k, \cdot)$.
4. Calculamos

$$\alpha(x_k, y) := \begin{cases} \min \left\{ \frac{\pi(y)q(y, x_k)}{\pi(x_k)q(x_k, y)}, 1 \right\} & \text{si } \pi(x_k)q(x_k, y) > 0 \\ 1 & \text{si } \pi(x_k)q(x_k, y) = 0 \end{cases} \quad (8)$$

$$= \begin{cases} \min \left\{ \frac{h(y)q(y, x_k)}{h(x_k)q(x_k, y)}, 1 \right\} & \text{si } h(x_k)q(x_k, y) > 0 \\ 1 & \text{si } h(x_k)q(x_k, y) = 0 \end{cases}$$

5. Aceptamos $x_{k+1} = y$ con probabilidad $\alpha(x_k, y)$. En caso de que se de un rechazo, tomamos $x_{k+1} = x_k$.
6. Si $k < n_F - 1$, volvemos al paso 3.
Si $k = n_F - 1$, damos por finalizado el proceso.

Como resultado obtendremos una sucesión de valores $(x_0, \dots, x_{n_I}, \dots, x_{n_F})$, de la cual nos quedaremos con $(x_{n_I}, \dots, x_{n_F})$. Tomando n_i lo suficientemente grande, $(x_{n_I}, \dots, x_{n_F})$ tendrá distribución π (la distribución de la cual queremos samplear una muestra), y el tamaño de dicha muestra vendrá determinado por $n_F - n_I$.

Es interesante notar que el cálculo de $\alpha(x, y)$ no requiere el conocimiento de la constante normalizadora de π ($\int_{\mathbb{R}^M} h(z) dz$), ya que se cancela en el cociente.

Por otro lado, en el caso especial en el que la distribución generadora de candidatos $q(x, \cdot)$ sea simétrica, la probabilidad de movimiento $\alpha(x, y)$ se reduce a calcular $\min \left\{ \frac{\pi(y)}{\pi(x)}, 1 \right\}$. Este (importante) caso particular es el algoritmo de Metropolis.

Recordando las tres dificultades principales para la implementación del algoritmo de Simulated Annealing, vimos que la selección de temperaturas t_0, \dots, t_F depende específicamente del problema a resolver y mostramos cómo simular puntos a partir de las distribuciones π_t utilizando el algoritmo de Metropolis-Hastings.

Veamos ahora cómo resolver la construcción de una sucesión de densidades $\pi_{t_0}, \dots, \pi_{t_F}$ que concentren cada vez más su peso en los mínimos de χ^2 a medida que la temperatura decrezca.

Definición 4.1. Sean $f : \mathbb{R}^M \rightarrow \mathbb{R}$ y $T > 0$, definimos la distribución de Boltzmann como:

$$\pi_{f,T}(x) = \frac{1}{c_{f,T}} \exp\left(\frac{-f(x)}{T}\right), \quad (9)$$

donde

$$c_{f,T} = \int_{\mathbb{R}^M} \exp\left(\frac{-f(x)}{T}\right) \quad (10)$$

es una constante normalizadora para que $\int_{\mathbb{R}^M} \pi_{f,T}(x) dx = 1$.

De la construcción de estas distribuciones $\pi_{f,T}$ es claro que a medida que T se acerca a 0, $\pi_{f,T}$ concentra cada vez más su densidad en los valores pequeños de f^4 .

El diseño del algoritmo de Simulated Annealing para encontrar un mínimo global de f es el siguiente:

1. Elegimos una sucesión de temperaturas positivas (t_0, \dots, t_F) que decrezca acercándose a 0, una sucesión de números naturales (N_0, \dots, N_F) y familias de distribuciones $\mathcal{F}_0, \dots, \mathcal{F}_F$ (con $\mathcal{F}_k = \{q_k(x, \cdot) : \mathbb{R}^M \rightarrow \mathbb{R}_{\geq 0} / x \in \mathbb{R}^M\}$) con las que buscaremos los estados en cada cadena C_k .
2. Tomamos un estado inicial x_0 al azar de \mathbb{R}^M .

⁴Las bases de la prueba de esta propiedad se encuentran en ([Hägström, 2002])

3. Construimos una cadena de Markov C_0 utilizando el algoritmo de Metropolis-Hastings (usando la familia de distribuciones \mathcal{F}_0 para buscar estados) para samplear la distribución π_{f,t_0} . Corremos esta cadena por N_0 iteraciones.
4. Construimos una cadena de Markov C_1 utilizando el algoritmo de Metropolis-Hastings (usando la familia de distribuciones \mathcal{F}_1 para buscar estados) para samplear la distribución π_{f,t_1} . Corremos esta cadena comenzando desde el estado que haya obtenido el menor valor de f en el paso anterior, por una cantidad de N_1 pasos.
5. Continuamos con el proceso hasta llegar hasta la última cadena C_F .

La elección de (t_0, \dots, t_F) y (N_0, \dots, N_F) es llamada el esquema de annealing y es crucial para encontrar un buen mínimo ¿Cuán rápido debemos hacer decaer la temperatura hacia 0 a medida que aumentan las iteraciones? Si la temperatura decae lo suficientemente lento, la probabilidad de encontrar un mínimo global tiende a 1 ([Häggström, 2002]). Sin embargo, para poder asegurar esto en la mayoría de los casos deberíamos hacer decrecer la temperatura tan lentamente que el algoritmo demoraría demasiado tiempo en finalizar su ejecución, haciéndolo impráctico. Por lo tanto, la mayoría de los esquemas de annealing disminuyen su temperatura más rápido. Pero enfriar demasiado rápido puede hacer que el algoritmo se quede estancado en un mínimo local, imposibilitando la convergencia a un mínimo global. Es decir, la temperatura del sistema debe disminuirse lo suficientemente rápido para poder hallar un mínimo global en un tiempo razonable, pero lo suficientemente lento como para no estancarse en mínimos locales. La forma de construir estos esquemas carece de exactitud y está más asociado a la experiencia práctica ([Häggström, 2002]).

A continuación veremos cómo usando el método de Simulated Annealing podemos buscar los parámetros óptimos θ asociados a un modelo no lineal $y(x|\theta)$ que minimicen la figura de mérito $\chi^2(\theta)$.

1. Para cada $i = 1, \dots, M$ definimos un rango de valores razonables (según el contexto del experimento que estemos realizando) $I_i = [a_i, b_i] \subset \mathbb{R}$ que puede tomar cada parámetro θ_i . Llamamos el espacio de parámetros al conjunto $\prod_{i=1}^M I_i \subset \mathbb{R}^M$.
2. Elegimos una sucesión de temperaturas positivas (t_0, \dots, t_F) que decrezca acercándose a 0, una sucesión de números naturales (N_0, \dots, N_F) y familias de distribuciones $\mathcal{F}_0, \dots, \mathcal{F}_F$ (con $\mathcal{F}_k = \{q_k(x, \cdot) : \prod_{i=1}^M I_i \rightarrow \mathbb{R}_{\geq 0} / x \in \prod_{i=1}^M I_i\}$) con las que buscaremos los estados en cada cadena C_k .

Para este trabajo consideramos, para cada $k = 0, \dots, F$ y para cada $x \in \prod_{i=1}^M I_i$:

$$q_k(x, y) = \prod_{i=1}^M \frac{1}{\sqrt{2\pi\sigma_{k,i}^2}} e^{-\frac{(y_i - x_i)^2}{2\sigma_{k,i}^2}},$$

donde cada $\sigma_{k,i}$ es definida de antemano para cada cadena C_k .

3. Construimos una cadena de Markov C_0 utilizando el algoritmo de Metropolis-Hastings (usando la familia de distribuciones \mathcal{F}_0 para buscar estados) para samplear la distribución π_{χ^2, t_0} por N_0 iteraciones:

- I Generamos θ^0 utilizando una distribución uniforme sobre el espacio de parámetros $\prod_{i=1}^M I_i$.
- II Partiendo del valor actual θ^k , proponemos un nuevo valor θ sampleando la distribución $q_0(\theta^k, \cdot)$.

III Calculamos

$$\alpha(\theta^k, \theta) = \min \left\{ \exp \left(\frac{\chi^2(\theta^k) - \chi^2(\theta)}{T_0} \right), 1 \right\}$$

- IV Aceptamos $\theta^{k+1} = \theta$ con probabilidad $\alpha(\theta^k, \theta)$. En caso de que se de un rechazo, tomamos $\theta^{k+1} = \theta^k$.

V Si $k < N_0 - 1$, volvemos al paso II.

Si $k = N_0 - 1$, consideramos $\theta_{opt}^{C_0}$ como el primero de los $\theta^0, \dots, \theta^{N_0-1}$ tal que $\chi^2(\theta_{opt}^{C_0}) = \min_k \chi^2(\theta^k)$. Este $\theta_{opt}^{C_0}$ será nuestro vector de parámetros óptimos encontrado para C_0 .

4. Construimos una cadena de Markov C_1 como en el paso 3, utilizando el algoritmo de Metropolis-Hastings (usando la familia de distribuciones \mathcal{F}_1 para buscar estados) para samplear la distribución π_{χ^2, t_1} por N_1 iteraciones, comenzando con $\theta^0 = \theta_{opt}^{C_0}$.
5. Continuamos con el proceso hasta llegar hasta la última cadena C_F , siempre comenzando desde el vector de parámetros óptimos encontrado en la cadena anterior.
6. Consideramos θ_{opt} como el primero de los $\theta_{opt}^{C_0}, \dots, \theta_{opt}^{C_F}$ tal que $\chi^2(\theta_{opt}) = \min_k \chi^2(\theta_{opt}^{C_k})$. Este θ_{opt} será nuestro vector de parámetros óptimos encontrado para todo el proceso.

De la misma manera que hicimos con la búsqueda aleatoria, dado que consideramos modelos no lineales de la forma $y(x|\theta) = \sum_{j=1}^r \theta_j^L f_j(x, \theta^{NL})$, se puede mejorar el método de Simulated Annealing realizando la búsqueda sólo en el espacio de los parámetros no lineales (θ^{NL}) y calculando los parámetros lineales (θ^L) mediante ajuste por regresión lineal (método de cuadrados mínimos).

Teniendo en cuenta que $\theta = (\theta^L, \theta^{NL})$, donde $\theta^L = (\theta_1^L, \dots, \theta_r^L)$, $\theta^{NL} = (\theta_1^{NL}, \dots, \theta_s^{NL})$ y $r + s = M$, se propone buscar los parámetros θ que minimicen $\chi^2(\theta)$ de la siguiente manera:

1. Para cada $i = 1, \dots, M$ definimos un rango de valores razonables (según el contexto del experimento que estemos realizando) $I_i = [a_i, b_i] \subset \mathbb{R}$ que puede tomar cada parámetro θ_i . Llamamos el espacio de parámetros no lineales al conjunto $\prod_{i=1}^s I_i \subset \mathbb{R}^s$.
2. Elegimos una sucesión de temperaturas positivas (t_0, \dots, t_F) que decrezca acercándose a 0, una sucesión de números naturales (N_0, \dots, N_F) y familias de distribuciones $\mathcal{F}_0, \dots, \mathcal{F}_F$ (con $\mathcal{F}_k = \{q_k(x, \cdot) : \prod_{i=1}^s I_i \rightarrow \mathbb{R}_{\geq 0} / x \in \prod_{i=1}^s I_i\}$) con las que buscaremos los estados en cada cadena C_k .

Para este trabajo consideramos, para cada $k = 0, \dots, F$ y para cada $x \in \prod_{i=1}^s I_i$:

$$q_k(x, y) = \prod_{i=1}^s \frac{1}{\sqrt{2\pi\sigma_{k,i}^2}} e^{-\frac{(y_i - x_i)^2}{2\sigma_{k,i}^2}},$$

donde cada $\sigma_{k,i}$ es definida de antemano para cada cadena C_k .

3. Construimos una cadena de Markov C_0 utilizando el algoritmo de Metropolis-Hastings (usando la familia de distribuciones \mathcal{F}_0 para buscar estados) para samplear la distribución π_{χ^2, t_0} por N_0 iteraciones:

I Generamos γ^0 utilizando una distribución uniforme sobre el espacio de parámetros no lineales $\prod_{i=1}^s I_i$.

Consideramos $y(x|\theta^L, \gamma^0) = \sum_{j=1}^r \theta_j^L f_j(x, \gamma^0)$. Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β^0 .

Así, construimos $\theta^0 = (\beta^0, \gamma^0)$.

II Partiendo del valor actual $\theta^k = (\beta^k, \gamma^k)$, proponemos un nuevo valor α sampleando la distribución $q_0(\gamma^k, \cdot)$.

Consideramos $y(x|\theta^L, \gamma) = \sum_{j=1}^r \theta_j^L f_j(x, \gamma)$. Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β .

Así, construimos $\theta = (\beta, \gamma)$.

III Calculamos

$$\alpha(\theta^k, \theta) = \min \left\{ \exp \left(\frac{\chi^2(\theta^k) - \chi^2(\theta)}{T_0} \right), 1 \right\}$$

IV Aceptamos $\theta^{k+1} = \theta$ con probabilidad $\alpha(\theta^k, \theta)$. En caso de que se de un rechazo, tomamos $\theta^{k+1} = \theta^k$.

V Si $k < N_0 - 1$, volvemos al paso II.

Si $k = N_0 - 1$, consideramos $\theta_{opt}^{C_0}$ como el primero de los $\theta^0, \dots, \theta^{N_0-1}$ tal que $\chi^2(\theta_{opt}^{C_0}) = \min_k \chi^2(\theta^k)$. Este $\theta_{opt}^{C_0}$ será nuestro vector de parámetros óptimos encontrado para C_0 .

4. Construimos una cadena de Markov C_1 como en el paso 3, utilizando el algoritmo de Metropolis-Hastings (usando la familia de distribuciones \mathcal{F}_1 para buscar estados) para samplear la distribución π_{χ^2, t_1} por N_1 iteraciones, comenzando con $\theta^0 = \theta_{opt}^{C_0}$.

5. Continuamos con el proceso hasta llegar hasta la última cadena C_F , siempre comenzando desde el vector de parámetros óptimos encontrado en la cadena anterior.

6. Consideramos θ_{opt} como el primero de los $\theta_{opt}^{C_0}, \dots, \theta_{opt}^{C_F}$ tal que $\chi^2(\theta_{opt}) = \min_k \chi^2(\theta_{opt}^{C_k})$. Este θ_{opt} será nuestro vector de parámetros óptimos encontrado para todo el proceso.

Hemos definido y desarrollado una serie de métodos basados en los métodos de Monte Carlo para resolver el problema de encontrar los parámetros óptimos θ asociados a un modelo no lineal $y(x|\theta)$ que se ajuste a datos experimentales mediante la minimización de la figura de mérito $\chi^2(\theta)$. En la siguiente sección procederemos a estudiar su implementación para resolver problemas provenientes de las metodologías de caracterización de materiales, en este caso de dispersión de rayos X a bajos ángulos.

5. Resultados

5.1. SAXS

Los rayos (o radiación) X son ondas electromagnéticas de longitud de onda similar a la distancia de un enlace químico ($\sim 10^{-10}$ metros). De la interacción de la radiación X con la materia surgen diversas técnicas experimentales que permiten obtener información estructural atómica y electrónica. Entre dichas técnicas se encuentra la dispersión de rayos X a bajos ángulos (SAXS por sus siglas en inglés) la cual permite acceder a información estructural en la escala nanométrica ($1 \text{ nm} = 10^{-9}$ metros). El fenómeno de dispersión elástica, donde el fotón conserva su energía, ocurre cuando existen inhomogeneidades en la densidad electrónica de la materia (número de electrones por unidad de volumen) ([Pedersen, 1997]). Un experimento de SAXS consiste en irradiar con un frente de onda plano y monocromático una muestra, donde se registran los fotones dispersados en función del ángulo sobre un plano de detección como se muestra en la Figura 1.

La interpretación matemática más simple de un experimento de SAXS se encuentra bajo el marco formal del espacio de Fourier ([Craievich, 2016]). Es decir que el vínculo entre las coordenadas reales de la materia y el espacio de la intensidad detectada está mediada por el cuadrado de la amplitud de la onda obtenida de la transformada de Fourier del volumen del objeto.

En el presente trabajo de tesis se analizó un experimento realizado en un sistema micelar, donde desde el punto de vista químico una micela es una agrupación espontánea de moléculas de detergente en agua ([Huck-Iriart et al., 2016]; [Marcolongo and Mirenda, 2011]). En el sistema particular estudiado, dichas agrupaciones conforman dispersiones en forma de cintas largas como se muestra en la micrografía de transmisión electrónica de la Figura 2.

El patrón de SAXS fue provisto por gentileza del Prof. Dr. Martín Conda Sheridan (UNMC, USA) el cual fue obtenido en el sincrotrón Advance Photon Source (APS) en Chicago, USA. (Los sincrotrones son grandes aceleradores de electrones que emiten radiación X de alta intensidad). Para el desarrollo de un modelo que ajuste los datos experimentales, hay que tener en cuenta que una molécula de detergente tiene dos regiones químicamente diferenciales, una parte hidrofóbica (rechaza el agua) y una parte hidrofílica (atrae el agua); es decir que cada zona tiene una densidad electrónica diferente. Para el modelo consideraremos que el largo de la cinta es infinito (desde el punto de vista experimental) y que sólo tenemos acceso a la sección transversal de la cinta como se exhibe en la Figura 3.

La intensidad normalizada fue representada en función del módulo del momento transferido q , con $q = (\frac{4\pi}{\lambda})\text{sen}(\theta)$, donde λ es el longitud de onda del rayo X y 2θ es el ángulo de dispersión. Los patrones de SAXS fueron ajustados empleando un modelo de cintas infinitamente largas de sección rectangular con altura $h + 2t_h$ y ancho $w + 2t_w$ (ver Figura 3). Como los detergentes tienen dos partes químicamente diferenciadas, se considera además que la cinta tiene dos secciones concéntricas donde w y h conforman el interior hidrofóbico y t_h y t_w la sección externa hidrofílica. La diferencia de densidades electrónicas está representada en el término β . La intensidad total viene dada por la integración alrededor del ángulo acimutal φ :

$$I(q) = sc \left\{ \frac{1}{q} \int_0^{\frac{\pi}{2}} [g(\varphi, h, w, t_h, t_w, \beta)]^2 d\varphi \right\} + \text{back}, \quad (11)$$

donde

$$\begin{aligned}
 g(\varphi, h, w, t_h, t_w, \beta) &= S(q_h, h)S(q_w, w) \\
 &+ \frac{\beta}{2} [S(q_h, h + 2t_h) - S(q_h, h)] S(q_w, w) \\
 &+ \frac{\beta}{2} [S(q_w, w + 2t_w) - S(q_w, w)] S(q_h, h),
 \end{aligned} \tag{12}$$

con

$$\begin{aligned}
 q_h &= q \cos(\varphi) \\
 q_w &= q \sin(\varphi),
 \end{aligned} \tag{13}$$

$S(q, a)$ representa la transformada de Fourier de la función *rec*:

$$S(q, a) = \int_{\mathbb{R}} \text{rec}(x) e^{ixq} dx = a \frac{\text{sen}\left(\frac{qa}{2}\right)}{\frac{qa}{2}}, \tag{14}$$

donde $\text{rec}(x) = 1$ si $|x| < \frac{a}{2}$, y 0 en caso contrario, *sc* es un factor de escala general y *back* es la constante de fondo.

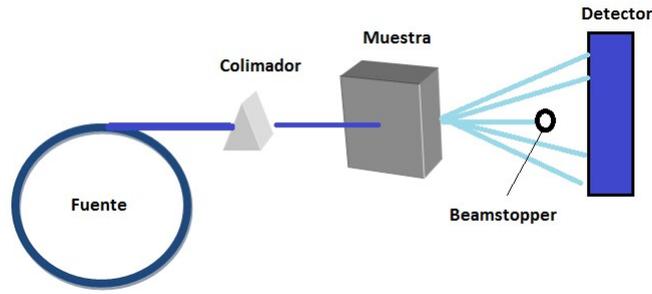


Figura 1: Esquema experimental básico de un experimento de SAXS.

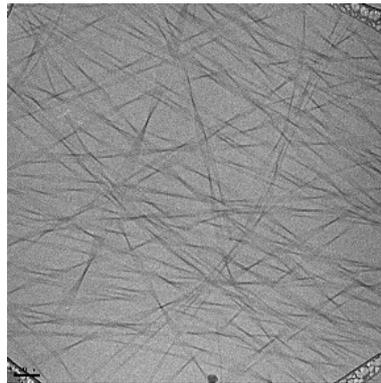


Figura 2: Micrografía de Transmisión electrónica del sistema micelar estudiado. La barra indica la escala de 50 nm.

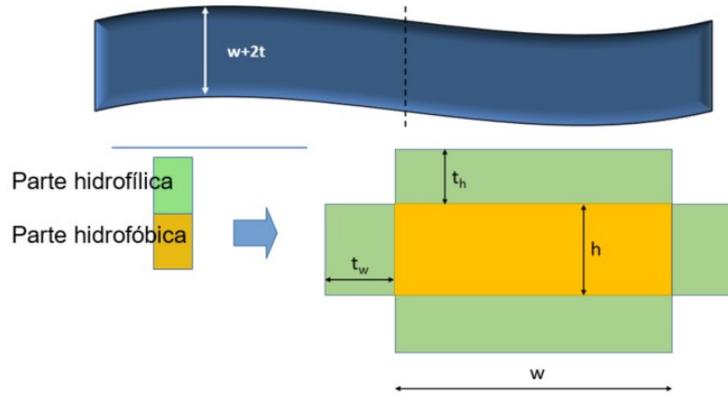


Figura 3: Representación de una cinta formada por moléculas de detergente donde experimentalmente tenemos acceso a la información de la sección transversal de la misma. Cada color representa una densidad electrónica diferente.

5.2. Datos experimentales

Los datos experimentales obtenidos pueden verse en la Figura 4. Forman un total de 918 mediciones. Así, tenemos (x_i, y_i) con $i = 1, \dots, 918$.

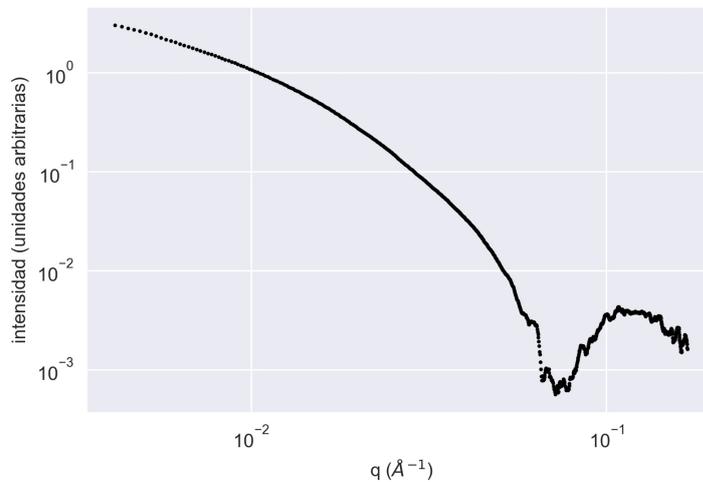


Figura 4: Curva experimental SAXS de la intensidad de dispersión en función del momento transferido q . Para una mejor visualización se emplean escalas logarítmicas en ambos ejes.

Para cada medición (x_i, y_i) consideraremos un error $\sigma_i = y_i * 0.1$.⁵

⁵Para este experimento sólo contamos con un conjunto de mediciones, por lo que no poseemos los valores de los errores estadísticos σ_i . Consideramos $\sigma_i = y_i * 0.1$ como valores razonables para la naturaleza del experimento. En este caso, como los σ_i fueron determinados de esta manera, no se puede asegurar que el valor óptimo esperado de $\chi^2(\theta)$ sea 1.

El modelo propuesto es:

$$y(x|\theta) = \theta_0^L + \theta_1^L f_1(x|\theta^{NL}), \quad (15)$$

donde:

$$f_1(x|\theta^{NL}) = \left\{ \frac{1}{x} \int_0^{\frac{\pi}{2}} [g(\varphi, h, w, t_h, t_w, \beta)]^2 d\varphi \right\}, \quad (16)$$

$$\begin{aligned} g(\varphi, h, w, t_h, t_w, \beta) &= S(x_h, h)S(x_w, w) \\ &+ \frac{\beta}{2} [S(x_h, h + 2t_h) - S(x_h, h)] S(x_w, w) \\ &+ \frac{\beta}{2} [S(x_w, w + 2t_w) - S(x_w, w)] S(x_h, h), \end{aligned} \quad (17)$$

$$\begin{aligned} x_h &= x \cos(\varphi), \\ x_w &= x \sin(\varphi), \end{aligned} \quad (18)$$

y

$$S(x, y) = y \frac{\text{sen}\left(\frac{xy}{2}\right)}{\frac{xy}{2}}. \quad (19)$$

Así

$$y(x|\theta) = \theta_0^L + \theta_1^L f_1(x|\theta^{NL}),$$

con

$$\theta^L = (\theta_0^L, \theta_1^L) = (\text{back}, \text{sc}) \text{ y } \theta^{NL} = (\theta_0^{NL}, \theta_1^{NL}, \theta_2^{NL}, \theta_3^{NL}, \theta_4^{NL}) = (h, w, t_h, t_w, \beta).$$

Teniendo en cuenta valores consistentes con la naturaleza física del experimento, consideramos los siguientes rangos de búsqueda para cada parámetro:

- $\theta_0^L \in I_0^L = [0, 0.4]$
- $\theta_1^L \in I_1^L = [0, 0.4]$
- $h[\text{Å}] \in I_0^{NL} = [5, 75]$
- $w[\text{Å}] \in I_1^{NL} = [5, 1500]$
- $t_h[\text{Å}] \in I_2^{NL} = [0, 100]$
- $t_w[\text{Å}] \in I_3^{NL} = [0, 100]$
- $\beta \in I_4^{NL} = [-10, 10]$

Así, el espacio de parámetros $E = (\prod_{i=0}^1 I_i^L) \times (\prod_{i=0}^4 I_i^{NL})$ es un subconjunto de \mathbb{R}^7 .

5.3. Implementación de los métodos

Para encontrar los parámetros óptimos θ asociados al modelo no lineal $y(x|\theta)$ que minimicen la figura de mérito $\chi^2(\theta)$, implementaremos los métodos desarrollados anteriormente aplicándolos a los datos de este experimento:

- Método 1: búsqueda aleatoria sobre el espacio de parámetros total E .
- Método 1.1: búsqueda aleatoria sobre el espacio de parámetros no lineales, con ajuste de regresión lineal para los parámetros lineales.
- Método 2: método de Levenberg - Marquardt modificado⁶.
- Método 3: Simulated Annealing sobre el espacio de parámetros no lineales, con ajuste de regresión lineal para los parámetros lineales.

Con el fin de implementar estos métodos, hemos desarrollado todos los códigos de los algoritmos en el lenguaje de programación Python.

Dado que los resultados de los métodos estocásticos dependen fuertemente de los valores desde donde se comienza la simulación, para cada método comenzaremos la simulación desde una cantidad prefijada de puntos del espacio de parámetros, sampleados con una distribución uniforme sobre dicho espacio. Para cada método se fijó un máximo de 1000 iteraciones por cada punto inicial (con una cantidad máxima de 200 puntos iniciales) o de 1200 segundos de tiempo de ejecución total, lo que ocurra primero. Sólo para el método 2 (Levenberg - Marquardt modificado) no logramos controlar la cantidad de iteraciones que se realizan, ya que su implementación está basada en la librería externa *lmfit* ([Newville et al., 2014]), por lo que su criterio de corte fue sólo el tiempo de ejecución. Teniendo en cuenta este criterio de corte, para los métodos 1, 1.1 y 3 se realizaron 200 inicios cada uno con 1000 iteraciones, mientras que para el método 2 se realizaron 40 inicios, sin controlar las iteraciones.

5.3.1. Implementación del método 1

- Para i entre 0 y 199 (cantidad de inicios):
 - Generamos un punto θ^0 inicial usando una distribución uniforme (cuya semilla depende de i) sobre el espacio de parámetros.
 - Definimos $\theta_{opt} = \theta^0$ y $\chi^2(\theta_{opt}) = \chi^2(\theta^0)$.
 - Para k entre 0 y 999 (cantidad de iteraciones):
 - Generamos un punto θ^k usando una distribución uniforme sobre el espacio de parámetros.
 - Calculamos $\chi^2(\theta^k)$.
 - Si $\chi^2(\theta^k) < \chi^2(\theta_{opt})$, entonces $\theta_{opt} = \theta^k$ y $\chi^2(\theta_{opt}) = \chi^2(\theta^k)$.
 - Guardamos los valores de θ_{opt} y $\chi^2(\theta_{opt})$ como los resultados del inicio i .
- Tomamos como vector de parámetros óptimos encontrado θ al primero de los θ resultante de los 200 inicios que minimice a la figura de mérito $\chi^2(\theta)$.

⁶Para más detalles de la modificación ver subsección 5.3.3 (Implementación del método 2).

5.3.2. Implementación del método 1-1

- Para i entre 0 y 199 (cantidad de inicios):
 - Generamos un punto γ^0 inicial usando una distribución uniforme (cuya semilla depende de i) sobre el espacio de parámetros no lineales.
 - Consideramos $y(x|(\theta^L, \gamma^0)) = \sum_j \theta_j^L f_j(x|\gamma^0)$.
Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β^0 . Así, construimos $\theta^0 = (\beta^0, \gamma^0)$.
 - Definimos $\theta_{opt} = \theta^0$ y $\chi^2(\theta_{opt}) = \chi^2(\theta^0)$.
 - Para k entre 0 y 999 (cantidad de iteraciones):
 - Generamos un punto γ^k usando una distribución uniforme sobre el espacio de parámetros no lineales.
 - Consideramos $y(x|(\theta^L, \gamma^k)) = \sum_j \theta_j^L f_j(x|\gamma^k)$.
Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β^k . Así, construimos $\theta^k = (\beta^k, \gamma^k)$.
 - Calculamos $\chi^2(\theta^k)$.
 - Si $\chi^2(\theta^k) < \chi^2(\theta_{opt})$, entonces $\theta_{opt} = \theta^k$ y $\chi^2(\theta_{opt}) = \chi^2(\theta^k)$.
 - Guardamos los valores de θ_{opt} y $\chi^2(\theta_{opt})$ como los resultados del inicio i .
- Tomamos como vector de parámetros óptimos encontrado θ al primero de los θ resultante de los 200 inicios que minimice a la figura de mérito $\chi^2(\theta)$.

5.3.3. Implementación del método 2

La librería *lmfit* permite encontrar θ_{opt} utilizando el método de Levenberg - Marquardt. El θ_{opt} encontrado dependerá del punto inicial θ^0 . Es así que usualmente se prueba manualmente con distintos puntos iniciales hasta obtener θ_{opt} satisfactorios para el problema que se quiere resolver. En este trabajo, proponemos una modificación que contempla la generación de una gran cantidad de puntos iniciales de acuerdo a una distribución uniforme sobre el espacio de parámetros y como resultado elegimos el θ_{opt} que menor $\chi^2(\theta_{opt})$ obtenga. Esta mejora evita la búsqueda manual de un punto inicial que genere resultados satisfactorios de $\chi^2(\theta_{opt})$.

La implementación de este método es la siguiente:

- Para i entre 0 y 39 (cantidad de inicios):
 - Generamos un punto θ^0 inicial usando una distribución uniforme (cuya semilla depende de i) sobre el espacio de parámetros.
 - Pasamos θ^0 y las restricciones sobre los parámetros al método *minimize* de *lmfit* y obtenemos θ_{opt} .
 - Calculamos $\chi^2(\theta_{opt})$.
 - Guardamos los valores de θ_{opt} y $\chi^2(\theta_{opt})$ como los resultados del inicio i .
- Tomamos como vector de parámetros óptimos encontrado θ al primero de los θ resultante de los 39 inicios que minimice a la figura de mérito $\chi^2(\theta)$.

5.3.4. Implementación del método 3

- Para i entre 0 y 199 (cantidad de inicios):
 - Generamos un punto γ^0 inicial usando una distribución uniforme (cuya semilla depende de i) sobre el espacio de parámetros no lineales.
 - Consideramos $y(x|(\theta^L, \gamma^0)) = \sum_j \theta_j^L f_j(x|\gamma^0)$.
Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β^0 . Así, construimos $\theta^0 = (\beta^0, \gamma^0)$.
 - Definimos $\theta_{opt} = \theta^0$ y $\chi^2(\theta_{opt}) = \chi^2(\theta^0)$.
 - Definimos $\theta_{busqueda} = \theta^0$ y $\chi^2(\theta_{busqueda}) = \chi^2(\theta^0)$.
 - Para k entre 0 y 999 (cantidad de iteraciones):
 - Calculamos la temperatura t utilizando el siguiente esquema de autorregulación:
 - ◇ Si $\chi^2(\theta_{opt}) < 10$, entonces $t = \chi^2(\theta_{opt})$.
 - ◇ Si $\chi^2(\theta_{opt}) \geq 10$, entonces $t = \frac{\chi^2(\theta_{opt})}{\log_{10}(\chi^2(\theta_{opt}))}$.
 - Generamos un punto $\gamma^k = (\gamma_0^k, \dots, \gamma_s^k)$ usando para cada $m = 0, \dots, s$ una distribución normal centrada en $(\theta_{busqueda}^{NL})_m$ con desviación estándar $longitud(I_m^{NL}) \times 0.1$.
 - Consideramos $y(x|(\theta^L, \gamma^k)) = \sum_j \theta_j^L f_j(x|\gamma^k)$.
Calculamos los parámetros lineales θ^L ajustando la regresión lineal resultante por cuadrados mínimos y los llamamos β^k . Así, construimos $\theta^k = (\beta^k, \gamma^k)$. (Notamos $\theta_{nuevo} = \theta^k$.)
 - ◇ Si θ_{nuevo} no cae en el espacio de parámetros, entonces procedemos a la siguiente iteración k .
 - ◇ Si θ_{nuevo} cae en el espacio de parámetros entonces:
 - I Calculamos $\chi^2(\theta_{nuevo})$.
 - II Calculamos $\alpha(\theta_{busqueda}, \theta_{nuevo}) = \min \left\{ \exp \left(\frac{\chi^2(\theta_{busqueda}) - \chi^2(\theta_{nuevo})}{t} \right), 1 \right\}$
 - III Generamos u de una distribución uniforme sobre $[0, 1]$.
 - IV Si $u > \alpha(\theta_{busqueda}, \theta_{nuevo})$ (rechazamos el cambio), entonces procedemos a la siguiente iteración k .
 - V Si $u \leq \alpha(\theta_{busqueda}, \theta_{nuevo})$ (aceptamos el cambio), entonces definimos $\theta_{busqueda} = \theta_{nuevo}$ y $\chi^2(\theta_{busqueda}) = \chi^2(\theta_{nuevo})$.
Además, si $\chi^2(\theta_{nuevo}) < \chi^2(\theta_{opt})$, entonces definimos $\theta_{opt} = \theta_{nuevo}$ y $\chi^2(\theta_{opt}) = \chi^2(\theta_{nuevo})$.
 - Guardamos los valores de θ_{opt} y $\chi^2(\theta_{opt})$ como los resultados del inicio i .
- Tomamos como vector de parámetros óptimos encontrado θ al primero de los θ resultante de los 200 inicios que minimice a la figura de mérito $\chi^2(\theta)$.

Es importante notar que para este algoritmo de Simulated Annealing hemos definido la autorregulación de la temperatura en función de $\chi^2(\theta_{opt})$ (a diferencia de los esquemas de enfriamiento tradicionales) y que utilizamos en la búsqueda de los parámetros no lineales una distribución normal⁷ centrada en $(\theta_{busqueda}^{NL})_m$ con desviación estándar $longitud(I_m^{NL}) \times 0.1$.

⁷Ver la sección 7 (Perspectivas) para otra alternativa en la elección de la distribución.

5.4. Comparación de los métodos

Una vez ejecutados los métodos, obtuvimos los resultados que se listan en el Cuadro 1:

- Obtuvimos un conjunto de 200 resultados para los métodos 1, 1_1 y 3 y 40 para el método 2. Es importante notar que los 200 puntos de inicio son los mismos para los métodos 1, 1_1 y 3 mientras que los 40 del método 2 coinciden con los 40 iniciales de los demás métodos, con el objetivo de poder realizar comparaciones entre los resultados obtenidos.
- Según la métrica de evaluación ($\chi^2(\theta_{opt})$) el método que obtuvo mejores resultados fue el 3 con $\chi^2(\theta_{opt}) \simeq 1.51$, seguido por el 2 con $\chi^2(\theta_{opt}) \simeq 2.11$, luego el 1_1 con $\chi^2(\theta_{opt}) \simeq 2.82$ y por último el 1 con $\chi^2(\theta_{opt}) \simeq 8.72 \times e^{12}$.
- El método 1 (Búsqueda Aleatoria sobre todo el espacio de parámetros) no logra resultados comparables con 2 (Levenberg-Marquardt sobre todo el espacio de parámetros) y 3 (Simulated Annealing sobre el espacio de parámetros no lineales). Sin embargo, cuando se modifica el algoritmo y se hace la búsqueda sólo sobre el espacio de los parámetros no lineales, los resultados mejoran notablemente (método 1_1).
- Los métodos que hacen la búsqueda sólo sobre el espacio de los parámetros no lineales y encuentran los parámetros lineales ajustando la regresión lineal resultante por cuadrados mínimos (3 y 1_1) obtienen un $\chi^2(\theta)$ inicial considerablemente menor a los que realizan la búsqueda sobre todo el espacio de parámetros.

Cuadro 1: Resultados de los métodos. Para cada método se fijó un máximo de 1000 iteraciones por cada punto inicial (con una cantidad máxima de 200 puntos iniciales) o de 1200 segundos de tiempo de ejecución total, lo que ocurra primero.

| Método | Reinicios | Iteraciones | χ_{optimo}^2 | $\chi_{inicial}^2$ | Tiempo de ejecución (seg) |
|------------|-----------|-------------|-----------------------|-----------------------|---------------------------|
| Método 1 | 200 | 1000 | $8.716 \times e^{12}$ | $6.515 \times e^{25}$ | 960 |
| Método 1_1 | 200 | 1000 | $2.825 \times e^0$ | $2.931 \times e^1$ | 982 |
| Método 2 | 40 | - | $2.107 \times e^0$ | $4.034 \times e^{21}$ | 1144 |
| Método 3 | 200 | 1000 | $1.510 \times e^0$ | $4.825 \times e^1$ | 607 |

A continuación compararemos el detalle de las iteraciones de los métodos 1, 1_1 y 3. No podremos hacer esto con el método 2 ya que al importar un método de una librería externa no podemos acceder al detalle de las mismas:

- Figura 5: Calcular los parámetros lineales ajustando la regresión lineal produce una mejora considerable en la minimización de $\chi^2(\theta_{opt})$ (notar la diferencia de magnitud de $\chi^2(\theta_{opt})$ entre el método 1 y el método 1_1).
- Figura 6: En la iteración 294 el método 1_1 encuentra un valor de θ que reduce drásticamente el valor de la figura de mérito y que será el θ óptimo para este método. Alrededor de la iteración 500 el método 3 encuentra valores de θ que reducen drásticamente el valor de la figura de mérito y para la iteración 782 el método ya encontró los parámetros que serán sus óptimos.

- Figura 7: El método 1 hace una búsqueda uniforme de los parámetros lineales, como podemos observar para el parámetro θ_0^L en la Figura 7, mientras que los métodos 1.1 y 3 directamente calculan los parámetros lineales ajustando la regresión lineal.
- Figura 8: Los métodos 1 y 1.1 hacen una búsqueda uniforme de los parámetros no lineales, como podemos observar para el parámetro h . El método 3 realiza la búsqueda usando Simulated Annealing.

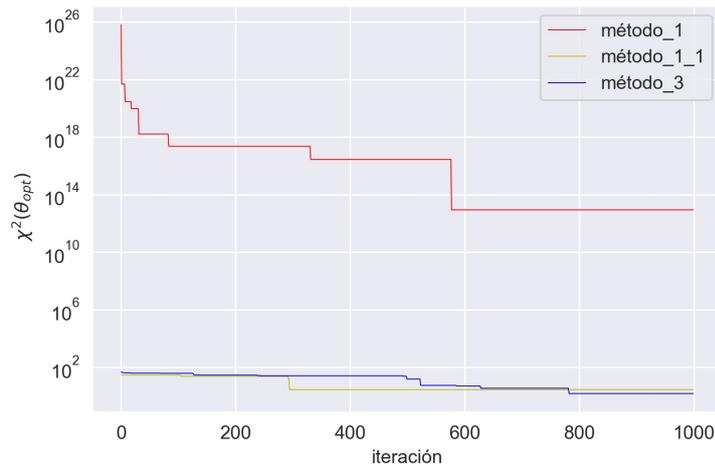


Figura 5: Comparación del comportamiento de $\chi^2(\theta_{opt})$ a lo largo de las iteraciones de los métodos 1, 1.1 y 3. Para una mejor visualización se emplea escala logarítmica en el eje y .

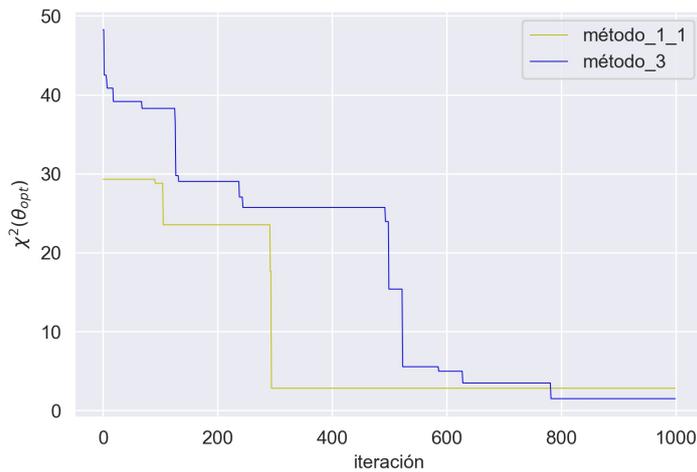


Figura 6: Comparación del comportamiento de $\chi^2(\theta_{opt})$ a lo largo de las iteraciones de los métodos 1.1 y 3.

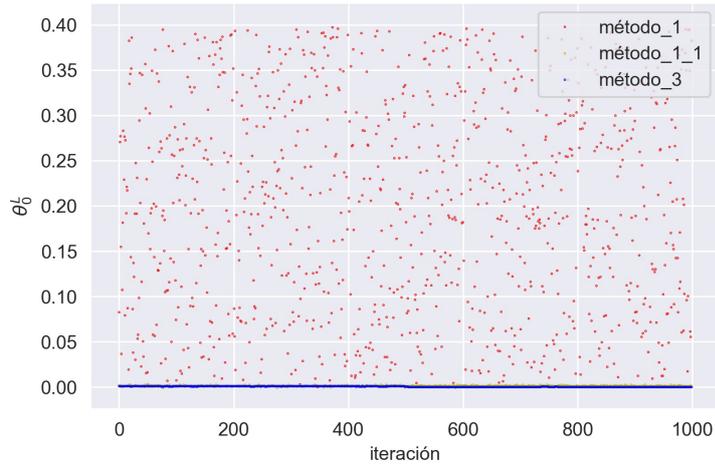


Figura 7: Comparación de la búsqueda del parámetro lineal θ_0^L para cada método.

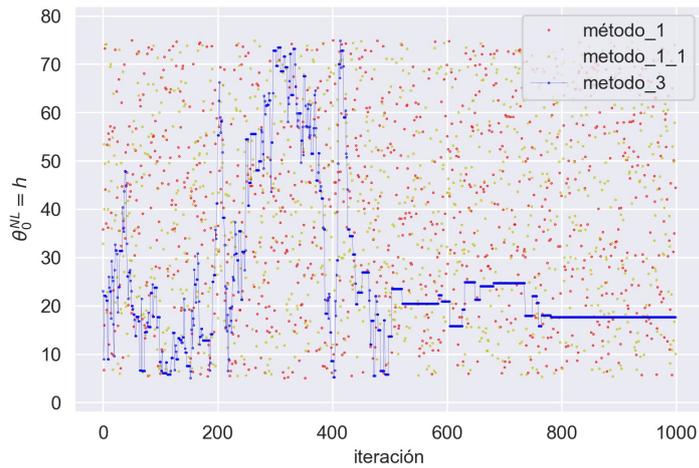


Figura 8: Comparación de la búsqueda del parámetro no lineal $\theta_0^{NL} = h$ para cada método.

5.5. Resultados del Simulated Annealing

Recordemos que para el método 3 obtenemos 200 θ , cada uno correspondiente a los parámetros óptimos encontrados a partir de cada uno de los puntos iniciales. El primero de estos 200 θ que obtenga el menor valor de la figura de mérito será el vector de parámetros óptimos encontrado por el método. Podemos visualizar el vector de parámetros óptimos encontrado por el método 3 en la columna 'optimo' del Cuadro 2. Utilizando estos parámetros óptimos se genera la curva modelo, que se ajusta a los datos experimentales (Figura 9). Por otro lado, utilizando los 200 θ correspondientes a los parámetros óptimos encontrados para cada uno de los 200 puntos iniciales podemos generar una distribución para cada parámetro (ejemplo para el parámetro $\theta_2^{NL} = t_h$, como se muestra

la Figura 10), con sus estadísticos de interés correspondientes (columnas “desviación estándar”, “media”, “mediana”, “mínimo” y “máximo” del Cuadro 2).

Tomando el último 30 por ciento de las iteraciones correspondientes al inicio que llegó a los parámetros óptimos del método 3 generamos un gráfico con las correlaciones entre todas las coordenadas de θ_{nuevo} , junto con $\chi^2(\theta_{nuevo})$ (correlación de Pearson en la Figura 11 y de Spearman en la Figura 12). Estas correlaciones permiten analizar la inclusión de restricciones relacionales entre los parámetros, que resultarán en una reducción de la dimensionalidad del problema.

En cuanto al funcionamiento específico del método 3 podemos hacer algunas apreciaciones:

- Figuras 13 y 14: comparando estas dos figuras se puede ver la diferencia de los valores de la figura de mérito para los parámetros θ_{opt} y $\theta_{busqueda}$. Mientras que θ_{opt} sólo cambiará cuando esto traiga asociado una disminución en el valor de la figura de mérito, $\theta_{busqueda}$ podrá también aceptar cambios que produzcan un incremento en el valor de la figura de mérito. Notar que la aceptación de los $\theta_{busqueda}$ que incrementan el valor de la figura de mérito es mucho más frecuente en las primeras iteraciones, donde el sistema todavía no se ha enfriado. Estos cambios hacia parámetros que incrementan el valor de la figura de mérito son los que posibilitan que el algoritmo no se estanque en mínimos locales.
- Figura 15: a lo largo de las iteraciones los valores de θ_{opt} sólo cambian si hay una mejora (disminución del valor de la figura de mérito). Por otro lado, los valores de $\theta_{busqueda}$ permiten cambios (con probabilidad α) que incrementen el valor de la figura de mérito, logrando que el sistema pueda escapar de mínimos locales. A su vez, la búsqueda de nuevos candidatos θ_{nuevo} se realiza utilizando una distribución normal centrada en $\theta_{busqueda}$, con una desviación estándar del 10 por ciento de la longitud del rango de cada parámetro.
- Figura 16: la probabilidad de aceptación α de nuevos parámetros va cambiando a lo largo de las iteraciones. A medida que el sistema se va enfriando, aceptar cambios que no disminuyan el valor de la figura de mérito se hace cada vez más improbable. Notar que a partir de la iteración 524 (marcada en negro) la temperatura toma valores menores a 10.

Cuadro 2: Resultados estadísticos del método 3 para cada parámetro teniendo en cuenta los 200 inicios. Los parámetros h , w , t_h y t_w se encuentran expresados en Å .

| Parámetro | Óptimo | Desv. estándar | Mediana | Mínimo | Máximo |
|--------------|------------------------|------------------------|------------------------|------------------------|-----------------------|
| θ_0^L | $3.739 \times e^{-5}$ | $3.305 \times e^{-4}$ | $6.535 \times e^{-4}$ | $3.440 \times e^{-6}$ | $1.186 \times e^{-3}$ |
| θ_1^L | $8.954 \times e^{-12}$ | $2.023 \times e^{-10}$ | $5.428 \times e^{-12}$ | $6.155 \times e^{-13}$ | $2.068 \times e^{-9}$ |
| h | $1.765 \times e^1$ | $9.253 \times e^0$ | $1.763 \times e^1$ | $5.241 \times e^0$ | $6.896 \times e^1$ |
| w | $1.801 \times e^2$ | $4.076 \times e^2$ | $4.585 \times e^2$ | $1.850 \times e^1$ | $1.406 \times e^3$ |
| t_h | $2.399 \times e^1$ | $5.511 \times e^0$ | $2.321 \times e^1$ | $6.613 \times e^0$ | $3.692 \times e^1$ |
| t_w | $3.800 \times e^1$ | $2.206 \times e^1$ | $3.792 \times e^1$ | $6.307 \times e^{-1}$ | $9.890 \times e^1$ |
| β | $-7.152 \times e^0$ | $6.217 \times e^0$ | $-8.239 \times e^{-1}$ | $-9.760 \times e^0$ | $9.922 \times e^0$ |

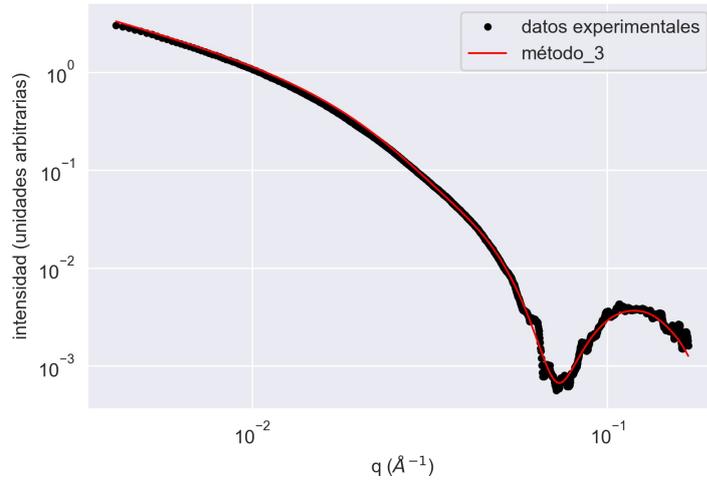


Figura 9: Curva modelo obtenida por el método 3. Para una mejor visualización se emplean escalas logarítmicas en ambos ejes.

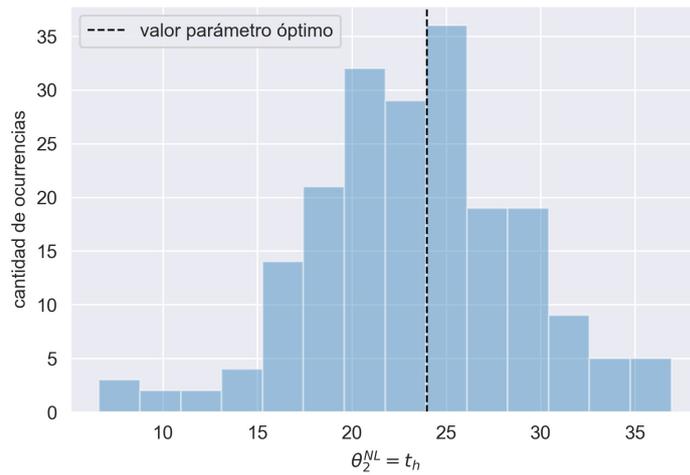


Figura 10: Distribución de los parámetros $\theta_2^{NL} = t_h$ obtenidos teniendo en cuenta los 200 inicios del método 3.

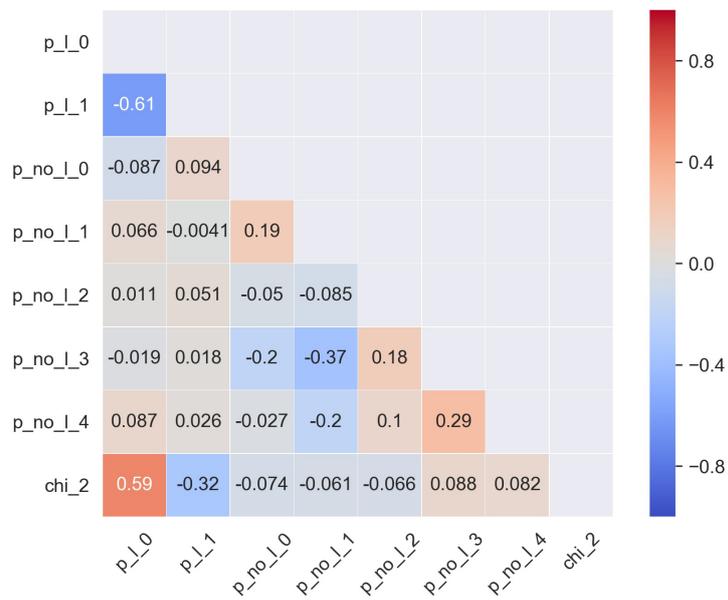


Figura 11: Correlaciones de Pearson para el mejor inicio del método 3.

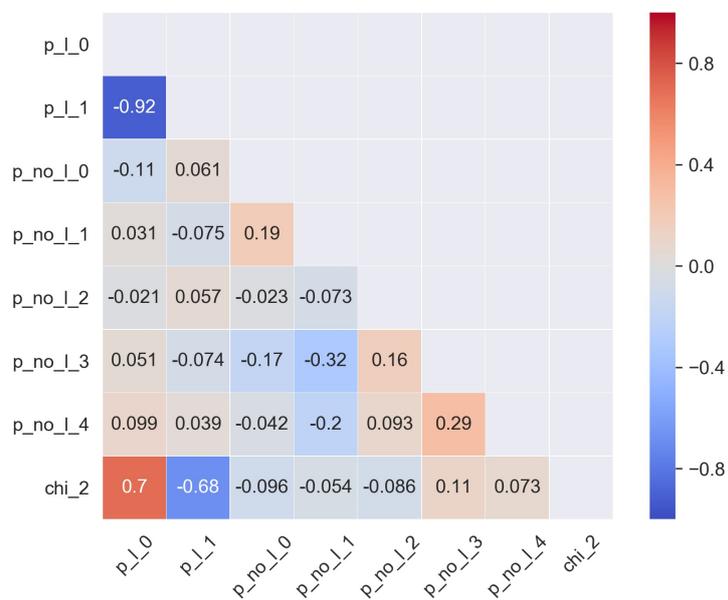


Figura 12: Correlaciones de Spearman para el mejor inicio del método 3.

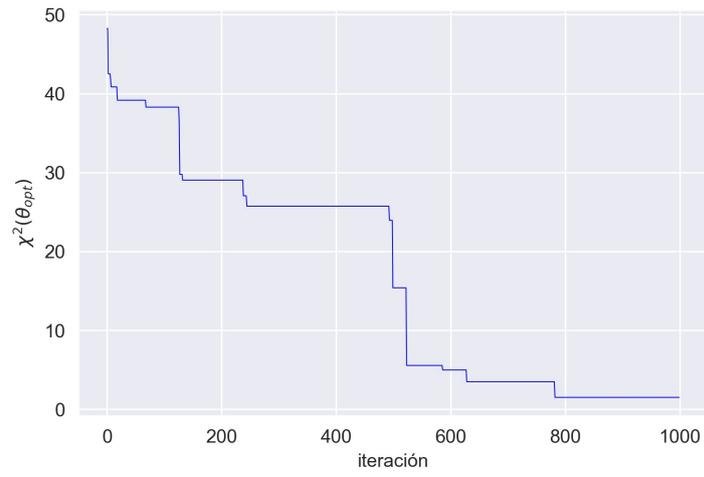


Figura 13: Comportamiento de $\chi^2(\theta_{opt})$ a lo largo de las iteraciones para el método 3.

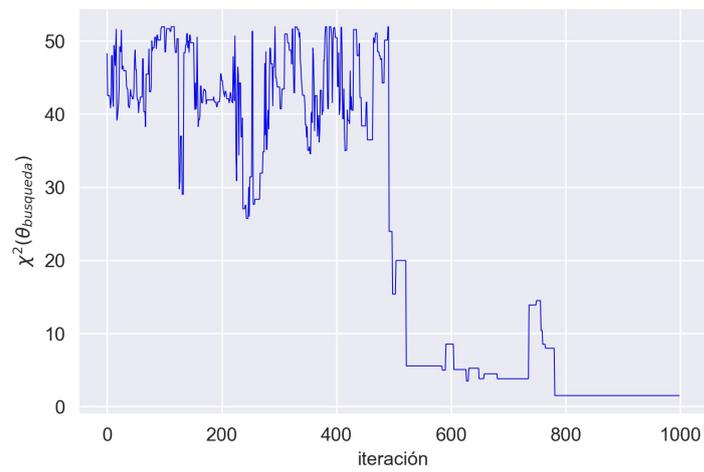


Figura 14: Comportamiento de $\chi^2(\theta_{busqueda})$ a lo largo de las iteraciones para el método 3.

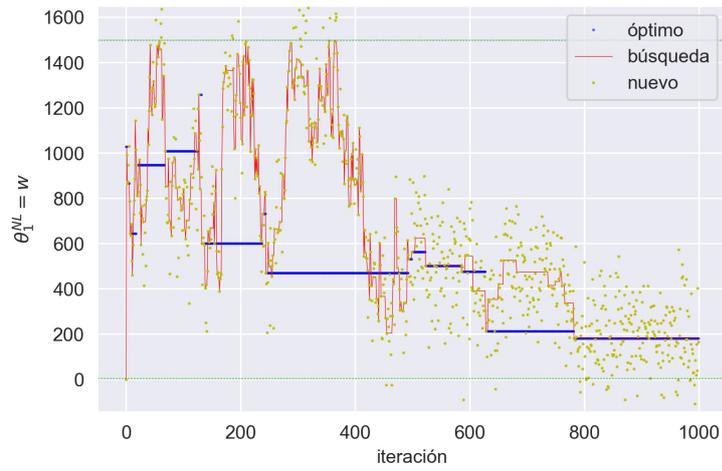


Figura 15: Valores óptimo, búsqueda y nuevo de $\theta_1^{NL} = w$ para el método 3. En verde se pueden ver los límites de I_1^{NL}

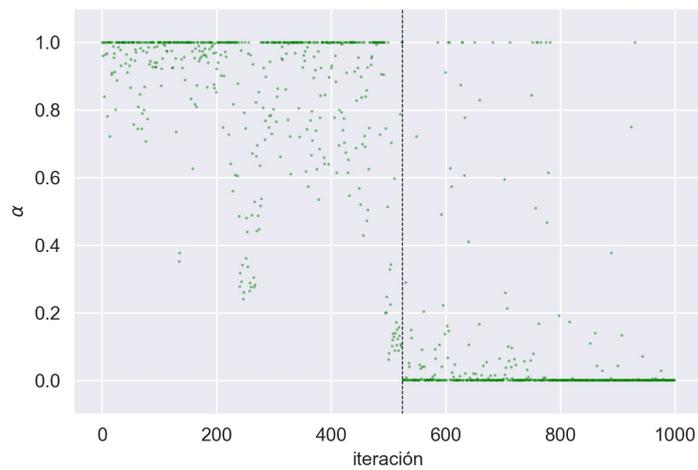


Figura 16: Comportamiento de α (probabilidad de aceptación) a lo largo de las iteraciones para el método 3.

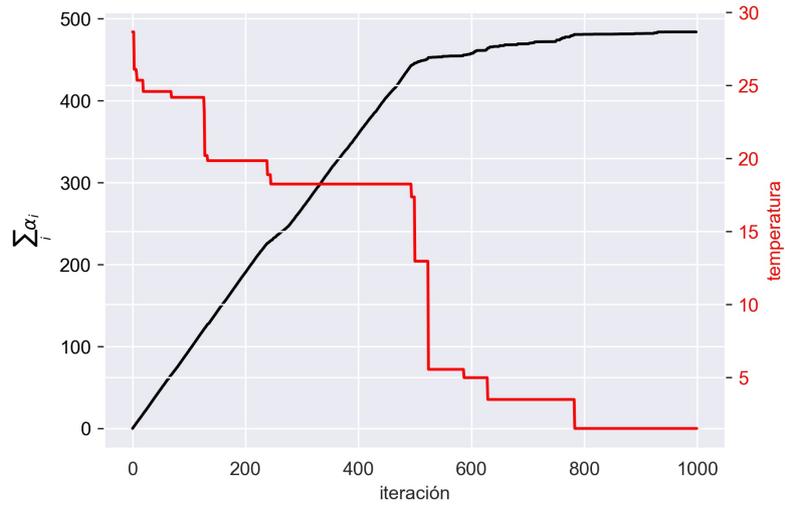


Figura 17: Comportamiento de la temperatura autorregulada (rojo) y de la suma acumulada de α (probabilidad de aceptación) en función de las iteraciones (negro), para el método 3.

6. Conclusiones

Para lograr resolver el problema de encontrar los parámetros óptimos θ asociados a un modelo no lineal $y(x|\theta)$ que se ajuste a datos experimentales mediante la minimización de la figura de mérito $\chi^2(\theta)$ hemos estudiado y desarrollado una serie de métodos basados en los métodos de Monte Carlo. Para su implementación se crearon una serie de algoritmos programados en Python, y se utilizaron para resolver problemas provenientes de las metodologías de caracterización de materiales, en este caso de dispersión de rayos X a bajos ángulos, como modelo de estudio.

A diferencia de los métodos basados en gradientes, pudimos resolver el problema de optimizar la figura de mérito sin tener que recurrir al cálculo de derivadas, ya sea de forma analítica o numérica.

La modificación que realizamos en la aplicación del método de Levenberg-Marquardt permitió automatizar la búsqueda de puntos iniciales que lleven a resultados razonables, algo que los usuarios de este tipo de métodos suelen llevar a cabo manualmente.

Los resultados de los algoritmos de búsqueda mejoraron notablemente con la separación de las contribuciones lineales de las contribuciones no lineales, debido a que el espacio de búsqueda se reduce drásticamente. Además de lograr un mejor ajuste a los datos experimentales, esta separación permite el cálculo automático de los parámetros lineales (mediante rutinas de cuadrados mínimos lineales) sin necesidad de tener que contar con un conocimiento tan exhaustivo de sus rangos de búsqueda.

La autorregulación de la temperatura en función del valor de la figura de mérito evita la construcción manual de esquemas de enfriamiento, logrando automatizar este proceso que suele ser poco generalista y atado a la experimentación con prueba y error para el problema específico que se esté tratando.

Todo el código de programación escrito podrá ser reutilizado para la resolución del tipo de problemas tratados en este trabajo y quedará a disposición para futuras mejoras.

7. Perspectivas

- Muestreo con densidades de probabilidad simétricas y asimétricas con regulación de varianza:
La propuesta es optimizar el método planteado inicialmente por ([da Silva Costa et al., 2015]), manteniendo la separación de variables lineales de las no lineales. Se procederá a realizar algoritmos de búsqueda, acotados, mediante funciones de probabilidad simétricas y asimétricas en un intervalo acotado. Cada parámetro no lineal θ_j^{NL} será perturbado de forma aleatoria mediante una distribución de probabilidad normal con desviación estándar σ_j , tal como se describió en párrafos previos. En cada perturbación los parámetros lineales se recalculan y se emplea alguna figura de mérito estadística para decidir si se acepta o no el cambio. Como el espacio de muestreo se acota según $\theta_j^{NL} \pm 2\sigma_j$, se propondrán análisis estadísticos para controlar el ancho de distribución a lo largo del proceso, equivalente a un Simulated Annealing, para reducir el espacio de muestreo alrededor del mínimo hallado. Se propone comenzar la rutina de ajuste con muestreo Gaussiano e inducir asimetrías en la distribución mediante análisis estadísticos (ej. elucidación de a priori) y controlar individualmente los valores de la varianza de cada parámetro (ej. matriz de covarianzas) en función de la respuesta que genera cada perturbación a la figura de mérito de estudio -*Machine Learning*-. De esta manera se pretende restringir el número de muestreos innecesarios.
- La rutina de autorregulación de la temperatura fue una prueba de concepto exitosa. Se pueden estudiar otras rutinas de autorregulación que conduzcan a mejores resultados.
- En cada iteración, al realizar la búsqueda de los parámetros, si alguno de estos no pertenece a los rangos de búsqueda definidos se procede a descartar dicha iteración y se prosigue con la próxima iteración. Este comportamiento puede generar que se descarten una gran cantidad de iteraciones, sobre todo cuando algún parámetro se encuentra cercano al borde de su rango de búsqueda.
- El código del método 3 permite definir una etapa de *burn-in* (esto quiere decir que durante cierta cantidad de iteraciones se mantiene una temperatura fija), sólo que no lo implementamos en este trabajo. Esto podría llevar a mejores resultados, ya que debido a la mejora de la separación en términos lineales y no lineales el valor inicial de la figura de mérito comienza bastante bajo, con lo que el sistema se enfría rápidamente y no llega a explorar por demasiadas iteraciones en todo el espacio.
- En todos los métodos (excepto el 2) la interrupción del algoritmo se da por la cantidad de iteraciones. Hay otras formas de hacer esto, como la tolerancia ($|\chi^2(\theta_{opt}) - 1|$) menor a cierto valor prefijado o la razón de cambio de $\chi^2(\theta_{opt})$. Esto permitiría agregar otros criterios de corte de los algoritmos, logrando una mayor automatización del proceso para el usuario final.
- Estudiar el escalado del tiempo de máquina con el número de parámetros no lineales.
- Estudiar la velocidad de convergencia del método de Simulated Annealing en función de la distribución utilizada para realizar la búsqueda (ej. Uniforme, Gaussiana, Lognormal, etc.)

8. Anexo: definiciones

Sea $C = (X^0, X^1, \dots)$ una sucesión de variables aleatorias $X^n : \Omega \rightarrow S$. En esta sección consideraremos $S = \mathbb{R}^M$.

C es una *cadena de Markov* si $\forall n \geq 0, x \in \mathbb{R}^M, A, A_0, \dots, A_{n-1} \in \mathcal{B}(\mathbb{R}^M)$ ⁸ se cumple:

$$P(X^{n+1} \in A | X^n = x, X^{n-1} \in A_{n-1}, \dots, X^0 \in A_0) = P(X^{n+1} \in A | X^n = x) \quad (20)$$

Llamaremos a S el *conjunto de estados* de la cadena de Markov C .

Una cadena de Markov es *homogénea* si $\forall n \geq 0, x \in \mathbb{R}^M, A \in \mathcal{B}(\mathbb{R}^M)$ se cumple:

$$P(X^{n+1} \in A | X^n = x) = P(X^1 \in A | X^0 = x) \quad (21)$$

En este trabajo sólo consideraremos cadenas de Markov homogéneas ⁹.

Definición 8.1. Decimos que una función $\mathcal{P} : \mathbb{R}^M \times \mathcal{B}(\mathbb{R}^M) \rightarrow [0, 1]$ es un núcleo de transición de C si $\forall x \in \mathbb{R}^M, A \in \mathcal{B}(\mathbb{R}^M)$:

$$\mathcal{P}(x, A) = P(X^1 \in A | X^0 = x) \quad (22)$$

Decimos que una función $p : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^M$ es función de densidad de \mathcal{P} si $\forall x \in \mathbb{R}^M, A \in \mathcal{B}(\mathbb{R}^M)$:

$$\mathcal{P}(x, A) = \int_A p(x, y) dy \quad (23)$$

Definición 8.2. Dado $m \geq 1$, decimos que una función $\mathcal{P}^m : \mathbb{R}^M \times \mathcal{B}(\mathbb{R}^M) \rightarrow [0, 1]$ es un núcleo de transición de C sobre m pasos si $\forall x \in \mathbb{R}^M, A \in \mathcal{B}(\mathbb{R}^M)$:

$$\mathcal{P}^m(x, A) = P(X^m \in A | X^0 = x) \quad (24)$$

Decimos que una función $p^m : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^M$ es función de densidad de \mathcal{P}^m si $\forall x \in \mathbb{R}^M, A \in \mathcal{B}(\mathbb{R}^M)$:

$$\mathcal{P}^m(x, A) = \int_A p^m(x, y) dy \quad (25)$$

Definición 8.3. Decimos que una función $\pi : \mathbb{R}^M \rightarrow \mathbb{R}$ representa una distribución de probabilidad si $(\forall x \in \mathbb{R}^M, \pi(x) \geq 0)$ y $\int \pi(x) dx = 1$.

Notamos π^n , para cada $n \geq 0$, a la distribución de probabilidad de la cadena en el tiempo n . (Es decir, $P(X^n \in A) = \int_A \pi^n(x) dx$).

Con estas notaciones, tenemos que:

$$\pi^n(y) = \int_{\mathbb{R}^M} \pi^{n-1}(x) p(x, y) dx = \int_{\mathbb{R}^M} \pi^0(x) p^n(x, y) dx \quad (26)$$

⁸Notamos $\mathcal{B}(\mathbb{R}^M)$ a los borelianos de \mathbb{R}^M .

⁹El tema de cadenas de Markov inhomogéneas excede el propósito de este trabajo. Para mayores detalles consultar ([Sarymsakov, 1961]) y ([Müller et al., 2004]).

Definición 8.4. C es irreducible si $\forall x \in \mathbb{R}^M, A \in \mathcal{B}(\mathbb{R}^M), \exists n \geq 1$ tal que $P^n(x, A) > 0$.

C resultará aperiódica si para cada $x \in \mathbb{R}^M, \exists \varepsilon_x > 0$ tal que $P(x, B_{\varepsilon_x}(x)) > 0$.

Definición 8.5. Una distribución de probabilidad π es *estacionaria de C* si

$$\pi(y) = \int_{\mathbb{R}^M} \pi(x)p(x, y)dx \quad (27)$$

Referencias

- [Bevington et al., 1993] Bevington, P. R., Robinson, D. K., Blair, J. M., Mallinckrodt, A. J., and McKay, S. (1993). Data reduction and error analysis for the physical sciences. *Computers in Physics*, 7(4):415–416.
- [Björck, 1990] Björck, A. (1990). Least squares methods. *Handbook of numerical analysis*, 1:465–652.
- [Chib and Greenberg, 1995] Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335.
- [Craievich, 2016] Craievich, A. F. (2016). Small-angle X-ray scattering by nanostructured materials. *Handbook of Sol-Gel Science and Technology*, pages 1–46.
- [da Silva Costa et al., 2015] da Silva Costa, D., Huck-Iriart, C., Kellermann, G., Giovanetti, L. J., Craievich, A. F., and Requejo, F. G. (2015). In situ study of the endotaxial growth of hexagonal CoSi₂ nanoplatelets in Si (001). *Applied Physics Letters*, 107(22):223101.
- [Del Moral and Miclo, 1999] Del Moral, P. and Miclo, L. (1999). On the convergence and applications of generalized simulated annealing. *SIAM Journal on Control and Optimization*, 37(4):1222–1250.
- [Gamerman and Lopes, 2006] Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC.
- [Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.
- [Godambe and Heyde, 2010] Godambe, V. P. and Heyde, C. C. (2010). Quasi-likelihood and optimal estimation. In *Selected works of cc heyde*, pages 386–399. Springer.
- [Guzdial and Ericson, 2016] Guzdial, M. and Ericson, B. (2016). *Introduction to computing and programming in python*. Pearson.
- [Haario and Saksman, 1991] Haario, H. and Saksman, E. (1991). Simulated annealing process in general state space. *Advances in Applied Probability*, 23(4):866–893.
- [Häggström, 2002] Häggström, O. (2002). *Finite Markov chains and algorithmic applications*, volume 52. Cambridge University Press.
- [Hansen, 1994] Hansen, P. C. (1994). Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Numerical algorithms*, 6(1):1–35.
- [Hastings, 1970] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications.
- [Huck-Iriart et al., 2016] Huck-Iriart, C., De-Candia, A., Rodriguez, J., and Rinaldi, C. (2016). Determination of surface tension of surfactant solutions through capillary rise measurements: An image-processing undergraduate laboratory experiment. *Journal of Chemical Education*, 93(9):1647–1651.

- [Ingber, 1993] Ingber, L. (1993). Simulated annealing: Practice versus theory. *Mathematical and computer modelling*, 18(11):29–57.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.
- [Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168.
- [Madsen et al., 2004] Madsen, K., Nielsen, H. B., and Tingleff, O. (2004). Imm methods for non-linear least squares problems.
- [Marcolongo and Mirenda, 2011] Marcolongo, J. P. and Mirenda, M. (2011). Thermodynamics of sodium dodecyl sulfate (SDS) micellization: an undergraduate laboratory experiment. *Journal of Chemical Education*, 88(5):629–633.
- [Marquardt, 1963] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441.
- [Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.
- [Müller et al., 2004] Müller, P., Sansó, B., and De Iorio, M. (2004). Optimal Bayesian design by inhomogeneous Markov chain simulation. *Journal of the American Statistical Association*, 99(467):788–798.
- [Newville et al., 2014] Newville, M., Stensitzki, T., Allen, D. B., and Ingargiola, A. (2014). *LM-FIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python*, doi: 10.5281/zenodo.11813.
- [Pan, 2012] Pan, W.-T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26:69–74.
- [Pedersen, 1997] Pedersen, J. S. (1997). Analysis of small-angle scattering data from colloids and polymer solutions: modeling and least-squares fitting. *Advances in colloid and interface science*, 70:171–210.
- [Pelikan et al., 1999] Pelikan, M., Goldberg, D. E., and CantÃ³-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, pages 525–532. Morgan Kaufmann Publishers Inc.
- [Pilgrim and Willison, 2009] Pilgrim, M. and Willison, S. (2009). *Dive Into Python 3*, volume 2. Springer.
- [Pollock et al., 2016] Pollock, M., Fearnhead, P., Johansen, A., and Roberts, G. O. (2016). An unbiased and scalable Monte Carlo method for Bayesian inference for big data. In *preparation*. see <http://www.birs.ca/events/2015/5-day-workshops/15w5160/videos/watch/201506021533-Roberts.html>.

- [Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.
- [Robert and Casella, 2013] Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.
- [Robert et al., 2010] Robert, C. P., Casella, G., and Casella, G. (2010). *Introducing monte carlo methods with r*, volume 18. Springer.
- [Sarymsakov, 1961] Sarymsakov, T. A. (1961). Inhomogeneous Markov chains. *Theory of Probability & Its Applications*, 6(2):178–185.
- [Smith and Roberts, 1993] Smith, A. F. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23.
- [Spall, 2005] Spall, J. C. (2005). *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons.
- [Tsallis and Stariolo, 1996] Tsallis, C. and Stariolo, D. A. (1996). Generalized simulated annealing. *Physica A: Statistical Mechanics and its Applications*, 233(1-2):395–406.
- [Van Laarhoven and Aarts, 1987] Van Laarhoven, P. J. and Aarts, E. H. (1987). Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer.
- [Wang et al., 2019] Wang, A. Q., Kolb, M., Roberts, G. O., and Steinsaltz, D. (2019). Theoretical properties of quasi-stationary Monte Carlo methods. *The Annals of Applied Probability*, 29(1):434–457.