

**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**Generador Virtual de Microestructuras Policristalinas**

**Juan Carlos Guerra**

**David Bracho**

Tesis de grado presentada como requisito para la obtención del título de Ingeniero Mecánico

Quito, Mayo 2009

Agradecimiento a nuestros profesores:

Edisón Bonifaz

Laurent Sass

Martin Hirschorn

Victor Guerrero

Que son grandes personas, maestros y amigos

© Derechos de autor

Juan Carlos Guerra

David Bracho

2009

## **Resumen**

Un generador virtual de microestructuras policristalinas se desarrolla en base a la investigación del método de MonteCarlo-Potts. Se inicia definiendo los conceptos básicos necesarios para entender los fenómenos de evolución microestructural. La investigación incluye una descripción de los fenómenos termodinámicos involucrados durante el crecimiento de grano. A continuación se detallan los parámetros requeridos por el algoritmo que resulta del método MonteCarlo-Potts. Se incluye los diferentes tipos de dominio que pueden utilizarse para realizar la simulación de evolución microestructural, así como diferentes tipos de condiciones de borde. El Hamiltoniano y la función de Metrópolis son analizados como parámetros importantes para modificar el comportamiento del generador virtual. Se explica en detalle la relación de estos parámetros con los conceptos de movilidad y energía del borde de grano. La programación del generador virtual se realiza utilizando lenguaje MATLAB. Además, se utiliza el paquete computacional COMSOL Multiphysics para simular recocido. En base a los resultados obtenidos se propone un método para analizar la evolución microestructural en un tratamiento térmico sin necesidad de experimentación. Adicionalmente, se desarrollan funciones auxiliares para visualización y validación de los resultados. Finalmente se incluyen ejemplos de la manera en que puede adaptarse el generador virtual de microestructuras policristalinas para simular fenómenos físicos reales: comportamiento de fases y gradientes de temperatura en una muestra.

## **Abstract**

A virtual generator for polycrystalline microstructures is developed based on the MonteCarlo-Potts method. First, basic concepts on microstructure evolution are explained. This research includes an explanation of the thermodynamic phenomena involved in grain growth. Next, the parameters required by the MonteCarlo-Potts algorithm are detailed. Different types of geometries that can be applied in the simulation are also described. Some types of border conditions are explained as well. The Hamiltonian and the Metropolis function are analyzed given their importance in simulation behavior. The impact on movility and boundary energy caused by modifications in both parameters is also described. The virtual generator was programmed in MATLAB environment, coded in M language. Additionally, an annealing treatment is simulated applying the heat transfer module in COMSOL Multiphysics. Using the results obtained with COMSOL, a method is proposed to analyze microstructure evolution during heat treatment without real experimentation. Other functions are programmed in MATLAB language to help visualize and validate simulation results. Finally, examples are included describing ways in which the program can be modified to simulate phase transformation and temperature gradients in samples.

# Contenido

---

---

	<b>Páginas</b>
<b>1. Introducción</b>	2
1.1. Definición de una microestructura policristalina	3-4
1.2. Evolución microestructural	4-5
1.3. Simulación de la evolución microestructural	5-6
1.4. Trabajos previos	7
<b>2. Termodinámica de la microestructura</b>	8
2.1. Primera Ley de la Termodinámica	8-10
2.2. Segunda Ley de la Termodinámica	10-13
2.3. Las ecuaciones de Gibbs y la energía libre de Gibbs	13-15
2.4. Equilibrio termodinámico y la función de Gibbs	16-17
2.5. Termodinámica aplicada a la microestructura	17-20
<b>3. Método de Monte Carlo</b>	21
3.1. Probabilidad y el método de Monte Carlo	21-23
3.2. El Hamiltoniano	23-24
3.3. Tipos de redes cristalográficas que se pueden utilizar	25-26
3.4. Condiciones de frontera	27-31
3.5. Algoritmo del modelo y las funciones de probabilidad utilizadas	32-36
3.6. Variando la movilidad del borde de grano.	37-39

3.7. Agregando componentes a la simulación.	39-40
<b>4. Análisis y Resultados del programa MicroSGEN realizado realizado para la evolución microestructural</b>	41-42
4.1. Programa codificado en Matlab 7.1 <sup>®</sup> llamado MicroSGen	42-57
4.2. Análisis y resultados obtenidos con el programa MicroSGen	57-62
<b>5. Simulación con COMSOL y MicroSGEN</b>	
5.1. Definición de tratamientos térmicos	63-64
5.2. Recocido	64-65
5.3. Simulación de recocido utilizando COMSOL Multiphysics	65-74
5.4. Resultados obtenidos con temperaturas y movilidad	75-79
5.5. Simulado efectos de gradiente de temperatura	79-80
5.6. Simulando un material con dos componentes o fases	80-82
<b>6. Conclusiones</b>	83
<b>Referencias</b>	84-85

# 1

## INTRODUCCIÓN

La ciencia e ingeniería de los materiales estudia la relación lineal que existe entre el procesamiento, la estructura, las propiedades y el desempeño. Si se desea que un material tenga un cierto desempeño, es necesario un riguroso procesamiento para determinar la estructura y propiedades del material. Se entiende que dicha estructura logrará las propiedades que cumplan con el objetivo propuesto. Hace no muchos años atrás, para el diseño de nuevos materiales, esta relación se la alcanzaba únicamente mediante ensayos experimentales de prueba y error. Solo de esa forma se podían reunir los conocimientos para obtener procesos de fabricación, estructura, propiedades y desempeño deseados. En la actualidad, las computadoras permiten realizar simulaciones para evitar y/o reemplazar lo que antes era un trabajo tedioso.



La evolución de microestructuras metálicas ha sido estudiada desde hace un siglo atrás. Se ha encontrado que no son suficientes los modelos analíticos utilizados para determinar las propiedades de los granos metálicos bajo determinadas condiciones. Es fundamental entender la evolución de la microestructura en el tiempo, y para ello una simulación computacional gráfica revela información más completa que cualquier modelo analítico.

Existen varios métodos para simular la evolución de microestructuras policristalinas. Estos métodos van desde algunos muy simples hasta otros que requieren de alta complejidad matemática. Sin embargo incluso los modelos más simples necesitan ser modificados para alcanzar resultados semejantes a las observaciones reales. El método de Monte Carlo Q Potts [ 6 ] es un ejemplo de estos. Métodos mucho más complejos son los que involucran cambios de fase y el modelado de difusión del estado sólido entre otros.

## **1.1 Definición de una Microestructura Policristalina**

Para los objetivos de nuestro estudio, consideramos las propiedades de un material divididas en dos escalas: la escala macroscópica y la escala microscópica. Por lo general, las escalas macroscópicas son las que pueden apreciarse a simple vista. Por otro lado las propiedades microscópicas son propiedades de las estructuras que solo aparecen bajo un microscopio. Todas las estructuras que se observan bajo el microscopio se conocen como microestructuras.

Cuando se habla de microestructuras generalmente se habla de defectos de dos o más dimensiones. Entre estos defectos encontramos los bordes de grano. Un grano está

formado por un solo cristal que puede poseer defectos de cero, o de una dimensión. Los bordes de grano se forman en la frontera de diferentes granos. Cada grano está caracterizado por su respectiva orientación cristalográfica.

Los granos son las estructuras que caracterizan a los metales y determinan muchas de sus propiedades mecánicas. Por esto la microestructura de los metales se la describe como policristalina, es decir, un conjunto de granos con diferentes orientaciones cristalográficas.

Existen tres diferentes tipos de granos: los granos dendríticos, los granos equiaxiales, y los granos alargados. Los granos dendríticos son producto de una solidificación. Este tipo de granos tiene una forma alargada y con ramificaciones. Los granos equiaxiales aparecen cuando se ha producido re-cristalización de un metal que ha sido trabajado en frío. Este tipo de granos también pueden aparecer cuando el metal se ha solidificado de manera lenta o de manera extremadamente rápida. Estos granos tienen una forma que podría describirse como circular. Finalmente existen los granos alargados que son producto del trabajo en frío de los granos equiaxiales y de materiales que resultan de fundición cuando la pieza se ha solidificado a una velocidad intermedia [ 8 ]. En la *Figura 1.1* se pueden apreciar los distintos tipos de grano mencionados.

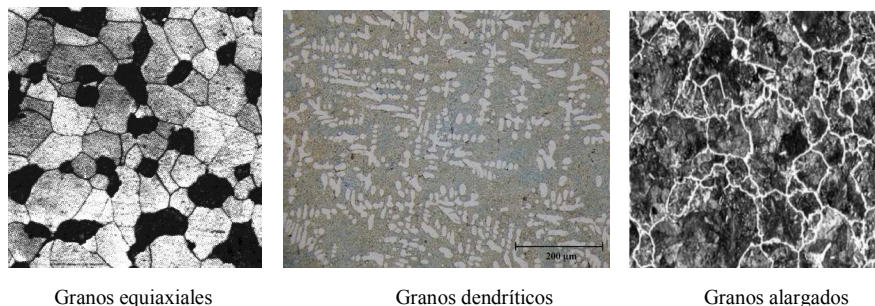


Figura 1.1 Tipo de Granos

Como se ha mencionado, los bordes de grano tienen importante influencia en las propiedades mecánicas del material. Un ejemplo de la manera en que los bordes de grano afectan la estructura es su efecto en el movimiento de los átomos. Se pueden considerar dos tipos de movimientos, uno ordenado y otro aleatorio [ 8 ]. Un desplazamiento

cristalográfico es un movimiento ordenado, mientras que la difusión es un movimiento aleatorio. En los bordes de grano el desplazamiento se bloquea y existe preferencia para la difusión. Por esto, los metales con granos de menor tamaño presentan mayor dureza que los que contienen granos de gran tamaño. El desplazamiento cristalográfico se bloquea y esto aumenta la resistencia y dureza del material.

## **1.2 Evolución Microestructural**

La evolución microestructural es la forma en la que microestructuras dentro de un material se transforman. Estas transformaciones pueden producirse en la forma de la microestructura, en la fase de la microestructura, o en el tamaño de la misma [ 7 ]. Las siguientes transformaciones que están involucradas en el procesamiento de materiales metálicos:

1. Cuando el material fundido se solidifica. Este tipo de cambio en la microestructura se observa principalmente en los trabajos de fundición y moldeo. En este caso el material pasa de una fase líquida a una fase sólida. La solidificación es una transformación de fase.
2. Transformación de fase en estado sólido. En este caso las redes cristalográficas cambian. Ese tipo de transformaciones ocurren durante diferentes tratamientos térmicos.
3. La difusión es otro proceso de transformación que puede sufrir una microestructura. La difusión se puede entender como una migración de átomos. A medida que aumenta la temperatura el proceso de difusión ocurre con mayor rapidez y es importante en el procesamiento de metales tales como: re-cristalización, crecimiento de grano en trabajo en frío, soldadura y tratamientos térmicos del acero.
4. Otro proceso que logra cambiar la microestructura es la re-cristalización y el crecimiento de grano. Estos procesos tienen relación con el proceso de difusión y

pueden ser modelados fácilmente por medio del método de Monte Carlo Potts [ 7 ].

### **1.3 Simulación de la evolución microestructural**

Hoy en día la tecnología permite fabricar materiales de todo tipo para satisfacer las necesidades de la mayoría de aplicaciones. Para fabricarlos se requiere modificar las propiedades a voluntad. Muchas de las propiedades que determinan el comportamiento de los materiales son afectadas por su microestructura. Por ejemplo, en lo que concierne a propiedades mecánicas un menor tamaño promedio de grano resulta en un metal con mayor dureza. El propósito de la simulación de la evolución microestructural es analizar el proceso que logra la microestructura deseada e imitarlo en un proceso real.

Para realizar un modelo computacional adecuado que genere la simulación de microestructuras policristalinas se debe considerar lo siguiente:

1. Entender el modelo físico de los procesos que se van a simular.
2. Programar adecuadamente el modelo físico con las teorías conocidas. Se debe incluir en el programa ecuaciones encontradas para predecir el comportamiento de la microestructura.
3. Lograr que el programa simule adecuadamente el modelo físico tomando en cuenta las leyes físicas reales. Es decir, debe entenderse las variables entregadas por el programa como variables físicas reales y tener una relación coherente. Por ejemplo, en un programa es posible hablar de ciclos y bucles que deben ser relacionados con medidas de tiempo reales. Podría considerarse que dos ciclos sean equivalentes a un segundo real. Estas relaciones deben ser encontradas cuidadosamente tanto en el programa como en experimentos.

4. Adicionalmente a la simulación necesaria para encontrar la microestructura del material sería muy útil realizar un programa para medir las propiedades físicas que resultan de la microestructura hallada. Es decir, el modelo computacional podría entregar una idea del comportamiento real del material y no solo figuras de la microestructura [ 7 ].

Existen varios métodos para simular los diferentes fenómenos que transforman la microestructura. Esta investigación científica se enfoca en el método de Monte Carlo Potts. Se debe aclarar que con este método se logra una aproximación adecuada de los fenómenos de re- cristalización y crecimiento de grano. La simulación basada en autómatas celulares, modelos de campo de fase (Phase Field Model) son otras aproximaciones más avanzadas. El método de Monte Carlo Potts es un método fácil de comprender pero que exige gran nivel de detalle para lograr una simulación que se aproxime correctamente a la realidad.

#### 1.4 Trabajos previos

Se han realizado algunos artículos que ponen en práctica el método de Monte Carlo Potts. La mayoría de los artículos encontrados explican el método en detalle. Estos artículos describen el algoritmo utilizado en la simulación. Los trabajos que se han investigado incluyen los artículos de *Acharjee, Swagato*, “*Grain growth simulation using Monte-Carlo methods*” y de *Blikstein, Paulo, Tschiptschin André*, “*Monte Carlo Simulation of Grain Growth*”. Este segundo artículo tiene un mayor alcance mostrando gráficas del crecimiento del tamaño de grano en el tiempo.

Otros artículos intentan ir más lejos utilizando el método para predecir el comportamiento de materiales reales. El artículo “*Monte Carlo Simulation of Heat-Affected Zone Microstructure in Laser-Beam Welded Nickel Sheet*” de *Li y Asibu* propone un método para conectar la simulación con los procesos reales de crecimiento de grano.

También intenta relacionar el tamaño de grano con las propiedades características del material estudiado.

Adicionalmente se han publicado muy recientemente algunos libros que recogen los métodos computacionales más aceptados para simulación de la evolución microestructural como el mencionado método de Monte Carlo, el método con el autómata celular y los modelos de campo de fase.

Sin embargo, en los métodos de los diferentes artículos y textos se pueden encontrar pequeñas diferencias lo que sugiere que los métodos siguen investigándose para alcanzar mejores aproximaciones a la realidad. La elaboración de modelos computacionales para simulación de microestructuras es más bien reciente y para comprender el tema todavía es necesario agrupar y analizar trabajos previos.

## 2

### **TERMODINÁMICA DE LA MICROESTRUCTURA**

En el análisis de microestructuras las leyes que explican los cambios y evolución de ellas son las leyes de la termodinámica. El tamaño, distribución, y número de granos está determinado por el intercambio de energía y depende principalmente de las llamadas variables de estado como son la temperatura, la presión, el volumen y el peso molar [ 3 ]. Para crear y entender el algoritmo que simula la evolución de la microestructura se debe comprender la teoría termodinámica y especialmente aquella que tiene que ver con soluciones y difusión.

Hoy en día existen paquetes computacionales que incluyen modelos termodinámicos CT (Computational Thermodynamics). Este tipo de paquetes calcula las fases estables de un material dependiendo de la temperatura, presión y los componentes

químicos [ 7 ]. Estos resultados podrían aplicarse para mejorar las simulaciones de evolución microestructural.

A continuación se detallan las leyes termodinámicas y su relación con la evolución de microestructuras.

## 2.1 Primera Ley de la Termodinámica

En termodinámica debe primero definirse el objeto de estudio al que se conoce como sistema. Un sistema puede ser abierto o cerrado dependiendo de sus características. En el caso en que la masa no puede fluir a través de las fronteras del sistema, éste se conoce como sistema cerrado. Cuando la masa sí puede atravesar las fronteras del sistema se habla de un volumen de control o sistema abierto. Se puede describir al sistema conformado por la microestructura de un metal como un sistema cerrado. El sistema cerrado si bien no permite el ingreso y egreso de masa, permite el intercambio de energía con los alrededores.

Si se considera a un sistema invariante, es posible medir las propiedades del mismo y describir su condición a través de estas propiedades. Esta condición se define como un estado del sistema [ 3 ]. Si alguna de las propiedades cambia, el sistema cambia de estado. Las propiedades que definen al estado se conocen como las variables de estado. Estas variables son: presión, volumen, temperatura, y composición química (P, V, T, N). La composición química N por lo general es un vector con elementos que indican la cantidad de sustancia de cada componente químico dentro del sistema.

La primera ley o principio de la conservación de la energía indica que *“toda la energía no puede ser creada ni destruida durante un proceso solo puede cambiar de forma”* [ 3 ]. Esta ley puede ser observada en un sistema cerrado cuando se cumple que:

$$U = Q + W \quad (2.1)$$

En donde  $U$  es la energía interna del sistema,  $Q$  es el calor que ingresa al sistema, y  $W$  el trabajo que ingresa al sistema. La ecuación anterior simplemente demuestra un estado del sistema, pero no demuestra como se comportan estas diferentes formas de energía. La relación se aprecia mejor por medio de la siguiente ecuación diferencial:

$$dU = dQ + dW \quad (2.2)$$

En la *ecuación 2.2* se observa que un cambio en la energía interna es solo causado por un cambio en el trabajo y en el calor que ingresa al sistema. Entonces, esta ecuación diferencial nos demuestra que los cambios solo suceden en la forma de la energía. Es en el cambio de los valores energéticos cuando se aprecia la magnitud de la primera ley de la termodinámica.

La energía interna del sistema ( $U$ ), se conoce como una función de estado, ya que está determinada por las variables de estado que ya se han mencionado: presión, volumen, temperatura y composición química. Las funciones de estado son independientes de los pasos que llevan al estado en particular [ 3 ]. Es decir, la energía interna tiene un valor que solo depende de las variables de estado y no de los estados anteriores por los que ha pasado el sistema.

Una forma útil de la primera ley de la termodinámica en aplicaciones mecánicas es la entalpía ( $H$ ). La entalpía es otra función de estado con la forma diferencial:

$$(2.3)$$

En donde  $dH$  es el cambio de entalpía,  $dU$  es el cambio en la energía interna y un cambio asociado al movimiento de las fronteras del sistema  $dV$  y a un cambio de presiones si existiese  $dP$ .

## 2.2 Segunda Ley de la Termodinámica

La segunda ley de la termodinámica analiza los procesos, es decir cambios en el estado del sistema. El recorrido del proceso es el grupo de estados que son precisos para alcanzar el estado final del proceso que se está analizando [ 7 ]. La primera ley limita el



estudio de los procesos a cambios de estado que pueden ser siempre reversibles, y sin pérdidas adicionales de energía. Es decir, es posible transformar trabajo en calor y viceversa sin pérdidas adicionales de energía. Sin embargo, este no es el caso. Si bien los procesos ideales pueden ser revertidos, los procesos reales son irreversibles. Los cambios de estado pueden ir en una dirección definida dependiendo del proceso [ 3 ].

La segunda ley de la termodinámica trata sobre la calidad de la energía y no solo de la cantidad. Para recordar y ejemplificar los efectos de la segunda ley se detalla a continuación el ciclo de Carnot. El ciclo de Carnot considera dos reservorios de calor, uno de alta temperatura  $T_H$  y otro de baja temperatura  $T_L$ . Los procesos pueden ser demostrados en un diagrama presión-volumen (P-V). El ciclo representado en la *Figura 2.1* tomado de la referencia [ 3 ].

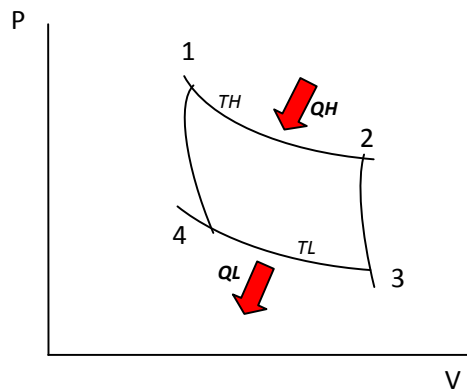


Figura 2.1 Esquema del ciclo de Carnot [ 3 ].

Los números en la figura representan los diferentes estados del proceso.

- 1-2: Expansión isotérmica a reservorio de alta temperatura  $T_H$ . Calor entra  $Q_H$ .
- 2-3: Expansión adiabática. El sistema genera trabajo que llamaremos  $W_1$ .
- 3-4: Compresión isotérmica a reservorio de baja temperatura  $T_L$ . Calor sale  $Q_L$ .
- 4-1: Compresión adiabática, existe un trabajo sobre el sistema  $W_2$ .

El ciclo de Carnot es idealmente reversible, todos sus procesos pueden ir de una dirección a otra. Sin embargo, la segunda ley de la termodinámica indica la necesidad de

tener dos reservorios por los cuales entra y sale calor claramente identificados en la *Figura 2.1* por medio de flechas rojas. Es decir, para cualquier máquina de calor, no existen atajos que evadan el ciclo de Carnot. El ciclo es el modelo a seguir, el límite ideal de lo que puede realizarse. Es un ciclo ideal únicamente por ser reversible al contrario de lo que ocurre con todo proceso en la realidad.

Para medir las consecuencias de la segunda ley de la termodinámica existe otra propiedad llamada entropía. La entropía  $dS$  definida en forma diferencial se aprecia en la *ecuación 2.4*:

$$dS = \left( \frac{dQ}{T} \right)_{\text{internamente reversible}} \quad (2.4)$$

Aquí  $dQ$  es el cambio de calor y  $T$  es la temperatura del sistema. La ecuación demuestra otra función de estado. Su valor es independiente de los procesos y depende únicamente de las variables de estado [ 3 ]. La *ecuación 2.4* indica que la relación es únicamente válida para procesos totalmente o internamente reversibles. Es decir, para aquellos procesos ideales como los encontrados en el ciclo de Carnot. La *ecuación 2.5* indica que la integral de curva cerrada es cero.

$$\oint dS = \oint \left( \frac{dQ}{T} \right) = 0_{\text{internamente reversible}} \quad (2.5)$$

¿Qué sucede con los procesos reales donde existen inevitablemente procesos irreversibles? El cambio de entropía en esos casos sigue la *desigualdad 2.6*

$$dS \geq \left( \frac{dQ}{T} \right) \quad (2.6)$$

Cuando existe igualdad en la relación 2.6, se tiene el caso propuesto de un ciclo ideal, un ciclo reversible. Mientras  $dS$  sea mayor que la parte derecha de la expresión, mayor será la irreversibilidad, lo que indica una menor eficiencia del proceso real.

La relación entre la primera y la segunda ley de la termodinámica con los procesos de la microestructura se detallará en las siguientes secciones. La idea general consiste en que el cambio de entropía deberá ser mayor a 0 si se desea un proceso espontáneo tanto de un proceso mecánico (como es una máquina de calor), como de un proceso químico (como lo es la difusión atómica). En el caso de la difusión atómica, la diferencia de entropía mayor que 0 indica un proceso que va desde una mayor a una menor concentración de un componente [ 7 ].

### 2.3 Las ecuaciones de Gibbs y la energía libre de Gibbs.

Las ecuaciones de Gibbs son ecuaciones en forma diferencial que expresan las funciones de estado únicamente utilizando las variables de estado y no otras funciones de estado [ 3 ]. Para obtenerlas es necesario trabajar entre las diferentes ecuaciones recordando la definición para la entropía dada en la *ecuación 2.4*. Para la energía interna la relación diferencial se muestra en la *ecuación 2.2*. El trabajo realizado por la energía interna de un sistema cerrado es siempre en contra de una presión hidrostática y sigue la relación de la *ecuación 2.7*.

$$dW = -PdV \quad (2.7)$$

Arreglando y reemplazando la *ecuación 2.4* y la *ecuación 2.7* en la *ecuación 2.2*, tenemos la *ecuación 2.8*.

$$dU = TdS - PdV \quad (2.8)$$

La entalpía también puede arreglarse de la misma manera reemplazando la *ecuación 2.8* en la *ecuación 2.3* obteniendo la *ecuación 2.9*.

$$dH = TdS + VdP \quad (2.9)$$

Existen dos funciones de estado adicionales que completan las ecuaciones de Gibbs, pero la de particular interés para el presente trabajo es la función de Gibbs o energía libre de Gibbs  $dG$  dada por la relación de la *ecuación 2.10*.

$$dG = -SdT + VdP$$

(2.10)

La relación proviene de la definición más clara de la función de Gibbs dada por la *ecuación 2.11*.

$$G = H - TS \quad (2.11)$$

Las relaciones mostradas suponen que la sustancia que será analizada es pura. Es decir, las variables de estado únicamente incluyen volumen, presión y temperatura. En el análisis microestructural se utiliza otra variable de estado que influye en las funciones de estado, la composición química  $N$  de la cual se habló en la primera sección de este capítulo. En una mezcla,  $N$  estará formada por elementos que indican la concentración de cada componente químico. Por lo tanto se define la función de Gibbs  $G$  para una mezcla como indica la *ecuación 2.12*.

$$G(P, T, N) = H(P, T, N) - T * S(P, T, N) = \min \quad (2.12)$$

La forma diferencial de la *ecuación 2.12* toma en cuenta también el cambio de la energía libre de Gibbs con respecto al cambio en uno de los componentes de la sustancia. Derivando parcialmente con respecto a las diferentes variables de estado tenemos las relaciones mostradas en las *ecuaciones 2.13*.

$$\begin{aligned} \frac{\partial G}{\partial P} &= V \\ \frac{\partial G}{\partial T} &= -S \\ \frac{\partial G}{\partial N_i} &= \mu_i \end{aligned} \quad (2.13)$$

Es necesario aclarar que para cada derivada parcial, las demás variables de estado que determinan la función de Gibbs permanecen constantes. La última ecuación es de particular interés. Con presión y temperatura constantes, y manteniendo la cantidad de todos los elementos constantes, excepto  $N_i$ , la última derivada parcial entrega  $\mu_i$  que se conoce como potencial químico del elemento  $i$  [ 3 ].

La *ecuación 2.14* expresa la derivada total de la función de Gibbs tomando en cuenta el potencial químico.

$$dG = VdP - SdT + \sum_i \mu_i * dN_i \quad (2.14)$$

Para condiciones en las que el sistema se encuentra a temperatura constante y presión constante, la energía libre de Gibbs queda únicamente definida por el elemento de sumatoria. Es decir se define como:

$$G = \sum N_i * \mu_i \quad (2.15)$$

La relación expuesta arriba puede ser también presentada por medio de fracciones molares simplemente cambiando el término  $N_i$  por  $X_i$ , donde  $X_i$  es una fracción molar del componente  $i$ . En tal caso solo es necesario tomar en cuenta que la suma de las fracciones molares no puede ser mayor que 1. Además, el potencial químico debe ser un potencial químico del componente  $i$  siempre relacionado a un potencial de referencia ( $\mu_i - \mu_{ref}$ ) [ 7 ].

Otra manera de cuantificar el efecto de los diferentes elementos que existen dentro de la microestructura o de cualquier sustancia es utilizando una variable de la probabilidad de los procesos internos ( $\xi$ ) que se pueden presentar, y su valor se encuentra entre 0 y 1. Un proceso interno puede ser el cambio de un componente a costa de otro componente, que se observaría en un cambio de  $\xi$ . Este cambio se define como positivo en la dirección de un proceso espontáneo. Esto se puede observar en la *ecuación 2.16* extraída de [ 7 ].

$$D = \frac{TdS}{d\xi} \quad (2.16)$$

Aquí,  $D$  se define como la fuerza de impulso para una reacción, generalmente reacciones que tienen que ver con cambios de fase. Se observa que para una reacción espontánea  $dS$  debe ser mayor que 0 por la segunda ley de la termodinámica,  $T$  es siempre mayor que 0, y  $d\xi$  debe ser mayor que 0. Es decir el proceso es espontáneo si el valor de  $D$  es mayor que 0. Como se observa  $D$  es análoga al potencial químico definido previamente y también tiene relación con la función de Gibbs por medio de una derivada

parcial mostrada en la *ecuación 2.17* manteniendo la temperatura y presión constantes [ 7 ].

$$D = -\frac{\partial G}{\partial \xi} \quad (2.17)$$

## 2.4 Equilibrio termodinámico y la función de Gibbs:

¿Por qué es tan necesaria la función de Gibbs? Porque es la única función que toma en cuenta los efectos combinados de entalpía y de entropía de un sistema. Sólo tomando los efectos combinados de esas otras funciones de estado es posible definir un estado de equilibrio del sistema. Este estado se encuentra cuando:

$$G(P, T, N) = H(P, T, N) - T * S(P, T, N) = \min \quad (2.18)$$

La minimización de la energía libre de Gibbs señala el estado donde cualquier cambio en las variables de estado P, T, N resultará inevitablemente en un aumento del valor de  $G$ . Un aumento de valor de  $G$  corresponde a un estado inestable donde sí podrían acontecer procesos espontáneos [ 7 ]. Como se puede observar,  $G$  combina la primera y la segunda ley para sistemas cerrados.

Otra manera de definir el equilibrio termodinámico de un sistema es por medio del potencial químico. Suponiendo que tenemos un componente químico que desea pasar de una región 1 a una región 2, el proceso será espontáneo en caso de minimizar la energía libre de Gibbs. Para minimizar la función únicamente igualamos la derivada parcial a 0. Se tiene finalmente la relación mostrada en la *ecuación 2.19*:

$$\frac{\partial G^1}{\partial N_i} - \frac{\partial G^2}{\partial N_i} = \mu_i^1 - \mu_i^2 = 0 \quad (2.19)$$

Es decir, si existe una diferencia del potencial químico esto lleva inevitablemente a un proceso espontáneo para alcanzar la igualdad.

Finalmente una forma de equilibrio de particular interés es aquella en la que participa la fuerza de impulso para un proceso interno  $D$ . Es de particular interés debido a que relaciona el cambio de un componente a costa de otro, es decir, involucra transformaciones, como por ejemplo transformaciones de fase. Como ya se ha establecido  $D$  debe ser mayor que cero para lograr un proceso espontáneo, entonces la condición para equilibrio será:

$$D \leq 0 \quad (2.20)$$

## **2.5 Termodinámica aplicada a la microestructura:**

En el estudio de microestructuras se toma en cuenta todas las definiciones anteriores. Sin embargo, el comportamiento de una microestructura no es un comportamiento de una mezcla, más bien se comporta como una solución sólida. Esto quiere decir que deben tomarse en cuenta propiedades características de las soluciones. En soluciones es necesario considerar la termodinámica de procesos microscópicos y llevarlos a un orden macroscópico [ 7 ].

Una manera de realizar lo antes mencionado es por medio del estudio de estadística termodinámica. Podemos considerar una solución como un gran número de átomos de dos sustancias. Cada átomo es un elemento de una sustancia A o una sustancia B. Al tener un gran número de átomos de diferentes características ya no solo debe tomarse en cuenta los cambios de la sustancia pura, también debe tomarse en cuenta los posibles cambios que sucedan cuando las sustancias están combinadas. Esto es aplicable a una microestructura, una solución de diferentes átomos de diferentes sustancias [ 7 ].

La estadística termodinámica invita a entender probabilidades relacionadas con entropía. Se establece la suposición que los átomos de la sustancia A son esferas negras y los de la sustancia B son esferas blancas y que cada grupo está inicialmente en su región. Existe igual número de esferas A y B, podría pensarse que existen 50 esferas negras y 50 esferas blancas. En un paso siguiente se elige al azar una esfera de todo el grupo. Luego

se elige otra esfera de todo el grupo. Finalmente se intercambian las esferas entre sí. Debido a que existen dos grupos, la probabilidad de que una esfera blanca encuentre su espacio en la zona de esferas negras y viceversa es de 0.5. Los grupos con tan alto número de esferas permanecen relativamente iguales en una primera etapa. Si se repite el proceso, la probabilidad de un cambio será aproximadamente igual a 0.5 [ 7 ].

Se desea ahora extraer las mismas esferas intercambiadas en el primer paso y regresarlas a su posición original. Esto resulta mucho más complicado debido a que sería necesario escoger la esfera negra entre 49 blancas seguido por una esfera blanca entre 49 negras del total de 100 esferas. La probabilidad de este proceso es de alrededor de  $10^{-4}$ . Es decir, el primer paso es un proceso espontáneo porque es muy probable que suceda. El segundo no lo es. En conclusión, el intercambio de átomos es un proceso irreversible. La *Figura 2.2* esquematiza este ejemplo.

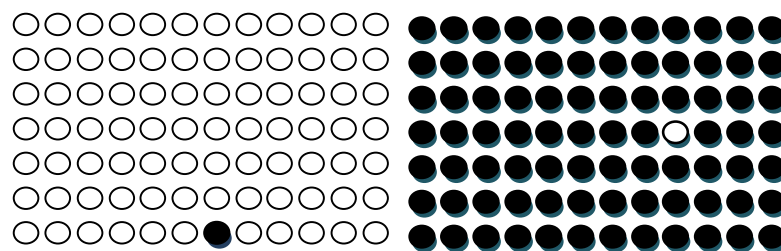


Figura 2.2 Grupo de esferas blancas y negras luego de haberse producido un primer intercambio. Atrapar aquellas pelotas intercambiadas una vez realizado el proceso es extremadamente improbable.

Existen muchas posiciones disponibles en los grupos negro y blanco. Por esta razón, existe un gran número de formas en las que se puede intercambiar la pelota blanca por la negra. Debido a que todas las posiciones siguen disponibles en un segundo paso, el número de formas para arreglar los intercambios aumenta. El número total de posibilidades de reordenar el grupo se vuelve cada vez más grande y alcanza el valor  $W$  dado por la *ecuación 2.21*.

$$W = \frac{N!}{N_A! - N_B!} \quad (2.21)$$



En donde  $N_A!$  es el número de átomos de A,  $N_B!$  es el número de átomos de B y  $N!$  es la suma de los átomos de B y A. A medida que los procesos avanzan,  $W$  disminuye, es decir las posibilidades para reordenar los grupos disminuyen. De este razonamiento Boltzman definió una entropía para soluciones que puede explicar el comportamiento de los átomos en las sustancias mezcladas. Definió la entropía  $S$  como una función logarítmica.

$$S = k_B * \ln W \quad (2.22)$$

Aquí,  $k_B$  es un factor de proporcionalidad conocida como la constante de Boltzmann igual a  $5.67E-8 \text{ W/m}^2\text{K}^4$ . Logró de esta forma contabilizar el desorden creado por los sistemas microscópicos en los sistemas macroscópicos.

Para varios componentes  $\ln W$  puede expresarse por una aproximación matemática y finalmente puede la entropía puede expresarse en términos de fracciones molares como:

$$S_i = k_B * N(X_i \ln X_i) \quad (2.23)$$

Donde  $N$  es el número de Avogadro,  $S_i$  indica la entropía particular para cada componente en la solución y  $X_i$  es la fracción molar del componente  $i$ .

Para finalizar, esta entropía debe incluirse en el cálculo de la energía libre para soluciones, lo que permite encontrar los procesos espontáneos que ocurren en la interacción de varias sustancias. La energía libre resultante es una energía libre que toma en cuenta la energía particular de la sustancia pura y agrega la entropía de Boltzman [ 7 ].

$$GmS = \sum (X_i * Gmi + RT \ln X_i) \quad (2.24)$$

Aquí,  $GmS$  es la energía de Gibbs de la solución, y  $Gmi$  es la energía de Gibbs para cada fracción de moles. En la ecuación 2.24  $R$  es la constante de Boltzman multiplicada por el número de Avogadro para utilizar la entropía de Boltzman con fracciones molares en el cálculo de la función de Gibbs.

Esta última función es la que se intenta alcanzar en las simulaciones más simples de microestructuras, es decir, tomar en cuenta la energía libre causada por una comparación entre dos sustancias puras y la causada por el gran número de átomos de las sustancias.

# 3

## MÉTODO DE MONTE CARLO POTTS

El método de Monte Carlo Potts es un método fácil de comprender pero con gran nivel de detalle para que la simulación resultante logre acercarse a la realidad. Para ello se necesita entender tanto las ecuaciones que considera el método como el algoritmo a utilizarse en la programación del mismo. El método de Monte Carlo Potts se basa en la probabilidad y en la resolución mediante números aleatorios. En este capítulo se detallan los elementos más importantes del método que incluyen:

- La aplicación de probabilidades para generar la evolución microestructural.
- El *Hamiltoniano*, ecuación que presenta un método para simular la energía libre de Gibbs.
- Las geometrías que pueden ser utilizadas como dominio para lograr la simulación.
- Métodos para definir las energías en las fronteras del material.
- El algoritmo del método y la función de Metrópolis.
- Los métodos para variar parámetros como movilidad del borde de grano y componentes dentro de la matriz.

### 3.1 Probabilidad y el método de Monte Carlo

El método de Monte Carlo está basado en la resolución de problemas utilizando números aleatorios y probabilidad. El método tiene este nombre en relación a los populares juegos de azar en los casinos de Monte Carlo, y es utilizado en numerosas aplicaciones [ 6 ]. Sin embargo, el método es computacional debido a que para obtener resultados es necesario realizar un gran número de repeticiones.

El método de Monte Carlo compara la probabilidad de un sistema utilizando un gran número de intentos aleatorios. A continuación en la *Figura 3.1* se presenta un

ejemplo simple que demuestra el éxito del método. Supongamos que queremos obtener el valor para el número pi. Para ello definimos un área de cuarto de círculo encerrada en un cuadrado. El cuadrado tendrá entonces lado = radio del círculo.

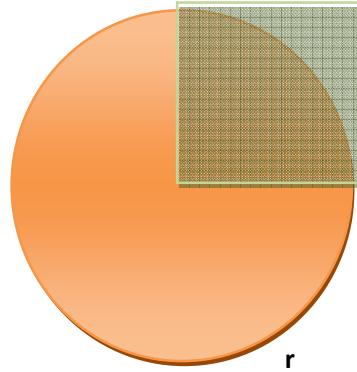


Figura 3.1 Demostración del éxito del método Monte Carlo

La probabilidad de encontrar el área del círculo dentro del área del cuadrado es simplemente:

$$P = \frac{\frac{1}{4}\pi r^2}{r^2} = \frac{1}{4}\pi \quad (3.1)$$

Se genera una ubicación aleatoria dentro del cuadrado. Esta ubicación está descrita por sus coordenadas en  $x$ ,  $y$ . Utilizando el teorema de Pitágoras tenemos que  $r^2=x^2+y^2$  [ 15 ]. Si esta relación es verdadera para una ubicación generada aleatoriamente, esta ubicación estará dentro del círculo. De esta forma la probabilidad está definida como:

$$P = \frac{\#valores\ generados\ aleatoriamente\ dentro\ del\ círculo}{\#valores\ generados\ totales} \quad (3.2)$$

Igualando las probabilidades se puede encontrar el valor de pi. Sin embargo el método exige un gran número de valores generados para mejorar la aproximación.

Una vez entendida la probabilidad que envuelve el método de Monte Carlo es importante hablar del Hamiltoniano; el cual es un tipo de matriz que indica la relación existente entre la posición seleccionada y sus vecinos.

### 3.2 El Hamiltoniano

El Hamiltoniano es una matriz la cual describe como funciona la interacción entre un punto específico elegido y sus vecinos, que representan la energía de la frontera de grano. A continuación la *ecuación 3.3* muestra lo anteriormente mencionado sobre el Hamiltoniano.

$$H = -J \sum_{nn} (\delta_{Q_i Q_j} - 1) \quad (3.3)$$

En la *ecuación 3.3*, H representa al Hamiltoniano (la cual no debe confundirse con la entalpía),  $Q_i$  representa una de las posibles orientaciones en el elemento i-ésimo de la matriz establecida. El símbolo  $\delta$  es la variable delta de Kronecker, la cual es 1 cuando dos elementos son iguales, y 0 cuando son distintos. De esta manera vecinos que tienen diferentes orientaciones contribuyen una cantidad  $J$  al sistema de energía y los de orientaciones iguales contribuyen con una cantidad de 0. La constante  $J$  es una constante

positiva que especifica la escala de interacción entre la orientación del elemento  $i$ -ésimo con sus vecinos. La suma se la realiza en relación a todos sus posibles vecinos [ 1, 2 ].

En el caso de una matriz de dos dimensiones con red cristalográfica cuadrada el número de vecinos para un elemento cualquiera va a ser de 8. En el caso de una matriz de tres dimensiones con red cristalográfica cuadrada, el número de vecinos para cualquier elemento es de 26.

En la *Figura 3.2* se indica claramente como se realizan las comparaciones entre un elemento seleccionado dentro de la matriz con red cristalográfica cuadrada con sus vecinos.

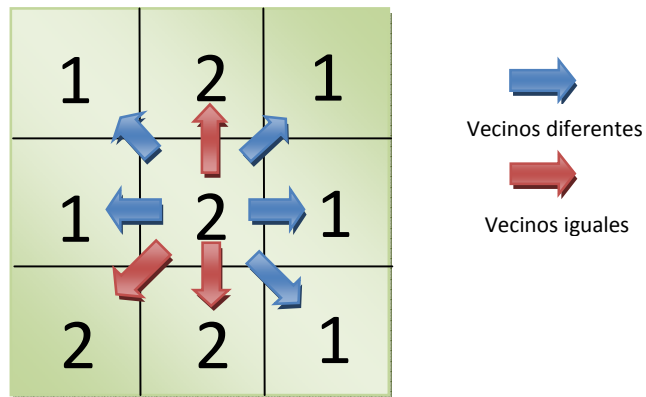


Figura 3.2 Cálculo de energía libre de un elemento con sus vecinos en matriz cuadrada 2D,  $Q_i = 2$  [ 7 ].

Como se puede ver en la *Figura 3.2*, el elemento seleccionado es el  $Q_i = 2$  y éste es comparado con sus ocho vecinos de los cuales tres son iguales a sí mismo y los cinco restantes representados por el número 1 son diferentes. Este ejemplo muestra dos orientaciones cristalográficas y se las puede interpretar como la existencia de dos granos. Es decir, dentro de esta matriz de la *Figura 3.2* la unión de todas las orientaciones cristalográficas de número 1 formarán un grano, y todas las orientaciones cristalográficas de número 2 formarán un segundo grano.

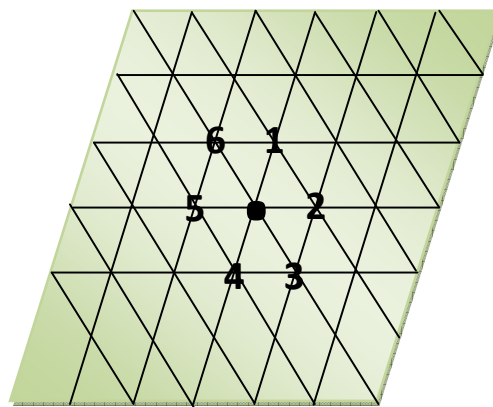
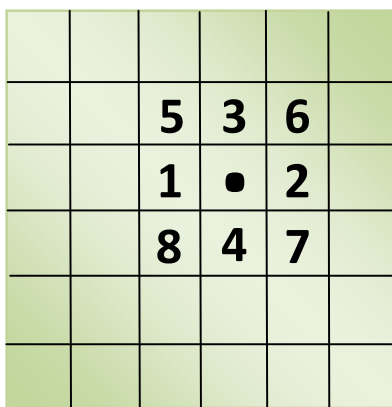
Para el cálculo de la energía libre se debe determinar cuantos vecinos diferentes tiene el elemento seleccionado. Para este ejemplo en particular, la energía libre es de  $G_i$

= 5. Es importante notar que esta comparación con los vecinos debe realizarse dos veces dentro de cada paso de Monte Carlo (MCS), pero con diferentes elementos seleccionados aleatoriamente. Es decir, que al finalizar la primera comparación se debe elegir un nuevo elemento en la matriz aleatoriamente y volver a comparar. Esto se realiza debido que al tener dos energías se puede comparar y “comprobar” la probabilidad de elegir esa posición con esa orientación cristalográfica.

### 3.3 Tipos de redes cristalográficas que se pueden utilizar

Los tipos de redes cristalográficas que se pueden utilizar son dos para el caso de una simulación en dos dimensiones. En cada tipo de red cristalográfica un elemento individual puede tener diferente número de vecinos cercanos. El primer tipo es la red cristalográfica cuadrada la cual para un elemento en la matriz tiene 8 vecinos cercanos como se muestra en la *Figura 3.3*. Como se puede observar en esta figura, el punto representa el elemento seleccionado aleatoriamente por el programa de simulación. Además los números del 1 al 8 indican el número de vecinos cercanos que este tipo de red cristalográfica puede tener. Adicionalmente los números muestran el orden en que va comparando el elemento seleccionado con sus vecinos verificando si tienen su misma orientación cristalográfica o no durante la simulación.

Por otro lado, se tiene un segundo tipo de red cristalográfica que es la triangular o que es también conocida como hexagonal debido a que la distribución de sus vecinos forma un hexágono. Este tipo de red cristalográfica, para cierto elemento, tiene 6 vecinos cercanos como se muestra en la *Figura 3.4*. Se puede observar en esta figura que se ha puesto los números del 1 al 6 que indican la distribución de los vecinos posibles en este tipo de red cristalográfica.



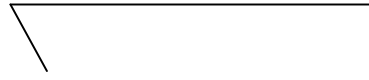


Figura 3.3 Red cristalográfica cuadrada de 2D [ 7 ]

Figura 3.4 Red cristalográfica triangular (hexagonal) de 2D [ 7 ]

Además de una simulación 2D se puede hablar de una simulación en 3D. A este tipo de red cristalográfica se la conoce como red cristalográfica simple cúbica. Un elemento elegido dentro del cubo en este tipo de red cristalográfica tiene un total de 26 vecinos cercanos posibles. Esto se observa en la *Figura 3.5* en donde se puede ver claramente como es la distribución de los vecinos del cubo en el centro de color rojo. Para esta simulación en 3D, la forma más efectiva es con una red cristalográfica cúbica.

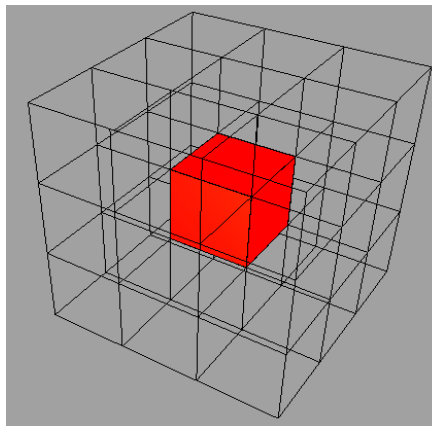


Figura 3.5 Red cristalográfica cuadrada en 3D con 26 vecinos cercanos para el elemento en rojo

Para esta investigación se ha decidido utilizar la red cristalográfica cuadrada de dos dimensiones por las siguientes razones: 1) En MATLAB 7.1<sup>®</sup> no es tan sencillo organizar una matriz triangular, además idealmente el tipo de red cristalográfica no debería afectar físicamente a la simulación y adicionalmente no influye en la termodinámica ni en la dinámica de la simulación, 2) MATLAB es muy útil para sobreponer imágenes sobre una matriz y mostrar los granos como se observaría en la realidad, 3) Para una simulación en 3D se requiere de otros paquetes como JAVA 3D<sup>®</sup> u OPENGL<sup>®</sup> para realizar las visualizaciones de grano respectivas [ 7 ].



### 3.4 Condiciones de frontera

Es importante explicar que no todos los elementos dentro de una matriz van a tener el mismo número de vecinos cercanos. Este es el caso de elementos en la matriz que se encuentran tanto en los lados de la matriz como se muestra en la *Figura 3.6* a continuación.

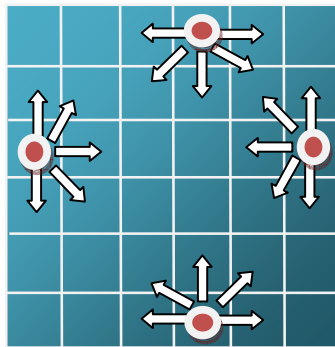


Figura 3.6 Elementos elegidos en los lados de la matriz cuadrada solo contienen 5 vecinos cercanos

Como se puede ver en la figura anterior, si se realiza las comparaciones de un elemento seleccionado, en el lateral de una matriz con sus vecinos cercanos, estos tendrán solo 5 vecinos cercanos para compararse. Por otro lado, los elementos de la matriz que se encuentran en las esquinas de la matriz cuadrada solo pueden compararse con 3 vecinos como se aprecia en la *Figura 3.7* a continuación.

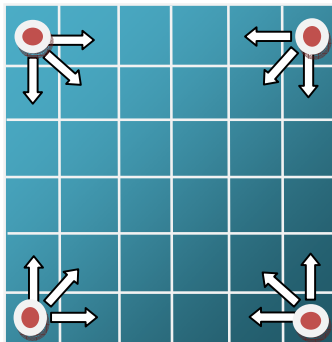


Figura 3.7 Elemento en esquinas de la matriz cuadrada solo contienen 3 vecinos cercanos

De esta manera solo los elementos en el centro tendrían 8 vecinos cercanos. Lo correcto es que todos los elementos de la matriz sin excepción tengan 8 vecinos con los cuales comparar (en el caso de una red cristalográfica cuadrada). Se debe hacer esto ya que, de lo contrario, el crecimiento de grano tendría preferencias y los granos en esos espacios serían diferentes a lo esperado en comparación a los granos internos.

De la misma manera, como se tiene un solo material sería extraño que en las esquinas o en los lados la microestructura tenga un comportamiento diferente. No se puede asumir que las esquinas o los lados son diferentes en un mismo material, por ello, se necesita que todos se comparen con 8 vecinos cercanos. Además, en el caso de que el límite sea otra substancia con energías diferentes debería tratarse de diferente manera. De esta forma es que se toman en cuenta las condiciones de frontera para esta simulación de la evolución microestructural en dos dimensiones.

Estos elementos especiales mencionados anteriormente (los elementos en las esquinas y en los lados) necesitan un trato especial. Es decir, para mantener control sobre la simulación se puede tratar las fronteras con condiciones de borde. Por otro lado se debe hacer que todos los elementos dentro de la matriz tengan 8 comparaciones, para ello existen diferentes condiciones de frontera que se pueden implementar en la simulación. A continuación se muestran los cuatro casos de condiciones de frontera son útiles al rato de realizar el programa de la simulación.

### **3.4.1 Condición de frontera de superficie**

Esta condición tiene que ver con la energía que existe entre superficies como por ejemplo una superficie sólida con una líquida ó una sólida con una gaseosa. Por ejemplo, si en este tipo de límite se tiene agua o aire, el grano puede cambiar de forma.

En la *Figura 3.8* se muestra este tipo de condición de frontera de superficie. En este ejemplo lo que se nota es la interacción entre la superficie del sólido con el agua y con energías diferentes. En el presente trabajo no se requiere de esta condición de frontera ya que lo que se trata de simular no es la frontera de superficie del material sino el interior de una probeta de acero. Es decir, se considera un material muy grande y entonces la superficie no tiene importancia.

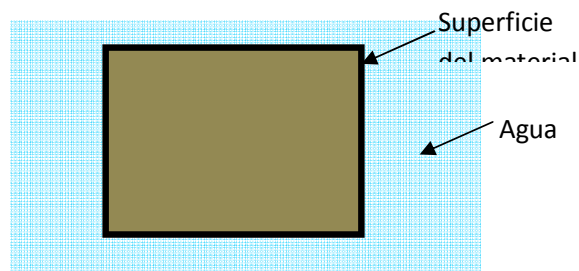


Figura 3.8 Condición de frontera de superficie

### 3.4.2 Condición de frontera tipo espejo

En este tipo de condición de frontera se reflejan los vecinos existentes para que puedan existir 8 sitios con los cuales comparar. Se asume que el material para esta simulación va a ser homogéneo de manera que al hacer este reflejo de los sitios no afectara en el crecimiento de grano. Esto se explica claramente en la *Figura 3.9* a continuación.

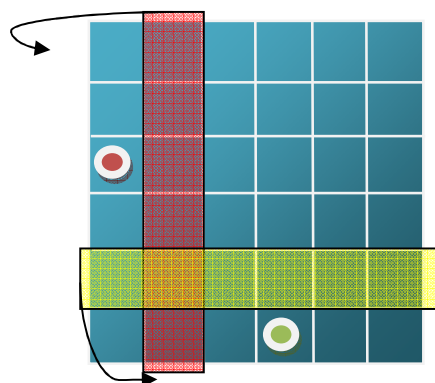


Figura 3.9 Condición de frontera tipo espejo

En la *Figura 3.9* se indica como funciona este tipo de condición de frontera. Todos los elementos de la matriz que se encuentren dentro se trasladaran o reflejarán hacia el lado opuesto de la frontera con sus mismos valores. Es decir, todos los elementos seleccionados con rojo en la figura anterior se moverán hacia la izquierda en este ejemplo en particular. De esta manera el elemento que se muestra en la *Figura 3.9* como punto rojo tendrá 8 vecinos para compararse. Ocurre exactamente lo mismo si el elemento estuviera en un lado inferior como se puede apreciar en la figura mencionada. Este tipo de condición es muy útil para lo que se intenta hacer en esta simulación pero existe una condición de frontera más popular utilizada en casi todos los casos y es la condición de frontera periódica.

### 3.4.3 Condición de frontera periódica

En este tipo de condición de frontera lo que se crea es una simulación muy parecida a la anterior. La diferencia radica en que en vez de repetirse solo un segmento de la matriz se repite la matriz completa a continuación de cada frontera. Para explicar mejor, se puede comprender como si se enrollara a la matriz en forma de toroide. Se puede asumir para este caso que el material es homogéneo y que la matriz va a repetirse a lo largo del material. En la *Figura 3.10* se puede entender mejor como funciona la condición de frontera periódica.

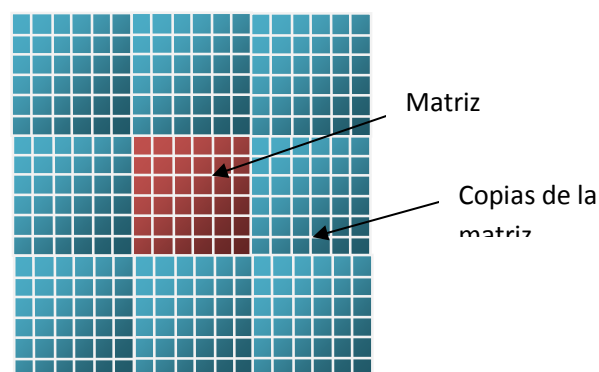


Figura 3.10 Condición de frontera periódica.

En la *Figura 3.10* se observan 9 matrices, la del medio que está en rojo es la matriz principal en la cual se esta realizando la simulación. Por otro lado las 8 restantes son copias de la matriz principal que se trasladan hacia las fronteras de la matriz principal. De esta forma los elementos en las fronteras de la matriz serán comparados con 8 vecinos cercanos y se mantiene una especie de homogeneidad en el material elegido.

### 3.4.5 Condición de frontera periódica torcida

Esta condición es una variante de la condición de frontera periódica en la cual se realiza lo mismo, pero, la diferencia es que en la frontera un vector de desplazamiento paralelo a la frontera es impuesto. Este tipo de condición de frontera se la usa cuando se trata de fronteras planas que no tienen intersección perpendicular con las fronteras del área de simulación. Esto se puede interpretar con el siguiente gráfico de la *Figura 3.11*. En esta figura se puede apreciar como existe periodicidad al añadir otra matriz al lado izquierdo pero manteniendo una línea recta continua a lo largo del material, de esta manera se intenta mantener la homogeneidad necesaria para el crecimiento del grano.

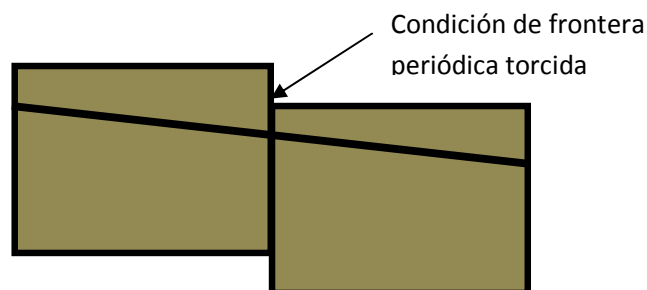


Figura 3.11 Condición de frontera periódica torcida [ 7 ].

Después de haber revisado y analizado estos cuatro tipos de condiciones de frontera, se llegó a la conclusión de que la condición de frontera periódica es la más útil para la simulación de la evolución microestructural. La razón es debido a que la condición de frontera periódica simula la homogeneidad del material. Es decir, puede considerarse a la matriz como un elemento del material que se repite continuamente.

### 3.5 Algoritmo del modelo y las funciones de probabilidad utilizadas

El algoritmo de un programa es un método que se realiza para poder resolver un problema y es esencial para la programación que en este caso sirve para realizar la simulación de la evolución microestructural [ 12 ]. Se puede entender al algoritmo como el ducto por el cual pasa el programa de la mente del ser humano hacia la computadora. El algoritmo es donde entra el pensamiento sistemático, es decir, la lógica necesaria para realizar el programa posterior que tenga características de versatilidad y utilidad práctica.

En la sección 3.4 se completaron las condiciones necesarias para realizar la comparación entre el elemento seleccionado y sus vecinos cercanos. También se ha definido la matriz necesaria para simular la red cristalográfica en la sección 3.3. A continuación se presenta como utilizar estas condiciones para lograr simular la evolución microestructural en base del método de Monte Carlo Potts.

La distribución de números en la matriz de la red cristalográfica representa un estado del sistema. El método de Potts consiste en utilizar el método de Monte Carlo para seleccionar un lugar en la matriz y cambiar el estado del sistema dependiendo de una probabilidad definida. Para entender mejor el método se ha elaborado los pasos del algoritmo necesarios para el funcionamiento del programa.

**1)** Se genera una matriz con disposición cuadrada. Cada posición de la matriz representa una ubicación en el material. A cada posición se asigna un valor representando su orientación cristalográfica. Esto se detalla en la *Figura 3.12*.

1	4	3	1	1	3
1	2	2	1	4	1
4	1	1	1	4	1
4	4	3	1	3	1
1	3	1	4	2	1
1	3	3	1	2	2

Figura 3.12 Matriz con distribución de 4 orientaciones cristalográficas.

Un mismo número representa la misma orientación cristalográfica.

Como se puede apreciar en la *Figura 3.13*, un grano está definido por un grupo de ubicaciones con el mismo valor de dirección cristalográfica. Los bordes de grano están definidos entre los granos con diferentes direcciones.

1	4	3	1	1	3
1	2	2	1	4	1
4	1	1	1	4	1
4	4	3	1	3	1
1	3	1	4	2	1
1	3	3	1	2	2

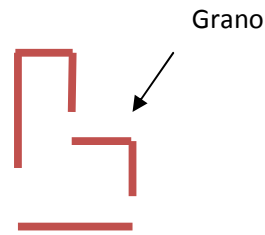


Figura 3.13 Forma de un grano con orientación cristalográfica 2

2) Se elige aleatoriamente una posición  $x, y$  en la matriz como se muestra en la *Figura 3.14*.

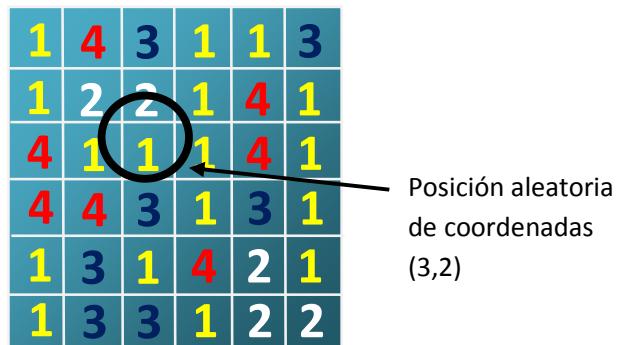


Figura 3.14 Posición aleatoria en la columna 3 y en la fila 2

3) En la posición seleccionada se encuentra el número de ubicaciones vecinas con diferente dirección cristalográfica. Se llamará  $EI(1)$  a dicho número. Se explicó en la sección 3.2 acerca de  $EI(1)$  al que llamamos Hamiltoniano. Ya se ha mencionado que el Hamiltoniano toma en cuenta los vecinos cercanos diferentes. En el caso de la *Figura 3.14*, la energía  $EI(1)$  sería de 7.



4) Se vuelve a elegir aleatoriamente otra dirección cristalográfica con las mismas coordenadas. Se reemplaza la nueva orientación cristalográfica con la ya existente y se compara nuevamente. Se almacena el número de ubicaciones vecinas con diferente dirección cristalográfica en una variable  $EI(2)$ .

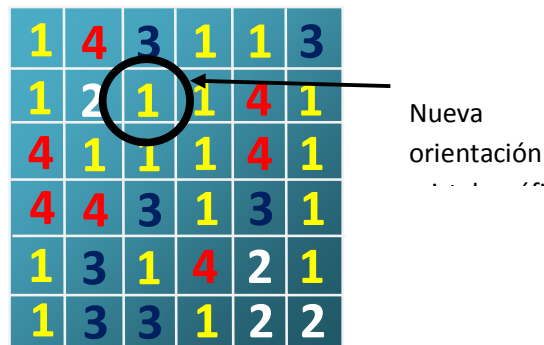


Figura 3.15 Selección y reemplazo aleatorio de nueva orientación cristalográfica

Como se muestra en la *Figura 3.15* aleatoriamente se ha obtenido la orientación 1 que se ha reemplazado por la orientación 2. Para este caso en particular  $EI(2)$  es 3.

5)  $EI(1)$  y  $EI(2)$  son valores que representan la energía libre de la ubicación en la matriz con las diferentes direcciones cristalográficas. Se comparan los valores de  $EI(1)$  y  $EI(2)$  y se escoge el de menor valor. Aquella dirección cristalográfica con la menor energía libre quedará grabada en la ubicación que está siendo evaluada. De la *Figura 3.15* y *3.14* se observa que  $EI(2) < EI(1)$  por ende se aceptará como disposición final la matriz mostrada en la *Figura 3.15*.

**6)** La matriz generada tiene N posiciones x N posiciones, se llama un paso de Monte Carlo (MCS) cuando se ha evaluado el número de veces equivalentes al número de ubicaciones en la matriz. Por ejemplo: en una matriz cuadrada NxN se deberán completar NxN evaluaciones aleatorias para completar un paso de Monte Carlo. En el caso de la matriz analizada en la *Figura 3.12*, en promedio se realizarán en 36 comparaciones entre *El (1)* y *El (2)*.

**7)** Una vez obtenido los valores de *El (1)* y *El (2)* se necesita hacer una comparación entre ambos sacando una diferencia de manera que dicha diferencia de energía sea útil para utilizarla en la probabilidad de Metrópolis. Dicha función de probabilidad incluye ciertas propiedades como la temperatura a la que se encuentra el material, y la constante de Boltzman [ 7 ].

$$P = \left\{ \begin{array}{ll} \exp \frac{-\Delta E}{kT_s}, & \text{si } \Delta E > 0 \\ 1, & \text{si } \Delta E \leq 0 \end{array} \right\} \quad (3.4)$$

En la ecuación 3.4 se puede apreciar la denominada probabilidad de Metrópolis. En el caso que la diferencia de energías resulte ser mayor que cero se deberá calcular el exponencial para aceptar la nueva orientación cristalográfica. Caso contrario la probabilidad de Metrópolis tendrá un valor de 1 como sucedió en el ejemplo de las *Figuras 3.14* y *3.15*.

Para comprender mejor el algoritmo se ha diseñado un flujograma en la *Figura 3.16* que enmarca todos los pasos explicados anteriormente. En el flujograma se utiliza la variable deltaE para comparar las energías libres en las fronteras *El (1)* y *El (2)*. De esta forma si deltaE es menor que cero *El (2)* es menor que *El (1)*. En el flujograma, MCS (Monte Carlo Step) es el número de pasos de Monte Carlo que se van a ejecutar. Por otro lado, N es el número de columnas o filas que tiene la matriz, y ya que la matriz es cuadrada,

tendrá un tamaño de  $N \times N$ .  $Q$  es la cantidad de orientaciones cristalográficas posibles dentro de la simulación.

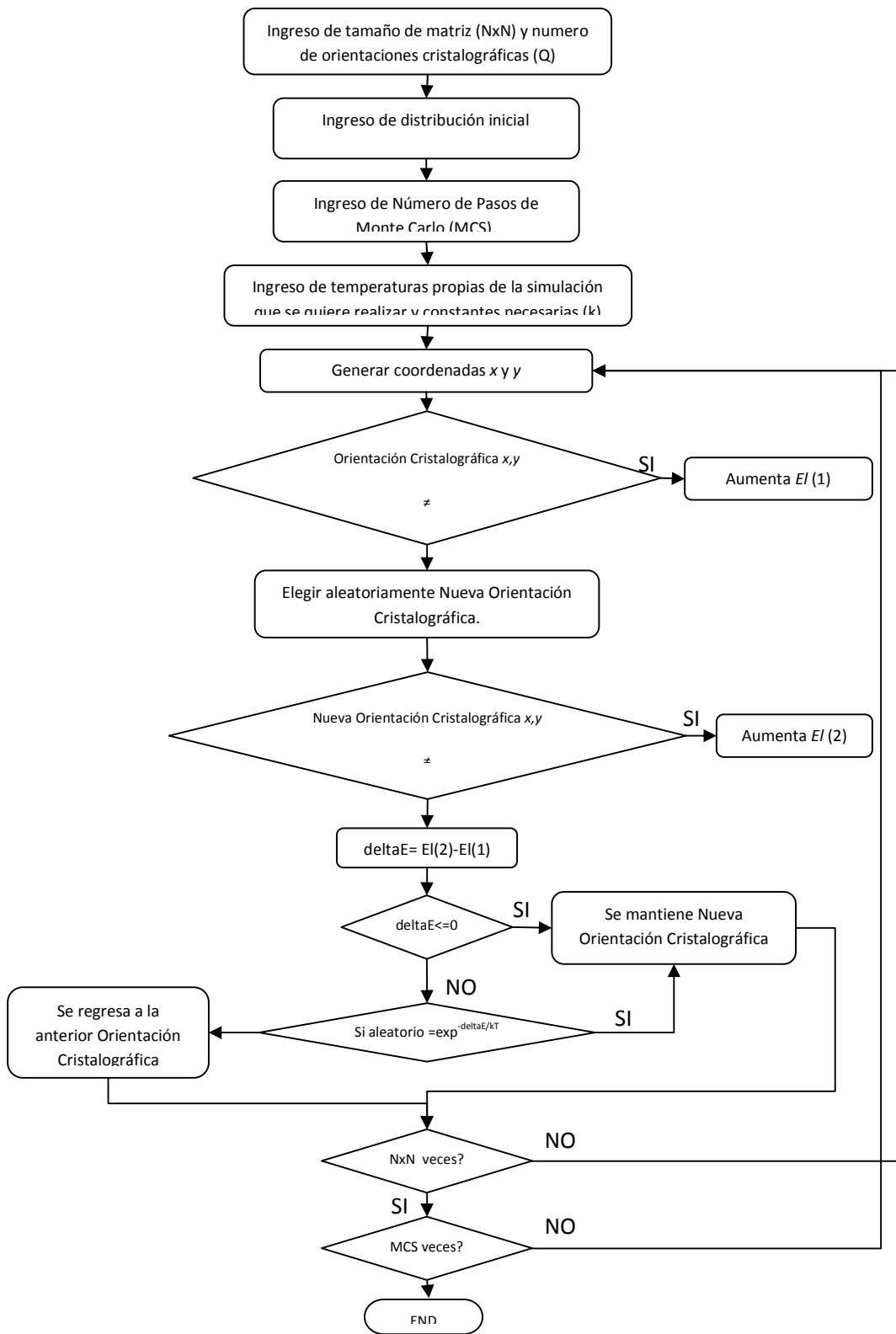


Figura 3.16 Flujograma del método de Monte Carlo Potts

### 3.6 Variando la movilidad del borde de grano.

Hasta ahora se podría pensar que para simular el crecimiento de grano a una temperatura dada basta únicamente cambiar la variable  $T$  en la función de Metrópolis explicada en la *ecuación 3.4*. Sin embargo, hasta el momento, no se han introducido variables reales dentro de la simulación. Es decir, podríamos considerar a la simulación realizada como una organización de pixeles en una matriz cuadrada que poco tiene que ver con la realidad. De esta forma la expresión  $kT$  en la probabilidad de Metrópolis de la *ecuación 3.4* agrega a la simulación un efecto térmico sin relación alguna con temperaturas reales pero que obliga a la matriz a comportarse como una solución sólida. [1] La variación de la expresión  $k*T$  crea pequeñas irregularidades en los bordes de grano que logran eliminar efectos no deseados por la forma que tiene la matriz, en el caso presentado en este trabajo, una forma cuadrada. Esta geometría no es una geometría que se encuentre en los fenómenos de evolución de microestructuras. Variar  $k*T$  permite eliminar los efectos de la geometría utilizada. [7]

¿Cómo lograr simular efectos térmicos reales? Uno de los métodos es alterar la movilidad del borde de grano. La movilidad del borde de grano indica simplemente qué tan rápido evoluciona un borde de grano en el tiempo. El borde de grano avanza conforme el grano aumenta de tamaño.

Manejar la movilidad dentro del algoritmo del generador de microestructuras es bastante simple. La movilidad depende de la probabilidad que se utilice para manejar el cambio entre orientaciones cristalográficas demostrado en la *Figura 3.2*. Es decir, alterando la función de Metrópolis utilizada podemos manejar la velocidad de avance del borde de grano. Esto se logra multiplicando a la función por un factor como se demuestra en la *ecuación 3.5*.

$$P = \left\{ \begin{array}{ll} u * \exp \frac{-\Delta E}{kT}, & \text{si } \Delta E > 0 \\ u * 1, & \text{si } \Delta E \leq 0 \end{array} \right\} \quad (3.5)$$

El factor  $u$  de la ecuación 3.5 se utilizará con el mismo nombre en la elaboración de los programas. Este factor  $u$  es menor que 1 para que la función de probabilidad sea válida. Multiplicar la probabilidad por  $u$  logra reducir la probabilidad de que exista cualquier cambio en una comparación. Por ejemplo, si  $u$  es 0.5, la probabilidad para aceptar cualquier cambio se verá reducida a la mitad. Es decir, los bordes no se verán afectados con la misma constancia que con la probabilidad original de Metrópolis. La Figura 3.17 muestra la segunda comparación entre la posición central y los alrededores de un elemento con orientación cristalográfica 1. En el caso original esta nueva posición se aceptaría definitivamente y la matriz cambiaría. En el caso modificado, esto no sucede siempre, solo cambia si se vence la probabilidad  $u*1$  como indica la ecuación 3.5.

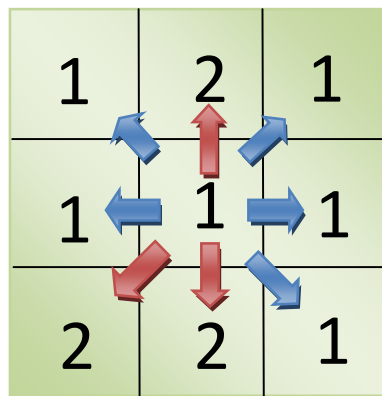


Figura 3.17 Energía libre de un elemento con sus vecinos. Si  $u < 1$  es posible que 1 no se escoja y la matriz no cambie [ 7 ].

El cambio del factor  $u$  permite fácilmente modificar la simulación y ajustarla a efectos reales tomando en cuenta lo siguiente:

- 1) La movilidad más alta del sistema sucede con  $u=1$ .

- 2)  $u$  puede ser función de alguna propiedad de la matriz. En un caso práctico podría ser función de la temperatura: el factor se reduce a menores temperaturas lo que simula menor movimiento de bordes debido a menores energías térmicas.
- 3)  $u$  también puede modificarse para relacionar la movilidad con el tipo de componente de la posición seleccionada. La forma de agregar componentes a la simulación se detalla en la siguiente sección.

### **3.7 Agregando componentes a la simulación.**

Se ha observado como modificar la movilidad de la frontera entre granos mediante la modificación de la probabilidad de aceptación de cambio en la ecuación 3.4. Existen otras modificaciones que se pueden realizar al algoritmo para obtener resultados interesantes.

Algo que es deseable simular es el efecto de diferentes componentes que pueden hallarse dentro de la microestructura. Por ejemplo, puede ser interesante observar el comportamiento de dos fases de un mismo material. ¿Cómo lograr este efecto? Se puede utilizar una segunda matriz que especifique los componentes deseados. De esta forma tendríamos dos matrices, una con orientaciones cristalográficas y otra con la identificación del componente. Así, una misma coordenada  $x,y$  identificará tanto la orientación cristalográfica de un grano, como su componente. Esto se detalla en la *Figura 3.18*.

1	1	4	1	3	1	1	1	3	2	
1	1	2	1	2	1	1	1	4	2	1
4	1	1	1	1	1	1	1	4	2	1
4	2	4	2	3	2	1	1	3	1	1
1	1	3	2	1	1	4	2	2	2	1
1	1	3	2	3	2	1	1	2	2	2

Figura 3.18 Cada posición tiene 2 identificadores: uno para orientación cristalográfica (esquinas superiores) otro para componente (esquinas inferiores).

Para agregar componentes no basta con localizarlos en la matriz. Es necesario asociar una energía propia para cada uno de los componentes. Esto se logra por medio de la variación del Hamiltoniano de la ecuación 3.3. Obsérvese que en la ecuación 3.3 existe una  $J$ . Hasta ahora se ha considerado  $J$  como 1 para aumentar así la energía libre dependiendo de los vecinos de un elemento de la matriz. Para colocar energías propias de los componentes,  $J$  puede variar dependiendo del componente que esté siendo comparado. En resumen, existen dos tipos de energía, aquella asociada a un cambio de orientación cristalográfica y aquella asociada a un cambio entre componentes. Una vez sumadas las energías en un solo término se comparan normalmente.

Así pues, al modificar el Hamiltoniano se debe tomar en cuenta lo siguiente:

- 1) La relación entre energías es lo que ocasiona el cambio. Si la energía de cambio de orientación cristalográfica es menor que la de los componentes, el crecimiento de grano crece con preferencia a los cambios por orientación cristalográfica (manteniendo el concepto de minimización de la energía de libre).



- 2) Si la energía de un componente es menor que la de otro componente, el cambio será preferencial al que posea una energía libre menor (nuevamente recordando la condición de minimización de la energía libre).
- 3) La energía puede variar en función de la dirección. Por ejemplo si consideramos la *Figura 3.2*, es posible que las comparaciones con elementos hacia la derecha aumenten más la energía que comparaciones con elementos a la izquierda. Esto ocasionará un crecimiento de grano con dirección preferencial [1]. Esto no es solo deseable en condiciones con varios componentes, también puede aplicarse en simulaciones con un único componente.

## 4

### **ANÁLISIS Y PRIMEROS RESULTADOS DEL PROGRAMA *MICROSGEN* REALIZADO PARA LA EVOLUCIÓN MICROESTRUCTURAL**

En esta sección, el análisis del programa MicroSGEN se realiza discutiendo las líneas de código generado en MATLAB 7.1<sup>®</sup> en diferentes secciones. Se explica en detalle las variables y funciones utilizadas.



En la *Figura 4.1* se muestra un gráfico de conexión de funciones en donde se parte de un Menú GUI que es manejado por el usuario. Las variables ingresadas en este menú serán llevadas a `animprincipal.m` en donde se inician el ciclo de visualizaciones y llaman a la función `tesis2.m` donde se encuentra el método de Monte Carlo y vuelve a `animprincipal.m`. Luego se llama a la función `tamaniograno.m` en donde se obtiene el tamaño de grano en el tiempo y este resultado vuelve a `animprincipal.m`. Adicionalmente se tiene una función `bordel.m` el cual sirve para crear bordes en las fronteras de los granos. Finalmente existe la función `animacion.m` que sirve para cargar la animación creada con MicroSGEN.

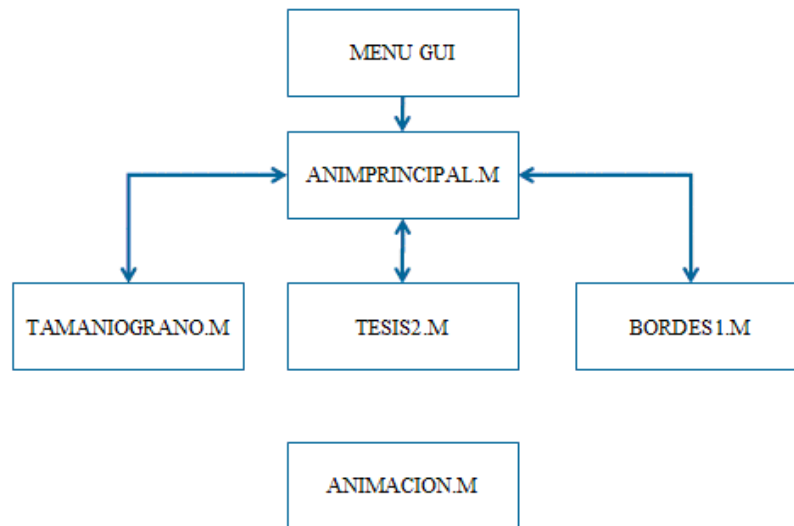


Figura 4.1 Gráfico de conexión de funciones

#### 4.1 Programa codificado en MATLAB 7.1<sup>®</sup> llamado MicroSGen

En MATLAB 7.1 se ha realizado un menú visual para ingresar los datos requeridos por la simulación. En la *Figura 4.1* se muestra dicho menú y un ejemplo de datos ingresados.

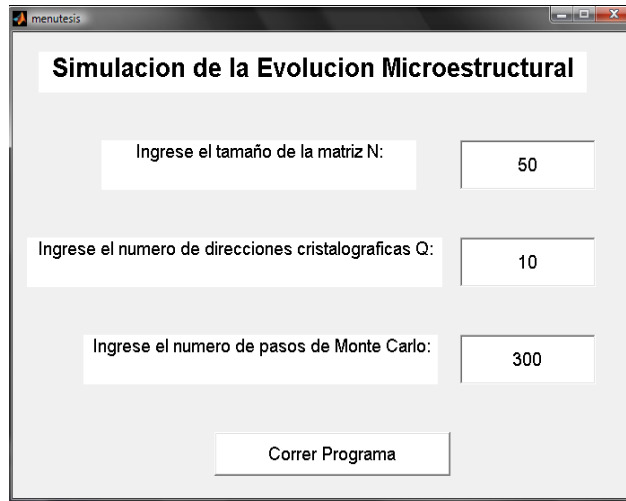


Figura 4.1.1 Menú de ingreso de datos

Para elaborar este menú se escogió dentro de MATLAB la opción de GUIDE (Graphic User Interface) que permite desplegar un gráfico que interactúa entre el usuario y el programa. Los cuadros de texto pasan como variables y se guardan en un espacio temporal de memoria en la computadora. Una vez que se ingresan todos los datos al presionar el botón “Correr Programa” se pasan todas estas variables hacia una función externa llamada `animprincipal.m` de la siguiente manera:

```
N=handles.edit1;      %Se pasa el tamaño de la matriz
Q=handles.edit2;      %Se pasa la cantidad de orientaciones cristalográficas
MCS=handles.edit5;    %Se pasa el número de pasos de Monte Carlo
close
animprincipal(N,Q,MCS) %Todos los datos se pasan a la función
```

Figura 4.2 Código para mandar variables ingresadas en el menú hacia `animprincipal.m`

En la *Figura 4.2*, las variables `edit1`, `edit2` y `edit5` son las variables en donde se ingresa el texto por el usuario que luego son mandadas mediante punteros hacia las variables `N`, `Q` y `MCS` respectivamente. Luego de ello empieza a correr la función `animprincipal.m`.

```

function animprincipal(N,Q,MCS,T)
T1=temp();
R=ceil(rand(N)*Q);           %Generacion de matriz aleatoria
mapa=rand(Q,3);             %Genera una matriz aleatoria de RGB
M=moviein(10,gcf);

```

Figura 4.3 Código para generación de matriz aleatoria y animación

En la *Figura 4.3* se presenta la parte inicial del programa de la función `animprincipal.m` que ingresa a la función de temperaturas necesarias para la simulación. Luego de ello se genera la matriz aleatoria que depende del tamaño de la matriz  $N \times N$  y de la cantidad de orientaciones cristalográficas  $Q$ . Es decir, se genera una matriz  $N \times N$  que llena los espacios con números del 1 hasta el valor  $Q$  ingresado previamente de manera aleatoria. Por otro lado se esta generando una matriz aleatoria con la matriz `mapa`. Dicho `mapa` señala los niveles RGB (Red, Green, Blue). `Mapa` es una matriz de tamaño  $(Q \times 3)$  en donde la primera columna representa los niveles de rojo de su respectiva orientación cristalográfica, la segunda columna representa los niveles de verde y la tercera los niveles de azul. Estos valores se encuentran entre 0 y 1. Por ejemplo, el vector  $[0.5 \ 0.5 \ 0.5]$  representa el color gris. También se genera una matriz `M` la cual contiene las visualizaciones que se presentarán posteriormente en la animación y con la función `moviein` se definen la cantidad de “snapshots” y el `gcf` (get color frame).

El código presentado en la *Figura 4.4* representa el comienzo de la simulación con un ciclo de 10 veces. Los 10 ciclos indican que la animación estará compuesta por 10 imágenes. Cada imagen es una visualización de la evolución del algoritmo. Luego todos los datos ingresados previamente son llevados hacia el programa principal en donde se encuentra el método de Monte Carlo Potts (Ver *Figura 4.5*).

```

for i=1:10
temp(i)
R=tesis2(N,Q,MCS,T1,R); %Llama a la funcion tesis2

```

Figura 4.4 Código que involucra a la función `temp()` que contiene las temperaturas que ingresan en la función `tesis2.m` que contiene al método de Monte Carlo Potts. Además genera los colores que reciben cada orientación cristalográfica en la simulación y animación posteriormente.

```
function final=tesis2(N,Q,MCS,T1,R);

k=5.67e-8;           %Constante de Stefan Boltzman

for m=1:MCS

for f=1:(N*N)       %Ciclo para realizar un MCS

x=ceil(rand(1,1)*N); %generacion randomica de indices
y=ceil(rand(1,1)*N);
```

Figura 4.5 Código inicial del método de Monte Carlo Potts

Como se aprecia en la *Figura 4.5*, para iniciar el programa es necesario definir la constante de Stefan Boltzman que tiene el valor de  $5.67e-8 \text{ W/m}^2\text{K}^4$ . Esta constante se introduce para calcular la probabilidad de Metrópolis dentro del programa. La función `tesis2` recibe como datos las variables `N`, `Q`, `MCS`, `T1` y `R`. Se comienza la simulación mediante un ciclo principal que termina cuando se han alcanzado los pasos de Monte Carlo (MCS) especificados. Dentro del ciclo principal encontramos el ciclo que realiza las repeticiones necesarias para un único paso de Monte Carlo ( $N \times N$ ). Se generan aleatoriamente las coordenadas  $x$ ,  $y$  que son útiles para ubicar la posición deseada en la matriz. De esta manera se recorre a través de la matriz  $N \times N$  y se comienza a comparar.

En la *Figura 4.6* se muestra las comparaciones del elemento seleccionado con coordenadas  $(x, y)$  en el caso de que sea una esquina con coordenadas  $(1,1)$ . Este código toma en cuenta las condiciones de borde periódicas mencionadas en la sección 3.4. Como se puede observar las comparaciones se realizaron mediante decisiones lógicas *if*. Este es un trabajo tedioso y el código resultante parece ser poco eficiente. Sin embargo, la formulación de las condiciones por medio de decisiones *if* permite modificar fácilmente el programa para colocar diferentes condiciones de borde.

El código presentado es específico para una primera esquina de coordenadas (1,1) y mediante pequeños cambios se puede adaptar para las tres esquinas restantes (1, N), (N, 1), (N, N). La línea de código  $E1(p) = E1(p) + 1$  aumenta la energía libre a la que llamamos con un vector  $El$ . El índice  $p$  muestra si es el primer ciclo de comparación o el segundo. Hay que recordar que primero se compara la posición escogida con sus vecinos y luego se reemplaza otro valor de orientación cristalográfica en la misma posición para realizar una segunda comparación.

```

for p=1:2
El(p)=0;

if (x==1 & y==1)           %Primera esquina
    if (R(1,1)==R(1,2))
        El(p)=El(p)+1;      %El = Energia libre
    end;
    if (R(1,1)==R(2,2))
        El(p)=El(p)+1;
    end;
    if (R(1,1)==R(2,1))
        El(p)=El(p)+1;
    end;

    %Comparacion periodica
    if (R(1,1)==R(1,N))
        El(p)=El(p)+1;
    end;
    if (R(1,1)==R(N,N))
        El(p)=El(p)+1;
    end;
    if (R(1,1)==R(N,1))
        El(p)=El(p)+1;
    end;
    if (R(1,1)==R(N,2))
        El(p)=El(p)+1;
    end;
    if (R(1,1)==R(2,N))
        El(p)=El(p)+1;
    end;
    %FIN DE LA COMPARACION PERIODICA

```

Figura 4.6 Código para comparación del elemento seleccionado (1,1) en una esquina

La *Figura 4.7* muestra el código para las comparaciones entre el elemento  $x, y$  que cae en el lado superior de la matriz con la excepción de las posiciones que tienen coordenadas (1, 1) y (1, N). De la misma manera la energía libre aumenta como se mencionó anteriormente. Asimismo, este código es útil para los otros tres lados de la matriz y siempre tomando en cuenta las condiciones de borde periódicas establecidas.

```

else if(x==1) & (y~=1) & (y~=N) %LADO SUPERIOR
    if (R(x, y)==R(x, y-1)) %Compara con los puntos aledaños
        El(p)=El(p)+1;
    end;
    if (R(x, y)==R(x, y+1))
        El(p)=El(p)+1;
    end;
    if (R(x, y)==R(x+1, y-1))
        El(p)=El(p)+1;
    end;
    if (R(x, y)==R(x+1, y))
        El(p)=El(p)+1;
    end;
    if (R(x, y)==R(x+1, y+1))
        El(p)=El(p)+1;
    end;

    %Comparacion periodica
    if (R(x, y)==R(N, y-1))
        El(p)=El(p)+1;
    end;
    if (R(x, y)==R(N, y))
        El(p)=El(p)+1;
    end;
    if (R(x, y)==R(N, y+1))
        El(p)=El(p)+1;
    end;
%FIN DE LA COMPARACION PERIODICA

```

Figura 4.7 Código para comparación del elemento seleccionado en el lado superior

Finalmente, la última sección de comparación se realiza en el centro de la matriz. En estas posiciones no es necesario establecer las condiciones de borde. Se procede a realizar las ocho comparaciones directamente como se muestra en la *Figura 4.8*.



```

else                                     %CENTRO
    if (R(x,y)==R(x,y-1))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x-1,y-1))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x-1,y))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x-1,y+1))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x,y+1))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x+1,y+1))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x+1,y))
        El(p)=El(p)+1;
    end;
    if (R(x,y)==R(x+1,y-1))
        El(p)=El(p)+1;
    end;
end

```

Figura 4.8 Código para comparación del elemento seleccionado (x, y) en el centro

Las decisiones mostradas en la *Figura 4.9* se utilizan para generar una nueva orientación cristalográfica en las mismas coordenadas generadas al inicio del ciclo. El valor inicial que se utiliza para la primera comparación se guarda en la variable `valorin` y la nueva orientación generada se guarda en la variable `valorfin`. Al finalizar la segunda comparación la matriz *R* vuelve al estado original.

```

if (p==1)
    valorin=R(x,y);
    R(x,y)=(ceil(rand(1,1)*Q));
end;
if (p==2)
    valorfin=R(x,y);
    R(x,y)=valorin;
end;

```

Figura 4.9 Código para guardar el primer valor de  $E_l$  y generar una nueva orientación cristalográfica en la misma posición

En la *Figura 4.10* se comparan ambas energías por medio de la definición de la variable  $\Delta E$ . Es necesario observar que  $E_l(1)$  y  $E_l(2)$  no están directamente relacionados con la definición de la energía libre de Gibbs. En todo el programa se han contado los vecinos semejantes de la orientación en la posición escogida. La energía del sistema aumenta cuando los vecinos de las posiciones seleccionadas son diferentes. El programa se simplifica si  $E_l$  cuenta los vecinos semejantes, e inclusive es fácil de implementar la probabilidad de Metrópolis. Solo es necesario tener en cuenta que para minimizar la energía libre se debe seleccionar el mayor  $E_l$ . En el caso que  $E_l(2)$  no sea mayor que  $E_l(1)$  se acepta  $E_l(2)$  a favor de la probabilidad de Boltzman. ¿Cómo se simula esta probabilidad? Se utiliza de nuevo el concepto de Monte Carlo, es decir, se selecciona un número al azar y se verifica si el número se encuentra dentro de la probabilidad calculada. En caso de que la variable aleatorio sea menor que la variable probabilidad se acepta  $E_l(2)$ . Esto es similar a lo que sugiere la *ecuación 2.24* descrita en el capítulo 2.

```
deltaE=E1(1)-E1(2);    %Cambio de energia
if (deltaE<=0)
    R(x,y)=valorfin;
else if ((k*T1(x,y))>0)
    probabilidad=exp((-deltaE)/(k*T1(x,y)));
    aleatorio=(rand(1,1));
    if aleatorio<probabilidad
        R(x,y)=valorfin;
    end
end
end
```

Figura 4.10 Código en donde se comparan las energías libres y se encuentra la probabilidad de Metrópolis

En la *Figura 4.11* se muestra el método para poder capturar la matriz y mediante la función `image(R)` se genera una imagen de la matriz  $R$ . Se utiliza como matriz de colores a la matriz señalada por la función `colormap`. Luego de esto se definen los parámetros y espacio en memoria para generar la animación.

```

image(R); %Exhibe la matriz como imagen
colormap(mapa); %Da el color respectivo a la imagen
M(:,i)=getframe(gcf);
end;
n=10; fps=1; save m grano M n fps

```

Figura 4.11 Código que genera la imagen de cada matriz generada por los pasos anteriores

En la *Figura 4.12* se muestra el programa `animación.m` que es otra función externa para cargar la animación generada previamente con la función `animprincipal.m`.

```

gcf; cla; clf;
load m_grano
loc=[30,30,0,0];
movie(gcf,M,1,0.2,loc)

```

Figura 4.12 Código para visualizar la animación generada

#### 4.1.1 Programa compacto para establecer las condiciones de borde periódicas

Una forma más eficiente para establecer las condiciones de borde periódicas es por medio de operaciones matemáticas realizadas sobre las coordenadas seleccionadas. En este caso es útil la función `mod` que obtiene el residuo de una división. El método está ligeramente sugerido en el texto “Computational Materials Engineering” de Koenraad Janssens [ 7 ] pero con ciertos errores detectados. A continuación se el código utilizando la nueva función.

```

for i=1:20

for m=1:MCS
    for f=1:(N*N)
        x=ceil(rand(1,1)*N);
        y=ceil(rand(1,1)*N);

        xv156=mod((x+1),N);

        xv278=mod((x+N-1),N);

        yv358=mod((y+N-1),N);

        yv467=mod(y+1,N);

        if(xv156==0)
            xv156=N;
        end
        if(xv278==0)
            xv278=N;
        end
        if(yv358==0)
            yv358=N;
        end
        if(yv467==0)
            yv467=N;
        end
    end
end

```

Figura 4.13 Código utilizando la función mod [ 7 ]

Como se puede observar en la *Figura 4.13*, el código que define las variables  $xv156$ ,  $xv278$ ,  $yv358$ , y  $yv467$  se utiliza para determinar las coordenadas de los vecinos. Se han nombrado de esta manera para identificar las coordenadas en  $x$  y  $y$  de cada uno de los ocho vecinos. De esta forma  $xv156$  corresponde a la coordenada  $x$  de los vecinos llamados 1, 5 y 6.

El código mostrado en la *Figura 4.14* indica como se realizan las comparaciones del elemento seleccionado  $x$ ,  $y$  con los vecinos cercanos pero utilizando ahora las variables  $xv156$ ,  $xv278$ ,  $yv358$ , y  $yv467$ . Al programar de esta forma el código se vuelve más compacto que el mostrado en las *Figura 4.6*, *4.7* y *4.8* el cual evita que sea tediosa la programación. La desventaja es que no puede ser utilizado para una simulación

en 3D o para colocar cualquier otro tipo de condición de borde indicado en la sección 3.4 que no sea una periódica. El código es eficiente pero poco flexible.

```

for p=1:2
    E1(p)=0;

    if (R(x,y)==R(xv156,y))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(xv278,y))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(x,yv358))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(x,yv467))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(xv156,yv358))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(xv156,yv467))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(xv278,yv467))
        E1(p)=E1(p)+1;
    end;
    if (R(x,y)==R(xv278,yv358))
        E1(p)=E1(p)+1;
    end;
end;

```

xv156, yv467	xv156, v	xv156, yv358
x, yv467	x, v	x, yv358
xv278, yv467	xv278, v	xv278, yv358

Figura 4.14 Código de comparación de vecinos cercanos para cualquier elemento seleccionado x,y de acuerdo al recuadro de la derecha.

Suponiendo que existen dos fases en la microestructura y queremos diferenciar los granos de cada uno de estos, se debe encontrar un método para observar todos los granos que pertenecen a dicha fase en un mismo color. Esto se puede lograr mediante una definición correcta de los bordes de grano. El programa que se presenta a continuación es útil para graficar los bordes de grano.

```

for y=1:N1
    for x=2:N1
        if T(x,y)==T(x-1,y)
            C(x,y)=1;
        else
            C(x,y)=2;
        end;
        if T(y,x)==T(y,x-1)
            M(y,x)=1;
        else
            M(y,x)=2;
        end;
    end;
end;
for i=1:N1
    for j=1:N1
        for j=1:N1
            if (C(i,j)==2) || (M(i,j)==2)
                P(i,j)=2;
            else
                P(i,j)=1;
            end;
        end;
    end;
end;

mapa=[0 0 1; 0 0 0];
image(P);
colormap(mapa);

```

Figura 4.15 Definición de bordes de grano

En la *Figura 4.15* se puede observar como se obtuvo la definición de los bordes para graficarlos. Se utilizan tres matrices M, C y P. M verifica el cambio entre orientaciones cristalográficas haciendo comparaciones en todas las filas. C realiza las comparaciones en las columnas y P interseca los resultados en las matrices M y C. Las posiciones en la matriz que se encuentran en los bordes de granos están definidos con el número 2.

#### 4.1.2 Función para calcular el tamaño de grano

Con el objetivo de verificar la evolución del tamaño de grano, se requiere de un método para medirlos a medida que avanza la simulación. Existen métodos diversos para realizar esta medición en micrografías reales. Debido al gran número de granos

presentes en la muestra metalográfica, los métodos recomendados son métodos sencillos y prácticos. A continuación se detalla algunos de éstos métodos de acuerdo a lo que indica la norma ASTM.

La Sociedad Americana para Prueba de Materiales (American Society for Testing Materials) propone 3 métodos para realizar la medición. En todos los métodos se encuentra un valor  $G$  que se relaciona con el número de granos en una micrografía con un aumento de 100X de acuerdo a la *ecuación 2.28*.

$$N = 2^{G-1} \quad (2.28)$$

Donde  $N$  es el número de granos por pulgada cuadrada a 100X. Los métodos que se describen a continuación pueden servir de guía para elaborar un algoritmo debido a que son métodos bastante prácticos.

### **1) Método de comparación:**

La norma ASTM expone muestras de tamaño de grano que pueden ser comparadas con muestras reales a 100x. Las muestras estándar están divididas en cuatro categorías para poder realizar esta comparación adecuadamente: los platos I, II, III, IV. Cada plato tiene graficadas muestras de granos de diferentes tamaños y de diferentes formas. Es posible imprimir las muestras en transparencias, comparar a contraluz, y decidir cual es la que mejor coincide con la muestra real. La norma indica un valor  $G$  dependiendo del plato utilizado.

### **2) Método de Jeffries o planimétrico:**

Este método propone el conteo de los granos encontrados en un recuadro o círculo sobre la muestra a 100x. La norma ASTM indica que el área de conteo debe

tener 5000 mm<sup>2</sup>. La muestra debe incluir al menos 50 granos. Los granos que no estén completamente dentro del área deben ser contados como medios granos. Se obtiene así el número de granos por milímetro cuadrado  $NA$  multiplicando el número obtenido por un factor de acuerdo a la magnificación encontrada en la norma. El  $NA$  puede compararse con la tabla 4 de la norma E112-96 para obtener el valor  $G$  de tamaño de grano. También se puede utilizar la *ecuación 2.29*.

$$G = (3.321928 \log_{10} NA) - 2.954 \quad (2.29)$$

$NA$  es el número sobre mm<sup>2</sup> a 1x.

### **3) Método de intercepción o de Heyn:**

El tamaño de grano puede estimarse contando el número de granos intersecados por una línea recta de longitud arbitraria a 100x. Se necesitan al menos 50 granos dentro de la muestra para poder realizar este método. Se deben graficar varias líneas para obtener un promedio y minimizar los errores. Los bordes de la línea deberán ser contados como medios granos si rebasan la intersección con el borde de un grano. De esta manera se obtiene el número de granos por mm. La tabla 4 de la norma ASTM E112-96 indica a qué tamaño de grano  $G$  corresponde el número de granos por mm.

Parecería que el más preciso de estos métodos para ser utilizado dentro del programa es el método planimétrico. En nuestro trabajo, el método consiste en identificar los diferentes números de la matriz de simulación y los bordes de grano. La idea plantea contar todos los números semejantes de orientaciones cristalográficas dentro de un borde de grano. El resultado sería el número de píxeles que ocupa cada grano y por lo tanto el área del mismo. Si bien es la forma más precisa para realizar el conteo, el algoritmo es complicado. La forma más sencilla exigiría realizar un conteo manual al inicio de la simulación para dar un nombre específico a cada grano y



monitorear su crecimiento durante la simulación. Esto resulta muy poco práctico. La *Figura 4.16* muestra el código para calcular el tamaño de grano en el tiempo.

```
for i=1:(N1-1)
    if R1(i,alet1(1))~=R1(i+1,alet1(1))
        contador1=contador1+1
    end;
end;
for j=1:(N1-1)
    if R1(alet1(2),j)~=R1(alet1(2),j+1)
        contador2=contador2+1
    end;
end;
result1=(N1-1)/contador1;
result2=(N1-1)/contador2;
result3=(vector3*(sqrt(2)))/contador3;
mean=(result1+result2+result3)/3
tama=mean;
```

Figura 4.16 Código para calcular el tamaño de grano en el tiempo

El método de intercepción es un método que puede ser aplicado muy fácilmente. Se puede definir vectores que atraviesen la matriz en lugares arbitrarios. Se puede generar de forma simple un vector en una columna, uno en una fila, y otra en la diagonal. El flujograma en la *Figura 4.17* muestra el esquema básico.

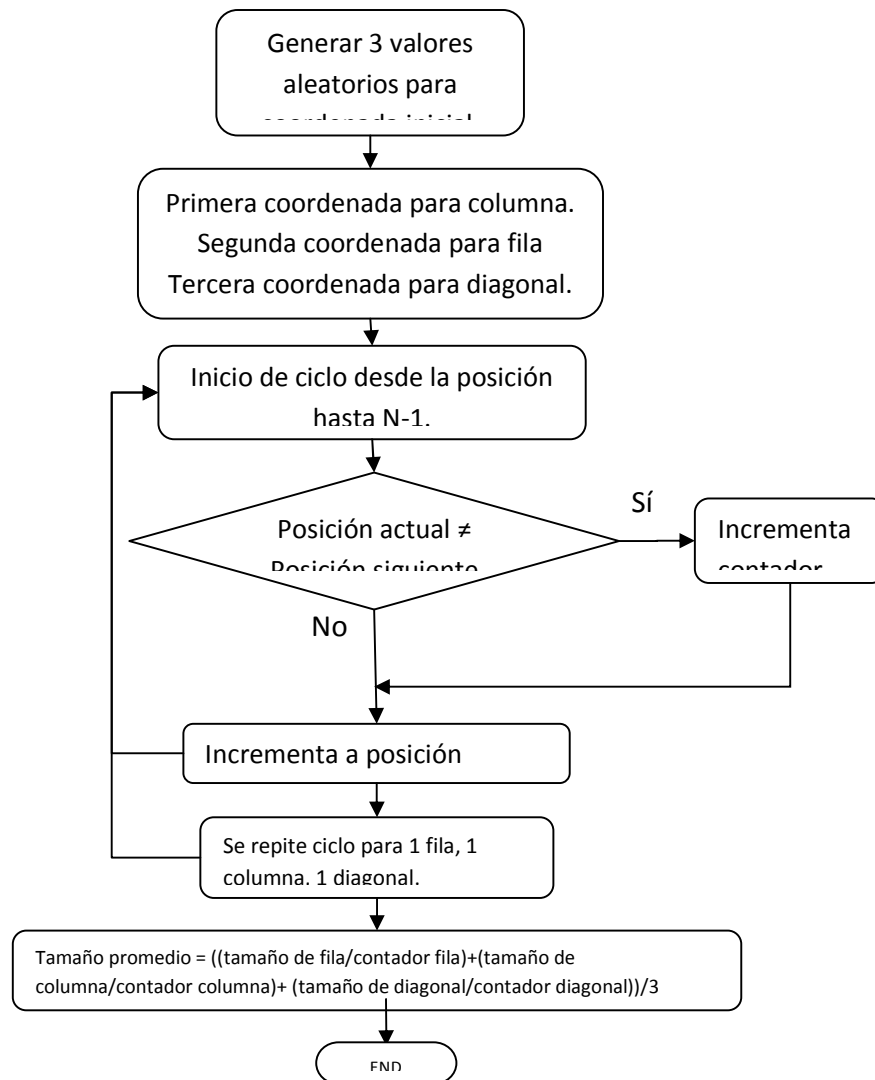


Figura 4.17 Flujograma de la función para medir el tamaño de grano.

#### 4.1.3 Resumen de las variables que serán manejadas:

A continuación se presenta un listado con las variables y las constantes más importantes que son manejadas dentro del programa.

**N:** número de columnas o filas asignadas a la matriz NxN.

**R:** matriz aleatoriamente completada para simular orientaciones cristalográficas.

**Q:** número máximo de diferentes orientaciones cristalográficas.

**MCS:** pasos de Montecarlo, señala el tiempo de la simulación.

**T1:** temperatura en la función de aceptación de metrópolis (no se relaciona con la temperatura real)

**k:** constante de Boltzmann.  $k \cdot T1$  reduce el efecto de la geometría seleccionada sobre los resultados de la simulación.

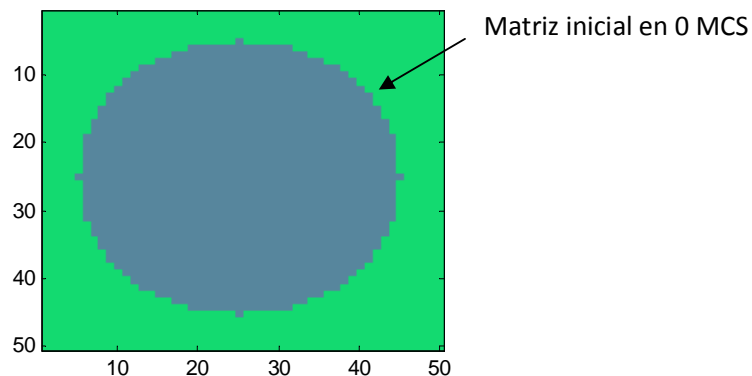
**u:** movilidad del borde de grano, u siempre debe tener un valor menor a 1.

**EI():** energía que aumenta de acuerdo a lo señalado por el Hamiltoniano. Puede variarse el peso de la energía para obtener fenómenos diferentes entre componentes y direcciones cristalográficas.

#### 4.2 Análisis y resultados obtenidos con el programa MicroSGEN

Una vez realizado el código del programa MicroSGEN es necesario entender su comportamiento por medio de algunos ejercicios. Estos ejercicios pueden ayudar posteriormente a implementar funciones más complejas al programa para que la simulación se acerque más a la realidad.

Para entender la manera en la que un grano cambia de forma por medio del método de Monte Carlo, se propone evaluar la evolución de un círculo que se encuentra en la mitad de una matriz. El círculo en las *Figuras 4.18 y 4.19* representan un grano reduciéndose.



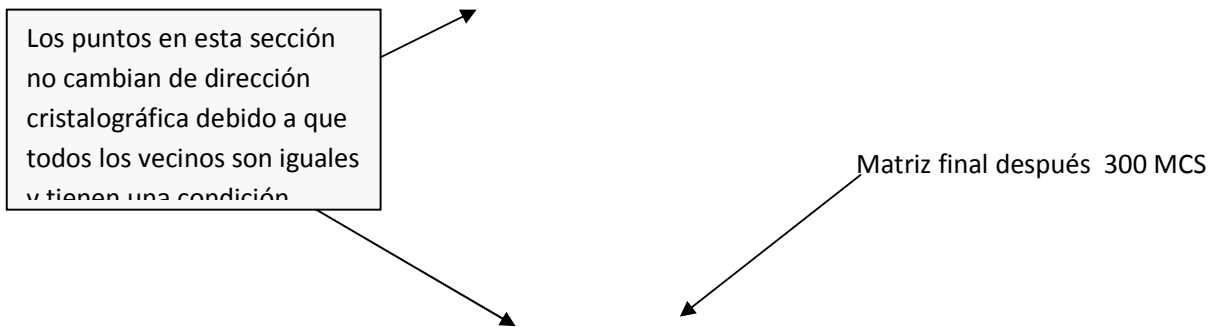
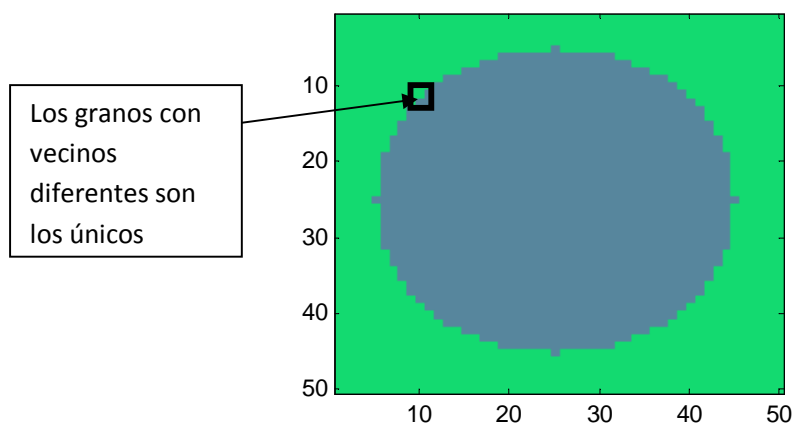


Figura 4.18 Reducción de un grano circular luego de 300 MCS



Dentro del círculo hay mayor número de granos que pueden cambiar. Los

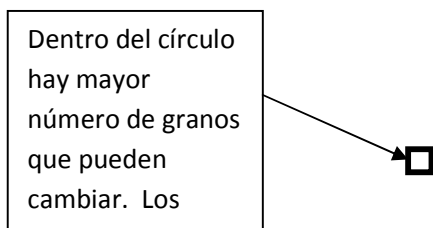


Figura 4.19 Continuación del análisis de la reducción de un grano circular

Una vez comprendido el programa utilizando únicamente dos direcciones cristalográficas se procede a probar un sistema más complejo con parámetros de la Tabla 4.1. El algoritmo de Q Potts presenta los resultados:

<b>Matriz 1</b>	30x30
<b>Lattice:</b>	Cuadrada
<b>Orientaciones Cristalográficas:</b>	6
<b>Forma de grano inicial:</b>	desorganizado
<b>kTs:</b>	0
<b>MCS:</b>	300

Tabla 4.1 Datos para simulación de una matriz 30x30 con 6 orientaciones cristalográficas

0 MCS

300 MCS

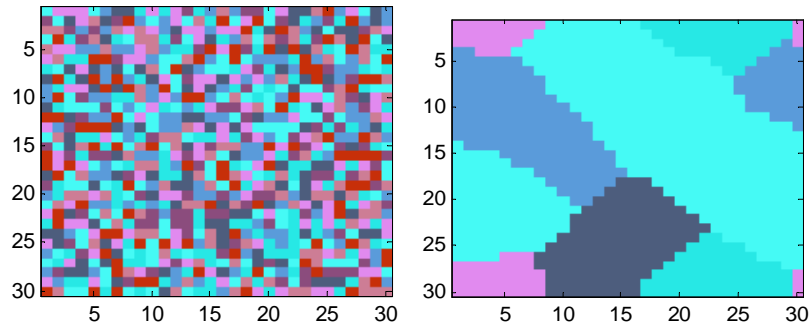


Figura 4.20 Demostración de la organización de los granos iniciando con una matriz aleatoria.  
Resultados obtenidos al ingresar los parámetros de la tabla 4.1.

Se corrió el programa una segunda vez con diferentes parámetros para observar diferencias.

<b>Matriz 2</b>	50x50
<b>Lattice:</b>	Cuadrada
<b>Orientaciones Cristalográficas:</b>	10
<b>Forma de grano inicial:</b>	Desorganizado
<b>kTs:</b>	0
<b>MCS:</b>	300

Tabla 4.2 Datos para simulación de una matriz 50x50 con 10 orientaciones cristalográficas

A continuación en la *Figura 4.21* se muestran los resultados obtenidos al ingresar los parámetros de la tabla 4.2.

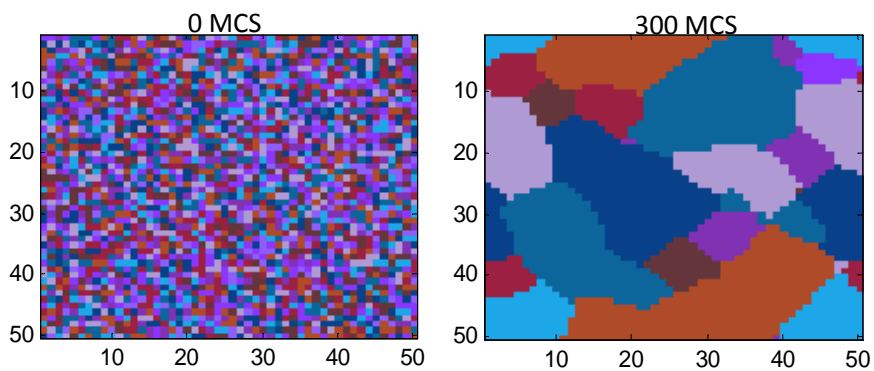


Figura 4.21 Demostración de la organización de los granos iniciando con una matriz aleatoria

En la primera simulación de la *Figura 4.20* se observa que en el resultado luego de 300 pasos de Monte Carlo, las orientaciones cristalográficas indicadas con el color rojo han desaparecido. El mismo fenómeno se observa en la *Figura 4.21* con la desaparición de algunos colores en el resultado. La conclusión evidente es que los granos mayores toman el espacio de los granos más pequeños y por consiguiente los granos pequeños desaparecen.

Para demostrar el comportamiento de la función para medir el tamaño de grano y la función para generar bordes se ha realizado una simulación con los parámetros de la tabla 4.3.

<b>Matriz 3</b>	100x100
<b>Lattice:</b>	Cuadrada
<b>Orientaciones Cristalográficas:</b>	15
<b>Forma de grano inicial:</b>	Desorganizado
<b>kTs:</b>	kTs
<b>MCS:</b>	30/visualización
<b>#visualizaciones</b>	10

Tabla 4.3 Datos para simulación de una matriz 100x100

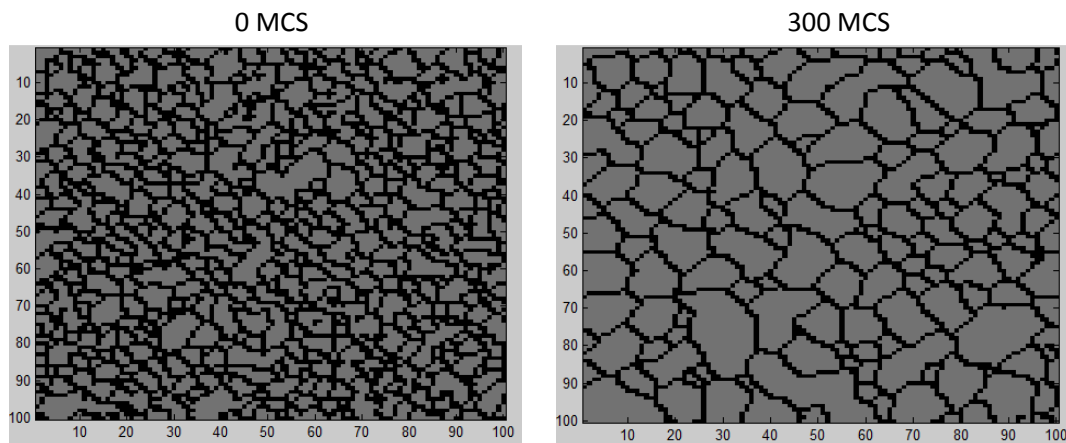


Figura 4.22 Demostración de la diferenciación de bordes

Como se puede observar en la *Figura 4.23*, el tamaño de grano aumenta de forma casi constante a medida que aumentan los pasos de Monte Carlo. Partiendo de esta recta podemos sacar una constante K que representa el coeficiente de crecimiento de grano para la simulación. Una constante K puede ser obtenida experimentalmente para cualquier material. Igualando estas constantes podremos tener una simulación de la evolución microestructural con resultados aplicables a la realidad.

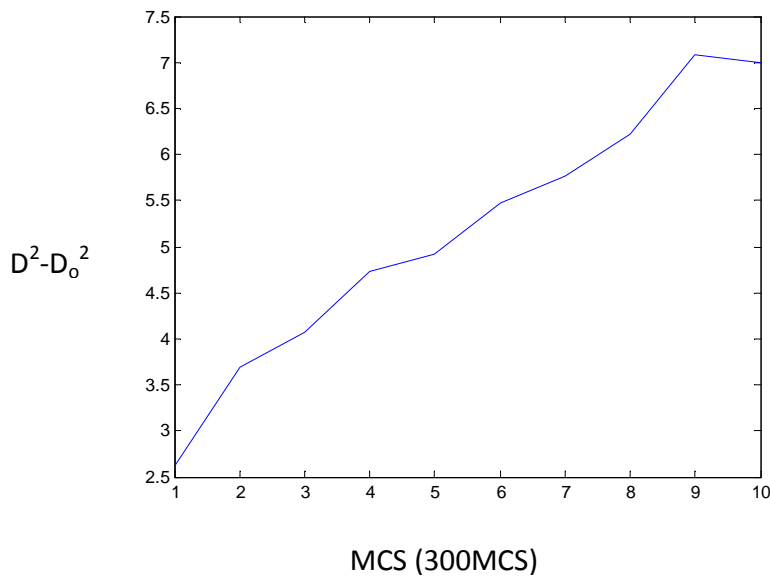


Figura 4.23 Tamaño de grano vs pasos de Monte Carlo



# 5

## SIMULACIÓN CON COMSOL Y MICROSGEN

### 5.1 Definición de tratamientos térmicos

Un tratamiento térmico es un proceso muy conocido por el área ingenieril. Dicho proceso tiene como base someter a tratamientos específicos a un metal u otro tipo de material con el propósito de mejorar sus propiedades mecánicas. Los tratamientos térmicos se hacen principalmente a materiales con hierro y carbono como el acero y la fundición. Estos tratamientos térmicos no modifican la composición química solo modifican su estructura cristalina. Al ser modificada la estructura cristalina lo que se puede tener son diferentes estructuras de grano con la misma composición química y a esto se le conoce como polimorfismo.

Un tratamiento térmico consta de someter al material a distintos procedimientos que constan de cambios bruscos de temperatura. El propósito final es lograr que las propiedades mecánicas del material aumenten. Las propiedades mecánicas que se busca aumentar en general son la dureza, la tenacidad, la maquinabilidad y la resistencia al desgaste y esto sirve posteriormente para aplicaciones como el forjado, o el estampado. Es importante cubrir al acero con limalla antes de realizar cualquier calentamiento de la pieza en el horno. Se recomienda hacer esto ya que se previene la descarburización que podría afectar al tratamiento térmico [ 12 ].

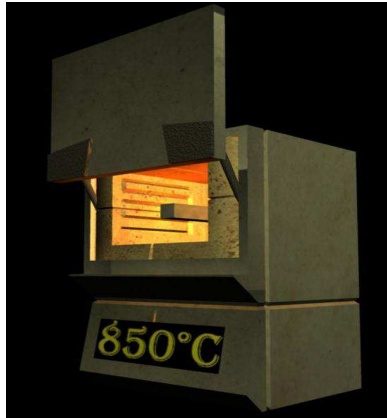


Figura 5.1 Calentamiento de la probeta de acero en horno para realizar un tratamiento térmico.

Para esta simulación se ha escogido el acero ya que es una aleación entre hierro, carbono y otros elementos más. El tener varios elementos aleantes es lo que logra que este material tenga buenas propiedades mecánicas y sea uno de los más usados en la industria.

## 5.2 Tratamientos térmicos

Existen varios tratamientos térmicos que se usan en la industria pero el que se realizó en la simulación es el recocido y a continuación se hace una breve reseña de que se trata:

- **Recocido:** En este tratamiento térmico se calienta al acero a temperaturas altas por encima de la temperatura de austenización. Luego se mantiene la probeta dentro del horno para asegurar su homogenización, aproximadamente unos 40 minutos. Después, el horno es apagado y se deja enfriar lentamente el acero dentro del horno. Con este tratamiento se logra ablandar al acero, y adicionalmente las tensiones internas son eliminadas. Además, por medio de este proceso se obtiene una mayor tenacidad y un acero con mayor maquinabilidad. Este tratamiento térmico es muy útil para un posterior trabajo en frío [ 12 ]. En la *Figura 5.2* se muestra este tratamiento térmico.

°C

Tiempo de homogenización  
con el horno prendido

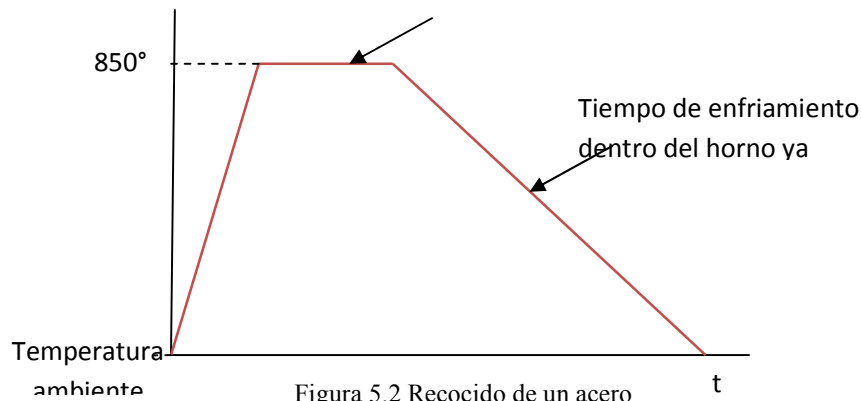


Figura 5.2 Recocido de un acero

### 5.3 Simulación de recocido utilizando COMSOL Multiphysics

COMSOL Multiphysics<sup>®</sup> es un paquete de elementos finitos que sirve para realizar varias simulaciones sobre problemas ingenieriles. Lo importante de este paquete es que incluye una interfaz a MATLAB 7.1<sup>®</sup> y ambos usan el mismo lenguaje y adicionalmente un ambiente amplio para programación y visualización gráfica. En COMSOL se puede utilizar las mismas funciones que en MATLAB lo cual ahorra mucho tiempo al rato de sacar resultados de COMSOL hacia MATLAB. Dentro de este software existen varios módulos ya programados en base a distintas áreas de la ingeniería.

El módulo que se utilizó para esta simulación es el de *Transferencia de Calor*. Se seleccionó este módulo debido a que es el necesario para poder hacer la simulación del tratamiento térmico elegido. El módulo de transferencia de calor nos brinda una amplia librería de ecuaciones diferenciales que contienen las ecuaciones de conducción, convección y radiación. Partiendo de este módulo se hizo la simulación del tratamiento térmico para que posteriormente la información de las temperaturas obtenidas en este paquete sean llevadas hacia MATLAB 7.1<sup>®</sup> directamente al exportar toda la información hacia un archivo *.m* que es entendido por MATLAB. Una vez obtenidas dichas temperaturas se las podrá ingresar en el programa realizado para la simulación de la evolución microestructural. A continuación se muestra el proceso realizado en COMSOL Multiphysics para simular el tratamiento térmico de recocido seleccionado para este caso en particular.

1) Definición de los dominios que deberán analizarse por el método de elementos finitos. Simplemente se grafica en el ambiente CAD de COMSOL las diferentes geometrías para su posterior análisis. Para simular un tratamiento de recocido, como se ha indicado en la sección 5.2, es preciso mantener la probeta dentro del horno refractario o mufla hasta que alcance la temperatura ambiente. Por esta razón se ha considerado graficar 3 dominios. El dominio interior es una geometría que representa la probeta metálica. Un dominio intermedio representa el aire entre la probeta y las paredes del horno o mufla. El dominio exterior representa los ladrillos refractarios característicos de un horno para tratamientos térmicos de laboratorio.

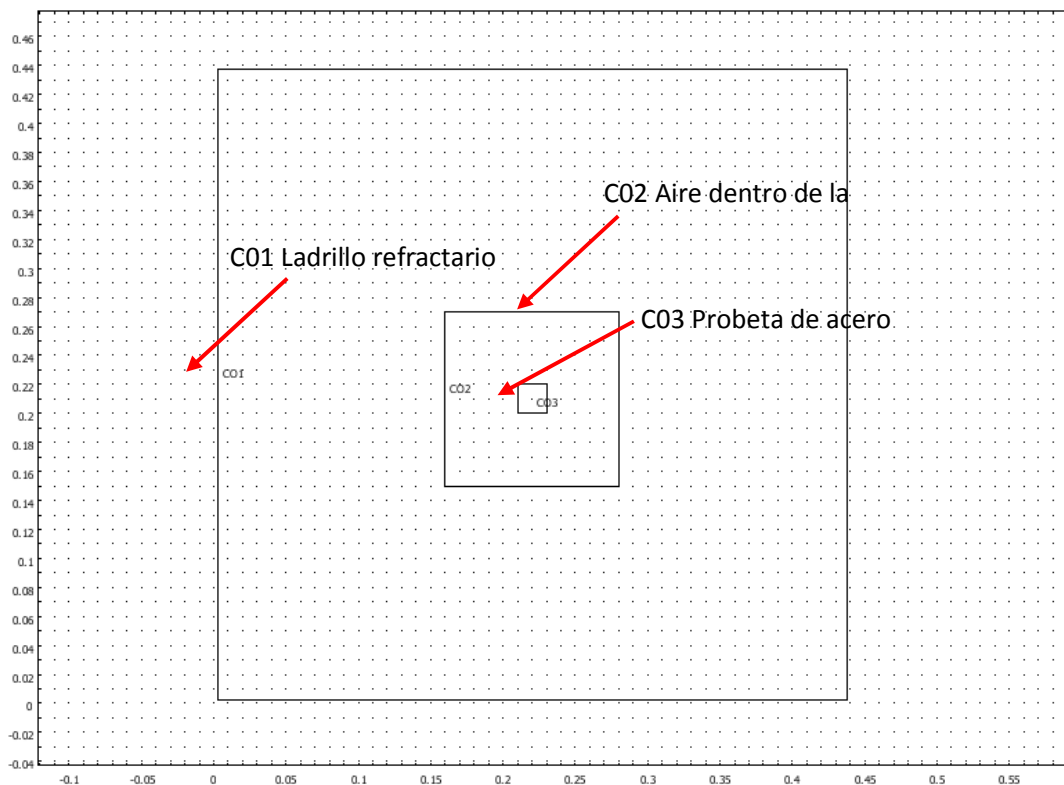


Figura 5.3 Geometrías de los dominios para simulación de recocido

En la *Figura 5.3* se aprecia los dominios que se analizarán por el método de elementos finitos. La probeta aparece señalada como C03, el aire dentro de la mufla aparece señalada como C02 y las paredes de ladrillo refractario están señaladas como C01. Las diferentes dimensiones de los dominios son tomadas de ejemplos reales. Es posible encontrar en el mercado hornos de tratamiento térmico del tamaño propuesto que

alcanzan las temperaturas necesarias para recocido. La Tabla 5.1 resume las diferentes dimensiones de los dominios.

	Ancho (m)	Alto (m)
<b>C01 Exterior de mufla</b>	0.445	0.445
<b>C02 Interior de mufla</b>	0.120	0.120
<b>C03 Area transversal de probeta.</b>	0.02	0.02

Tabla 5.1 Dimensiones de las geometrías [ 4 ].

- 2) Luego de realizadas las geometrías se deben definir las propiedades de cada dominio. COMSOL agrupa las geometrías en subdomains. A cada subdomain es posible asignar propiedades características de un material. COMSOL mantiene una librería de materiales que se utilizan frecuentemente en ingeniería. También es posible definir materiales nuevos y sus propiedades. Dependiendo del módulo seleccionado en COMSOL, el programa necesita el ingreso de diferentes tipos de propiedades.

En el caso del módulo de transferencia de calor en 2D, se debe ingresar la constante de conductividad térmica  $k$ , el calor específico  $C_p$ , la densidad  $\rho$ , y una fuente de calor  $Q$  en caso de que existiese. Es interesante observar que  $k$  puede ser considerada isotrópica pero COMSOL permite el ingreso de una constante  $k$  anisotrópica.

Para el subdomain C03 que representa la probeta metálica se ha seleccionado el acero AISI 4340 de la librería de COMSOL. En la *Figura 5.4* aparecen las propiedades en las unidades utilizadas.

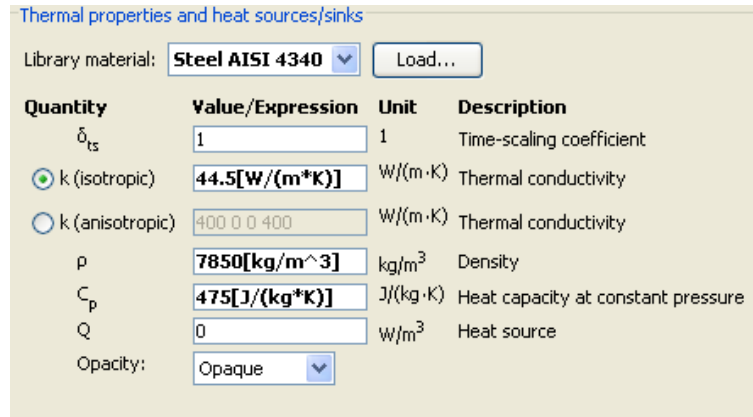


Figura 5.4 Propiedades térmicas del acero AISI 4340 definidas en ambiente de COMSOL

Para el subdomain C02 que representa el aire existente dentro de la cámara interna de la mufla. Debido a que COMSOL carece de las propiedades térmicas del aire, se encontró en otras referencias las propiedades. En la *Figura 5.5* aparecen dichas propiedades en las unidades utilizadas.

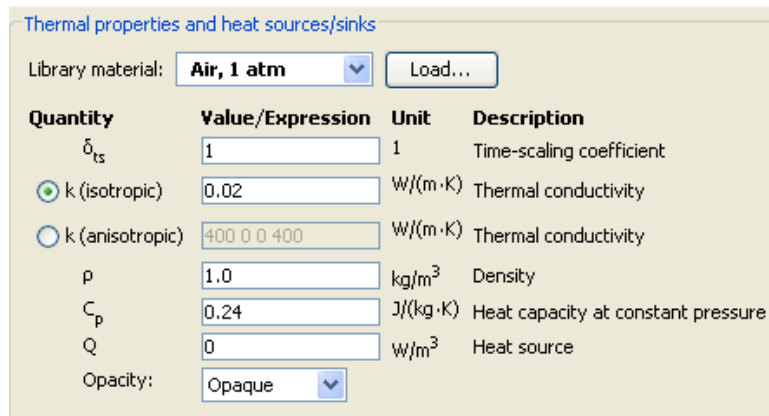


Figura 5.5 Propiedades térmicas del aire a 1 atm definidas en ambiente de COMSOL [ 5 ].

Para el subdomain C01 que representa los ladrillos refractarios se ha elegido un valor bajo de conductividad térmica. De la misma referencia [ 5 ] los valores de la conductividad térmica del ladrillo refractario caen en un rango desde 0.47 a 1.05 W/m K. Se ha elegido 0.5 porque es un valor que asegura la eficiencia del ladrillo en cualquier mufla. La mayoría de hornos para tratamientos térmicos utiliza ladrillos con un coeficiente de conductividad cercano al límite inferior del rango propuesto

para obtener mejores eficiencias. En la *Figura 5.6* aparecen las propiedades definidas para el ladrillo refractario.

The screenshot shows the 'Thermal properties and heat sources/sinks' dialog box in COMSOL. The 'Library material' is set to 'RefractoryBrick'. The following table lists the defined properties:

Quantity	Value/Expression	Unit	Description
$\delta_{ts}$	1	1	Time-scaling coefficient
<input checked="" type="radio"/> k (isotropic)	0.5	W/(m·K)	Thermal conductivity
<input type="radio"/> k (anisotropic)	400 0 0 400	W/(m·K)	Thermal conductivity
$\rho$	1900	kg/m <sup>3</sup>	Density
$C_p$	836	J/(kg·K)	Heat capacity at constant pressure
Q	0	W/m <sup>3</sup>	Heat source
Opacity:	Opaque		

Figura 5.6 Propiedades térmicas del ladrillo refractario definidas en ambiente de COMSOL [ 5 ].

3) A continuación se necesita definir las condiciones iniciales de temperatura. En COMSOL estas se definen dentro de las propiedades de los subdomains. En los subdomains C03 y C02 se inicializa la temperatura de acuerdo a la *Figura 5.7*.

The screenshot shows the 'Initial value' dialog box for subdomain C03 and C02. The initial temperature  $T(t_0)$  is set to 1123.15 K.

Figura 5.7 Condición inicial de temperatura para el aire y la probeta metálica definida en ambiente de COMSOL

En el subdomain C01 que representa las paredes de la mufla se inicializa con una temperatura mucho menor de acuerdo a la *Figura 5.8*.

The screenshot shows the 'Initial value' dialog box for subdomain C01. The initial temperature  $T(t_0)$  is set to 360.05 K.

Figura 5.8 Condición inicial de temperatura para el ladrillo refractario definida en ambiente de COMSOL

Se ha inicializado de esta manera ya que se asume que en una primera etapa se ha alcanzado la temperatura necesaria para el recocido sobre la temperatura de

austenización. Esta temperatura se mantiene dentro del horno mientras que la superficie exterior debe ser mucho menor para evitar la pérdida de calor.

- 4) El siguiente paso consiste en definir las condiciones de contorno. Para esto COMSOL define las fronteras como *boundaries*. Cada *subdomain* está también caracterizado por sus respectivos *boundaries*. Las propiedades de las fronteras pueden ser de tipos diferentes para el módulo de transferencia de calor. Existe la posibilidad de definir fronteras que permitan el flujo de calor, otro tipo de fronteras que permiten convección, otro tipo de fronteras con una temperatura constante, y finalmente fronteras aisladas que impiden el paso de flujo calórico.

El único dominio que requiere de la definición de fronteras en el presente trabajo es el dominio C01 que representa el exterior de la mufla. La *Figura 5.9* muestra los bordes definidos en la geometría.

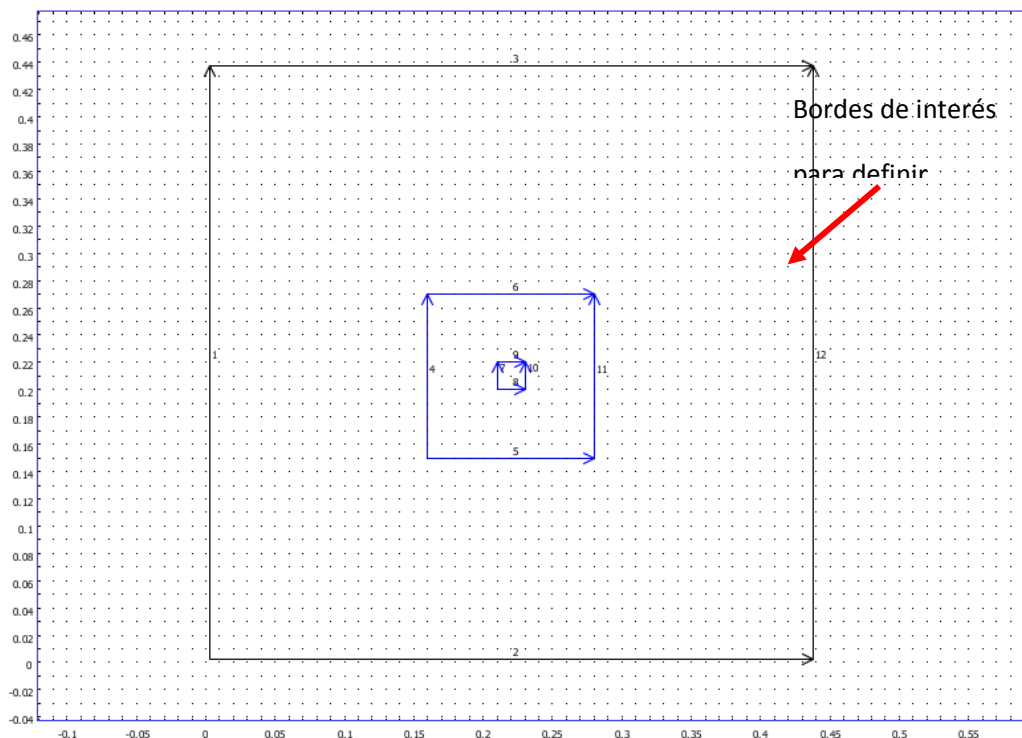


Figura 5.9 Bordes de la geometría indicados por las respectivas flechas.



Se observa que los bordes de las paredes internas de la mufla están en contacto con el dominio C02 de aire, que a su vez está en contacto con el dominio C03 de la probeta. Es por esta razón que no es necesario definir los bordes del interior de C01, los resultados de C02 y C03 definen las condiciones en estos bordes marcados con azul en la *Figura 5.9*.

La *Figura 5.10* muestra la definición de los bordes exteriores del dominio C01.

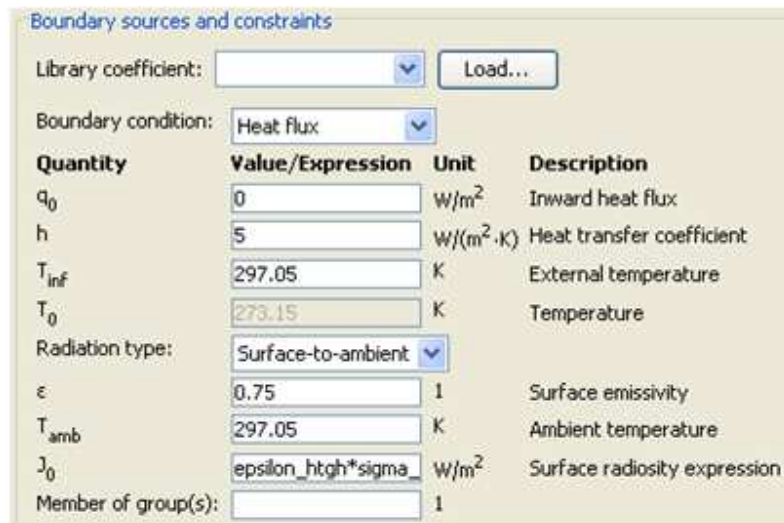


Figura 5.10 Condiciones de borde para subdomain C01 definidas en ambiente de COMSOL [ 10, 11 ].

(Contacto de ladrillo refractario con aire a temperatura ambiente)

Los datos que se observan indican que las paredes exteriores de la mufla están en contacto con aire a temperatura ambiente  $T_{inf}$  y un coeficiente de convección de 5 W/m<sup>2</sup>K. También se asume un grado de emisividad del material de 0.75 para tomar en cuenta pérdidas por radiación.

5) A continuación se puede discretizar el dominio y para ello se malla en elemento triangulares. Este tipo de mallado automático fue seleccionado debido a que con este método se puede afinar el mallado de manera que la solución sea lo más precisa posible. El mallado es útil para apreciar nodos que se generan en cada sólido, de esta manera en cada uno de estos nodos se va a encontrar el resultado de temperatura necesario para la simulación posterior. El mallado utilizado se puede ver en la *Figura*

5.11 en donde se aprecia los respectivos subdominios (ladrillo refractario, aire y acero) mallados.

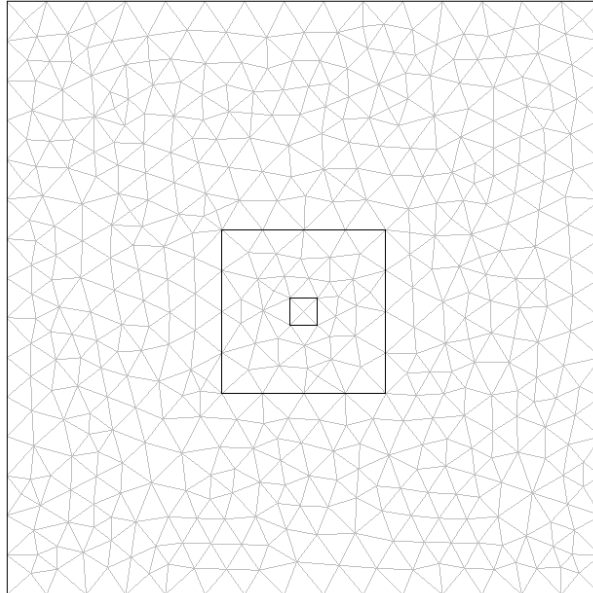


Figura 5.11 Mallado de la simulación de recocido para todas las geometrías en análisis.

- 6) Debido a que esta simulación es de estado transitorio, al resolverlo se puede generar una animación de cómo va variando la temperatura en el recocido a lo largo del tiempo. En el tratamiento térmico del recocido, el tiempo de enfriamiento es muy lento ya que la probeta de acero se la deja enfriar conjuntamente con el horno. Con esta simulación se demostró que el enfriamiento se completa luego de 24 horas desde que el horno fuera apagado. Por ello es necesario que en COMSOL se defina estos parámetros dentro del menú de “Solve Parameters.” De esta manera en el “time stepping” se define el intervalo de tiempo para que la probeta de acero se enfríe por completo. Esto se muestra a continuación en la *Figura 5.12*.

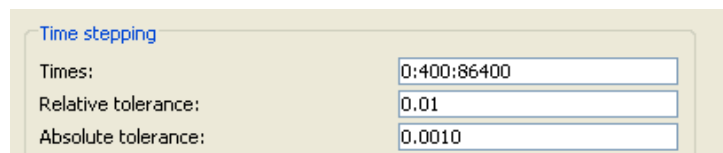


Figura 5.12 Definición de intervalo de tiempo para la simulación del enfriamiento en el recocido en ambiente de COMSOL

Como se puede ver, se ha determinado un intervalo de tiempo que va desde 0 hasta los 86400 segundos que vienen a ser 24 horas. Además al ser un intervalo muy largo se ha decidido que haya saltos de tiempo de 400 segundos para que la animación no sea tan lenta y se muestre todo el proceso de enfriamiento en un tiempo virtual más corto. Es decir, la animación no va a tardar 24 horas sino 216 segundos.

Una vez terminada la simulación hay como generar la animación mencionada anteriormente. Para ello es necesario entrar en el menú de “Plot Parameters” y asignar todos los datos necesarios para generar dicha animación. Entre ellos se necesita escoger el tipo de formato en que quiere que salga la animación, en este caso .avi. Por otro lado se necesita seleccionar todas las soluciones que se desea animar, en este caso transitorio serán los tiempos. Esto se puede observar en la parte seleccionada de azul en la *Figura 5.13*. En esta figura se puede ver todo lo necesario para generar la animación y adicionalmente se tiene un botón “Start Animation” que corre el programa y genera la animación deseada.

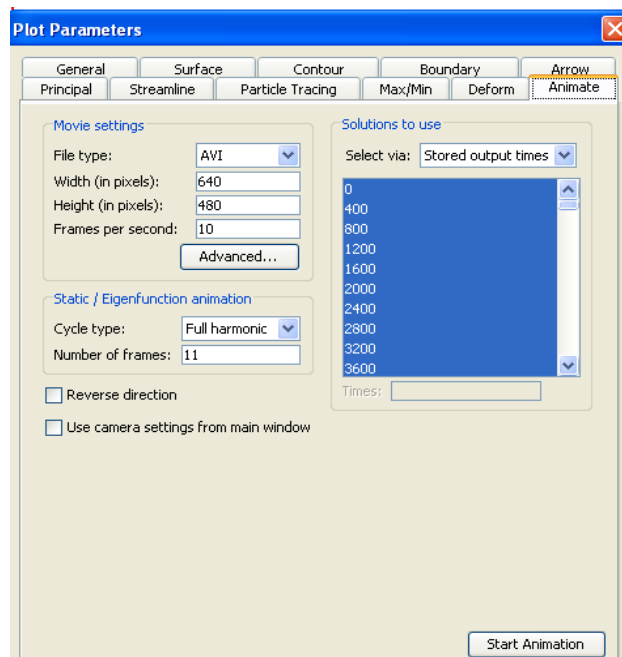


Figura 5.13 Generación de animación mediante intervalo de tiempo del recocido

7) Finalmente luego de realizar todo se obtienen los resultados como se muestra en la *Figura 5.14* en donde se muestra los ladrillos refractarios, el aire y la probeta de

acero. Además en esta figura se muestra la transferencia de calor (isotermas) y su correspondiente escala de colores. Al lado derecho se encuentra una barra de colores los cuales corresponden a temperaturas diferentes. El color azul representa una temperatura muy cercana a la ambiente mientras que el color rojo representa una temperatura elevada (temperatura de austenización). Todos estos cambios de temperaturas (cambio de color de rojo a azul) se puede apreciar en la animación adjunta en un CD a esta tesis. Con este resultado lo que se obtiene es la temperatura en cada nodo seleccionado con sus coordenadas respectivas a cada instante de tiempo. Para el caso de esta simulación lo que interesa son las temperaturas de la probeta de acero en cada instante de tiempo. De esta manera con las temperaturas dadas se podrá ingresar en el programa de Monte Carlo Potts para generar la simulación de evolución microestructural de grano.

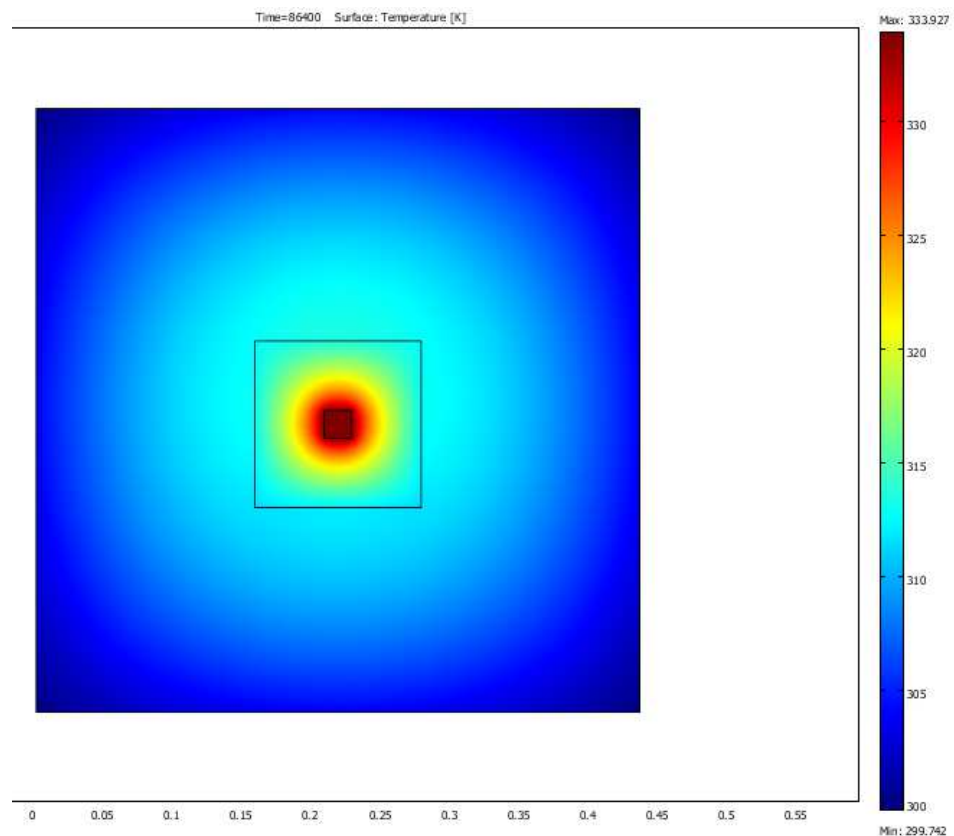


Figura 5.14 Solución final de temperaturas en grados Kelvin obtenida con COMSOL Multiphysics

## 5.4 Resultados obtenidos con temperaturas y movilidad

Para obtener las matrices de temperaturas es necesario exportar a un archivo *.txt* o también se las puede exportar a un script que entienda MATLAB que son los archivos *.m*. Esta es la gran ventaja de COMSOL que el programador se ahorra mucho tiempo de estar ingresado dato por dato las temperaturas. Lo único que se hace es juntar a todas las temperaturas en cada intervalo de tiempo dentro de una matriz grande a la que se llama *T* dentro de un archivo llamado *temperaturas.m*. En esta matriz de temperaturas solo se tomo en cuenta el área de la probeta debido a que ese es el área de interés para la simulación. El código realizado en MATLAB para obtener cada matriz de temperatura se muestra a continuación en la *Figura 5.15*.

```
for i=1:31
    T1=T(:, :, i);
```

Figura 5.15 Código para llamar a cada matriz de temperatura en cada instante de tiempo

Como se puede apreciar en la figura mencionada, se llama a cada matriz dependiendo del índice *i* que se le asigno previamente. *T* es una matriz de tres dimensiones en donde *i* indica el tiempo o intervalos de tiempo de solución. Entonces dependiendo de cada visualización se llamara a diferentes matrices de temperaturas que son asignadas a cada elemento de la matriz simulada. A lo largo del tiempo estas temperaturas irán decreciendo hasta llegar a un punto de equilibrio con la temperatura ambiente. Para distintos tratamientos el comportamiento de las matrices de temperatura será diferente.

**Recocido:**

<b>Matriz 4</b>	100x100
<b>Lattice:</b>	Cuadrada
<b>Orientaciones Cristalográficas:</b>	30
<b>MCS totales:</b>	1240
<b>Movilidad</b>	$u=T./1150$

Tabla 5.2 Datos para simulación de una matriz 100x100 con 30 orientaciones cristalográficas con temperaturas *T* y movilidad *u*

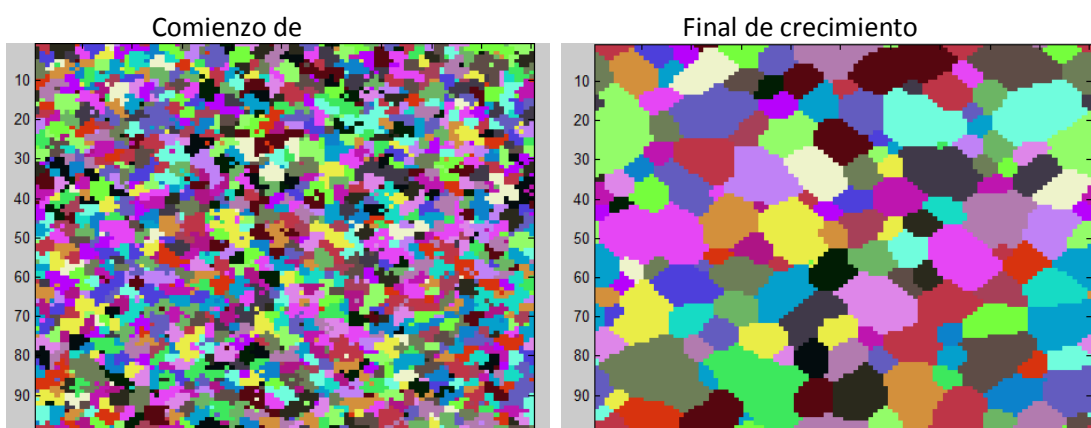


Figura 5.16 Simulación con movilidad alta por un lapso de tiempo y luego lentamente va bajando.

El tamaño de grano es mucho más grande

Como se puede observar en la *Figura 5.16* con temperaturas y movilidad la simulación se asemeja más a la realidad. En este caso el proceso de crecimiento de grano es de un recocido y se explica por dos factores: la movilidad y las temperaturas sacadas de la simulación en COMSOL. En este caso la movilidad varía con respecto a la temperatura de la siguiente manera:  $\mu = T. / 1150$ . De esta manera el grano ira creciendo rápidamente al comienzo pero mientras la probeta se va enfriando el tamaño de grano va aumentando lentamente y en algunos casos dejan de crecer. Por esto, la simulación tiene las características de un recocido. El crecimiento de grano se puede apreciar en la *Figura 5.17*.

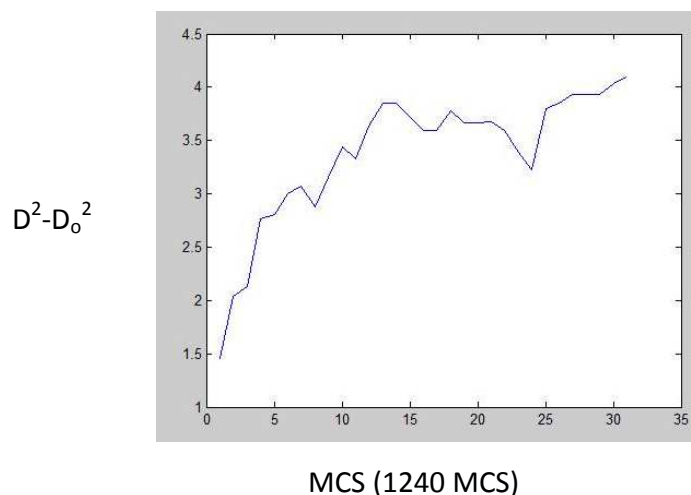


Figura 5.17 Tamaño de grano de la simulación

**Enfriamiento moderado:**

<b>Matriz 5</b>	100x100
-----------------	---------

<b>Lattice:</b>	Cuadrada
<b>Orientaciones Cristalográficas:</b>	30
<b>MCS totales:</b>	1240
<b>Movilidad</b>	$u = (T-200)/1150$

Tabla 5.3 Datos para simulación de una matriz 100x100 con 30 orientaciones cristalográficas con temperaturas  $T$  y movilidad  $u$

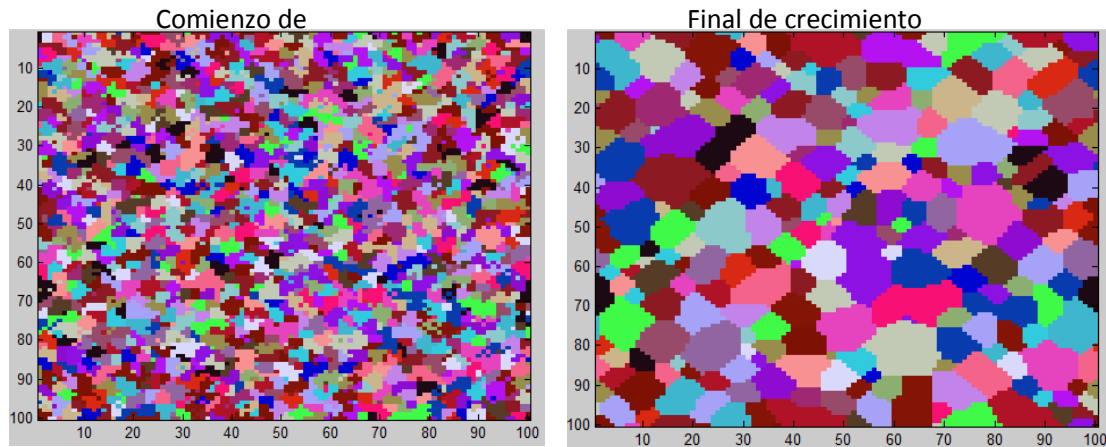


Figura 5.18 Simulación con movilidad alta por menos tiempo y baja mucho más rápido. El tamaño de grano es menor.

En este caso la simulación se la considera como un enfriamiento moderado debido a que el tamaño de grano aumenta lentamente. Esto se debe a que la movilidad se la ha obligado a disminuir en cantidades mayores a lo largo del tiempo. Por ello, menos tiempo van a evolucionar los granos debido a su movilidad baja y temperaturas que disminuyen más que en el recocido. Se puede apreciar este fenómeno en la *Figura 5.19* en donde si comparamos el tamaño con el del recocido en la *Figura 5.17* el normalizado tiene un tamaño de grano menor en el mismo tiempo virtual generado.

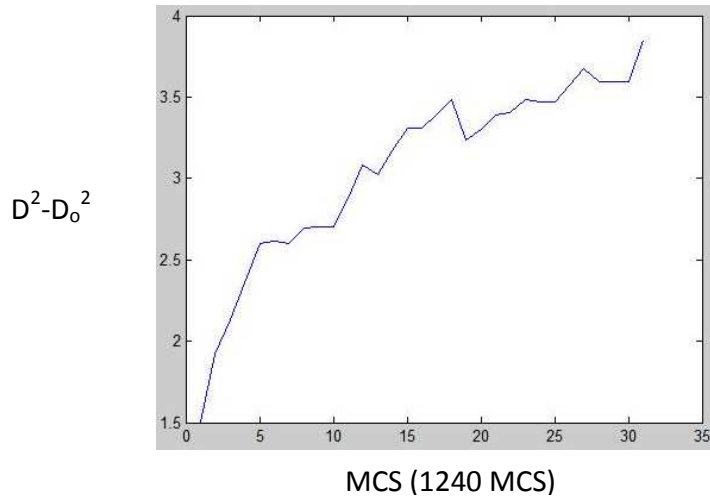


Figura 5.19 Tamaño de grano de la simulación

**Enfriamiento rápido:**

<b>Matriz 6</b>	100x100
<b>Lattice:</b>	Cuadrada
<b>Orientaciones Cristalográficas:</b>	30
<b>MCS totales:</b>	1240
<b>Movilidad</b>	$u=(T-400)/1150$

Tabla 5.4 Datos para simulación de una matriz 100x100 con 30 orientaciones cristalográficas con temperaturas  $T$  y movilidad  $u$

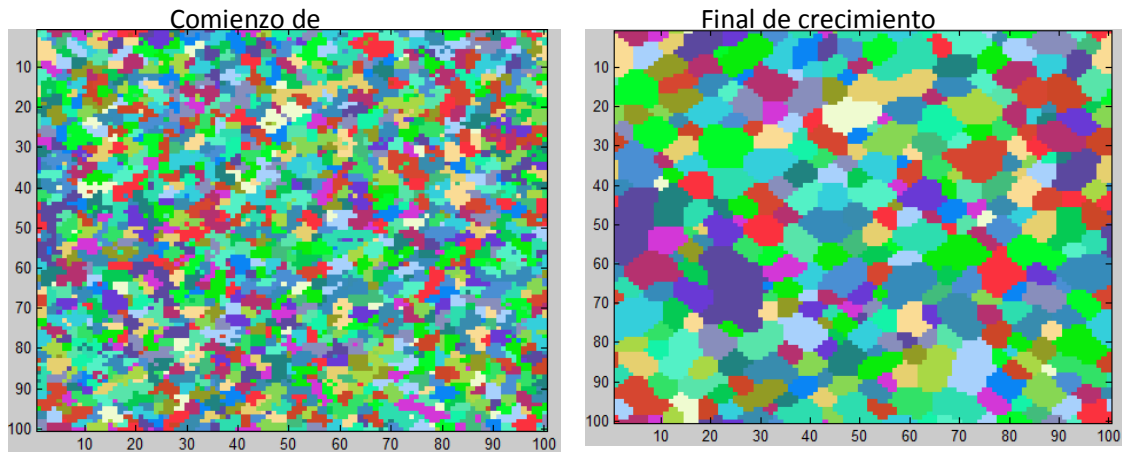


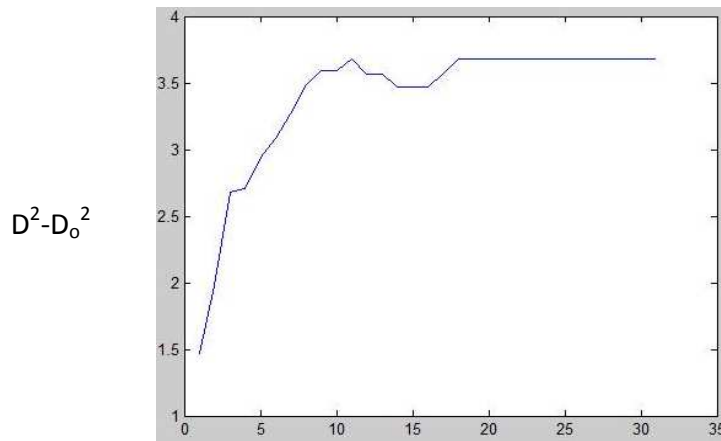
Figura 5.20 Simulación con movilidad que baja rápidamente.

El tamaño de grano es mucho menor.

Finalmente se simuló un enfriamiento rápido en donde el crecimiento de grano fue menor debido a que la movilidad casi siempre se mantuvo bajo. Como se puede observar



en la *Figura 5.20* los granos no aumentaron mucho de tamaño y esto se lo puede apreciar mejor en la *Figura 5.21*. En esta figura se observa que en un comienzo el tamaño de grano aumento pero ya a partir de cierto tiempo el tamaño se mantuvo en promedio constante lo que nos indica que esto es una simulación característica de un enfriamiento rápido.



MCS (1240 MCS)

Figura 5.21 Tamaño de grano de la simulación

### 5.5 Simulando efectos de gradiente de temperatura:

Puede desearse simular efectos de variaciones de temperatura en microestructuras. Es decir, aplicar sobre la matriz una gradiente de temperatura para observar su efecto en el tamaño de grano. Esto se logra manejando la movilidad en función de la temperatura. La tabla muestra los parámetros alimentados al programa. Los resultados se observan en la *Figura 5.22*.

Tamaño de matriz N*N	100x100
MCS	500

Temperatura	$T(x)=x * 100$ (Temperatura aumenta de izquierda a derecha en la matriz)
Movilidad $u$	En función de Temperatura
Q	10

Tabla 5.5 Parámetros para simulación con variación de temperatura.

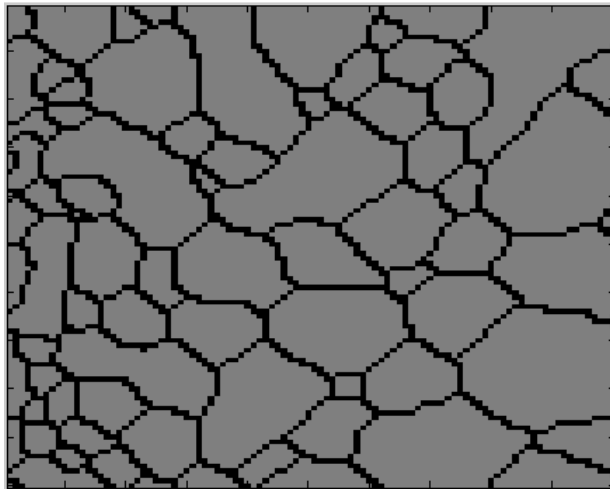


Figura 5.22 Tamaño de grano aumenta de izquierda a derecha.

Para apreciar de mejor manera la diferencia de tamaños, se midió el tamaño promedio de grano a lo largo de la primera columna de la matriz y a lo largo de la última columna. Los resultados de la función `tamaniogranomovilidad.m` según lo mostrado por MATLAB se encuentran en la *Figura 5.23*.

```
tamano1columna =
    7.1434
tamano2columna =
   14.6981
```

Figura 5.23 `tamano1columna` es el tamaño de grano a la izquierda de la matriz. `tamano2columna` es el tamaño de grano a la derecha de la matriz.

## 5.6 Simulando un material con dos componentes o fases.

Como se mencionó en la sección 3.7, es posible introducir componentes a una simulación. El generador de microestructuras puede recibir diferentes parámetros para conseguir los efectos deseados. En las figuras a continuación se presentan los resultados del generador de microestructuras utilizando los parámetros de la tabla 5.6. Existen dos componentes: un componente coloreado de blanco, y otro componente coloreado de gris.

% Componente Blanco	50%
% Componente Gris	50%
Movilidad u	1
Q	30
N	150
MCS	300

Tabla 5.6 Parámetros para simulación con variación de temperatura.

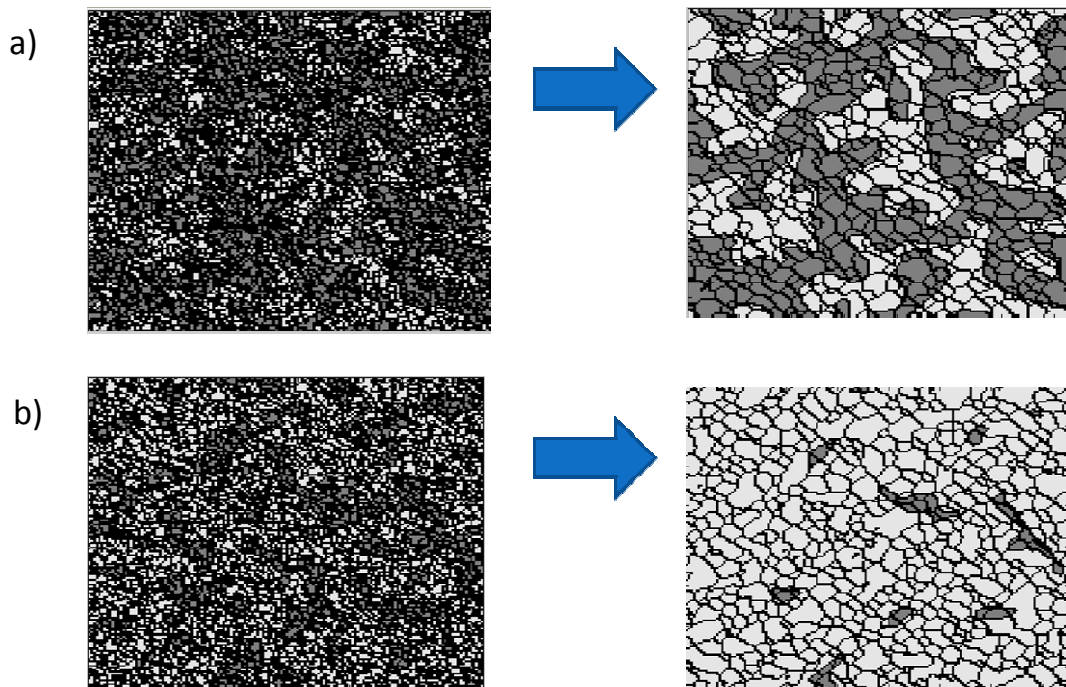


Figura 5.24 Las energías entre componentes no son consideradas en a) y la concentración de los componentes es la misma al inicio y al final. La minimización de la energía se logra favorablemente con el componente blanco y en b) la concentración del componente blanco aumenta a medida que el componente gris disminuye.

En un primer paso se considera que las energías de estos componentes no tienen efecto en el intercambio de energías debido a la diferencia en redes cristalográficas. El segundo ejemplo introduce energías a los componentes ocasionando que uno de ellos consuma al segundo componente como se observa en la figura 5.24. Una posible aplicación de este efecto es para señalar los resultados de un tratamiento térmico en un acero. A una temperatura señalada y luego de que alcanzar cierto tiempo, la fase austenítica cambia a una fase perlítica. En este caso, la fase perlítica minimiza la energía dependiendo del tiempo y la temperatura, como podría señalarse por medio de la fase blanca en la *Figura 5.24 b*).

# 6

## **Conclusiones:**

Finalmente podemos obtener algunas conclusiones finales sobre el método luego de haberlo utilizado ampliamente a través de toda la investigación.

El método tiene la ventaja de que puede simular fácilmente los fenómenos de crecimiento de grano. Es posible incluir efectos de temperatura, efectos de diversos componentes sobre la energía, efectos de direccionalidad en el crecimiento de grano, sin gran variación del código. Estos efectos se consiguen manejando los conceptos de movilidad y de energía de los bordes de grano.

Entre las desventajas del método está el gran número de procesos que requiere para realizar las operaciones necesarias. Es preciso recordar que el algoritmo evalúa la matriz una cantidad de veces equivalente a  $N*N$ . Aún más, introducir efectos complejos, que permiten a la simulación acercarse a la realidad, exigen definir mayor cantidad de matrices incrementando el número de procesos necesarios. Existen algoritmos más eficientes, pero si se desea obtener resultados precisos no hay forma de evitar el gran número de procesos que deben realizarse.

El poder del simulador de microestructuras es que permite tener un control total sobre los elementos que forman cualquier microestructura generada. Es posible identificar cualquier grano dentro de la matriz y asignarle características propias. Es posible contar el número de elementos que se encuentran dentro de un grano. Y, finalmente, es sencillo trasladar los resultados hacia otro paquete computacional para mayor análisis. Obsérvese que los resultados están simplemente señalados en una matriz cuadrada. Un siguiente trabajo podría llevar los resultados hacia una aplicación que utilice el método de elementos finitos para analizar la muestra aplicando desplazamientos, esfuerzos, presiones entre otros.

## **REFERENCIAS**

- [ 1 ] Acharjee, Swagato. "Grain growth simulation using Monte-Carlo methods."  
December 12 2002.
  
- [ 2 ] Blikstein, Paulo, Tschiptschin André, "Monte Carlo Simulation of Grain Growth."

Escola Politécnica da Universidade de São Paulo. Marzo 30 1999.

- [ 3 ] Çengel, Yunus A. Thermodynamics an engineering approach. Boston: McGraw-Hill Higher Education, 2006.
  
- [ 4 ] DelCarpio <<http://www.delcarpio.cl/muflas.html>>
  
- [ 5 ] FISICA NET. <[http://www.fisicanet.com.ar/fisica/termodinamica/tb03\\_conductividad.php](http://www.fisicanet.com.ar/fisica/termodinamica/tb03_conductividad.php)>
  
- [ 6 ] Holm, Elizabeth. Battaile, Corbett. “The Computer Simulation of Microstructural Evolution.” <<http://www.tms.org/pubs/journals/JOM/0109/Holm-0109.html>>
  
- [ 7 ] Janssens, Koenraad George Frans, Dierk Raabe, Ernest Kozeschnik, Mark A. Miodownik, and Britta Nestler. Computational Materials Engineering An Introduction to Microstructure Evolution. New York: Academic P, 2007.
  
- [ 8 ] Keyser, Carl. Ciencia de materiales para Ingenieria. Mexico: Limusa Wiley, 1972
  
- [ 9 ] Li, M.-Y. Asibu, Kanna Tey. “Monte Carlo Simulation of Heat-Affected Zone Microstructure in Laser-Beam Welded Nickel Sheet.” Welding Journal. March 2002.
  
- [ 10 ] Tabla emisividad. IMPIC termografía infraroja. <[http://www.impictermografia.com/noticias\\_termografia/index.php/emisividad/tablas-de-emisividad/9-tablas-emisividad/5-tabla-emisividad](http://www.impictermografia.com/noticias_termografia/index.php/emisividad/tablas-de-emisividad/9-tablas-emisividad/5-tabla-emisividad)>
  
- [ 11 ] Transferencia por convección.

<[www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r14178.DOC](http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r14178.DOC)>

[ 12 ] Wikipedia, "algoritmo," "tratamientos térmicos." <[es.wikipedia.org/](http://es.wikipedia.org/)>

[ 13 ] Typical Convection Coefficients to Estimate Thermal Transfer  
<<http://www.me.mtu.edu/~microweb/GRAPH/Intro/film.htm>>

[ 14 ] Wittwer, J.W., "Monte Carlo Simulation Basics" de *Vertex42.com*, June 1, 2004,  
<<http://vertex42.com/ExcelArticles/mc/MonteCarloSimulation.html>>

[ 15 ] Woller, Joy. "The Basics of Monte Carlo Simulations." University of Nebraska-  
Lincoln. 1996. <<http://www.chem.unl.edu/zeng/joy/mclab/mcintro.html>>