

**SIMULACIÓN DE IMPLANTACIÓN DE NITROGENO EN HIERRO  
POLICRISTALINO POR EL MÉTODO DE DINÁMICA MOLECULAR**

**EDMANUEL EDUARDO TORRES AMARIS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE CIENCIAS  
ESCUELA DE FÍSICA  
MAESTRÍA EN FÍSICA  
BUCARAMANGA  
2005**

**SIMULACIÓN DE IMPLANTACIÓN DE NITROGENO EN HIERRO  
POLICRISTALINO POR EL MÉTODO DE DINÁMICA MOLECULAR**

**EDMANUEL EDUARDO TORRES AMARIS**

**Trabajo de investigación para optar al título de Magíster en Física**

**Director  
VALERI DOUGAR-JABON.  
Físico, PhD**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE CIENCIAS  
ESCUELA DE FÍSICA  
MAESTRÍA EN FÍSICA  
BUCARAMANGA  
2005**

*“Su voz tiene un timbre celestial,  
lástima que sus palabras no signifiquen nada,  
lástima que no sirvan para nada<sup>1</sup>”.* Edmanuel Torres

---

<sup>1</sup>extraído de un texto escrito por, Edgar Allan Poe

## DEDICATORIA

Dedicada con mucho cariño a todas las personas que me han apoyado durante todo este tiempo. En especial a mi madre *Olga Amaris Luqueta* y a mi padre *Rafael Torres Martínéz*.

## AGRADECIMIENTOS

Quisiera agradecerle al **Dr. Valeri Dougar-Jabon** de quien he aprendido mucho a lo largo de todos estos años, y quien con su continua dirección ha hecho realidad la elaboración de este trabajo.

También quiero expresar mi gratitud a muchas personas con quien he compartido muchas discusiones y estudios durante todo este tiempo.

También estoy fuertemente agradecido con la comunidad de software libre **GNU**, y en general con toda la gente de **Linux**. Todo este trabajo se desarrollo bajo este sistema.

# Índice general

<b>1. Introducción</b>	<b>14</b>
<b>2. Fundamentos Básicos</b>	<b>16</b>
2.1. Implantación Iónica . . . . .	16
2.1.1. Implantación iónica por Haces Plasmicos . . . . .	17
2.1.2. Implantación Iónica por Inmersión en Plasma . . . . .	18
2.1.3. Implantación Iónica 3-Dimensional . . . . .	18
2.2. Introducción a la Estructura de los Sólidos . . . . .	19
2.2.1. Materiales Cristalinos . . . . .	19
2.2.2. Materiales Policristalinos . . . . .	20
2.3. Simulación de Materiales . . . . .	21
<b>3. Implantación de Iones</b>	<b>22</b>
3.1. Física de la Implantación de Iones . . . . .	22
3.2. Frenado de iones en Sólidos . . . . .	23
3.2.1. Frenado Electrónico . . . . .	25
3.2.2. Frenado Nuclear . . . . .	28
3.3. Física de las colisiones en cascada . . . . .	29
3.4. Canaleo de Iones . . . . .	30
3.5. Rango de Implantación . . . . .	30
<b>4. Aproximación Clásica</b>	<b>32</b>
4.1. Aproximación de Born-Oppenheimer . . . . .	32
4.2. Ecuaciones de Newton-Lorentz y Potenciales Interatómicos . . . . .	34

4.2.1.	Potencial en metales . . . . .	35
4.2.2.	Potencial de interacción ion-átomo . . . . .	36
<b>5.</b>	<b>Modelamiento Computacional</b>	<b>38</b>
5.1.	Ideas básicas del Modelamiento . . . . .	39
5.2.	Dinámica Molecular (DM) . . . . .	39
5.2.1.	Cómputo de las Fuerzas . . . . .	41
5.2.2.	Unidades adimensionales . . . . .	44
5.2.3.	Algoritmos de Integración en el tiempo . . . . .	46
5.2.4.	Condiciones de Frontera Periódica . . . . .	51
5.3.	Métodos de optimización . . . . .	52
5.3.1.	Método Lista-Celda . . . . .	53
5.3.2.	Método Lista-Capa . . . . .	54
5.3.3.	Paso de Tiempo Variable . . . . .	56
<b>6.</b>	<b>Simulación de Implantación de Iones</b>	<b>58</b>
6.1.	Algoritmo para Implantación de Iones . . . . .	59
6.2.	Parámetros Físicos y Computacionales . . . . .	61
<b>7.</b>	<b>Resultados</b>	<b>63</b>
<b>8.</b>	<b>Conclusiones</b>	<b>71</b>
	<b>Bibliografía</b>	<b>84</b>

# Índice de figuras

2.1. Esquema bidimensional de un material cristalino, cada círculo represente a un átomo del material . . . . .	19
2.2. Esquema granular bidimensional de un material policristalino, dode cada círculo represente a un átomo del material . . . . .	20
3.1. Ilustración de la trayectoria de un ion en un material sólido. . . . .	23
3.2. Una comparación entre el frenado electrónico y nuclear . . . . .	25
5.1. Interacción de todas las partículas, en este esquema se pueden apreciar la diferencia del número de partículas para dos radios de acción. . . . .	42
5.2. Esquema bidimensional de Condiciones de Fronteras Periódicas . . . . .	51
5.3. Condiciones de fronteras periódicas para un sistema de lado $l$ . . . . .	52
5.4. Esquema del método lista-celda para la búsqueda de los átomos vecinos. . . . .	53
5.5. Esquema del método lista-capa para la búsqueda de los átomos vecinos. . . . .	55
5.6. Esquema completo del método de optimización. . . . .	56
6.1. Esquema de semillas y átomos libres. . . . .	59
6.2. Policristal formado por nucleación. . . . .	60
6.3. Esquema bidimensional del estado inicial de una simulación en implantación. . . . .	61
7.1. Gráfica de la trayectoria de 100 iones implantadas en bcc-Fe monocristalino con energía inicial de $10keV$ . . . . .	64
7.2. Gráfica de la trayectoria de 100 partículas implantadas en bcc-Fe policristalino con energía inicial de $10keV$ . . . . .	64
7.3. Perfil de implantación para partículas con energía inicial de $10keV$ en bcc-Fe monocristalino. . . . .	65

7.4. Perfil de implantación para partículas con energía inicial de $20keV$ en bcc-Fe monocristalino. . . . .	65
7.5. Perfil de implantación para partículas con energía inicial de $10keV$ en bcc-Hierro policristalino. . . . .	66
7.6. Perfil de implantación para iones de Nitrógeno con energía inicial de $20keV$ en bcc-Hierro policristalino. . . . .	66
7.7. Perfil de implantación para iones de Nitrógeno con energía inicial de $30keV$ en bcc-Hierro policristalino. . . . .	67
7.8. Superposición de los perfiles de implantación en bcc-Fe policristalino a 10, 20 y $30keV$ . . . . .	67
7.9. Gráfica de los rangos proyectados en función de la energía inicial de implantación para bcc-Fe policristalino. . . . .	68
7.10. Perfiles SIMS de implantación para iones de Nitrógeno con energías de $10keV$ y $20keV$ en Acero. . . . .	69
7.11. Superposición de los perfiles de implantación experimentales con los calculados a 10 y $20keV$ . . . . .	69

# Índice de cuadros

4.1. Coeficientes de la formula (4.10) . . . . .	36
5.1. Coeficientes de la formula (5.8) . . . . .	43
6.1. Parámetros empleados durante las simulaciones. . . . .	62
7.1. Rangos medios proyectados para las diferentes energías. . . . .	68

# TÍTULO: SIMULACIÓN DE IMPLANTACIÓN DE NITROGENO EN HIERRO POLICRISTALINO POR EL MÉTODO DE DINÁMICA MOLECULAR

Edmanuel Eduardo Torres Amaris <sup>2</sup>

## Palabras Claves

1. Implantación de iones. 2. Dinámica Molecular. 3. Tratamiento superficial.

## Resumen

La distribución final de iones implantados es una característica muy importante en el tratamiento superficial de materiales. Cuando iones con igual energía entran en un material y atraviesan algunas capas atómicas, son dispersados por los electrones y los núcleos de los átomos del metal; de manera que después de algún tiempo tendrán esencialmente distintas energías y trayectorias. Debido a la imposibilidad que hay en determinar la trayectoria de estos iones ya sea por métodos analíticos o experimentales, los experimentos numéricos han dado la única posibilidad de obtener esta información. En simulación de implantación de iones hay un buen número de técnicas que pueden ser empleadas, tales como: el método Monte Carlo, Partícula-Malla, Dinámica Molecular entre otros. En este trabajo, se optó por la técnica de Dinámica Molecular para simular el movimiento de los iones implantados para estudiar el proceso de la penetración de los iones en el material. Dentro de los resultados, se encontró la dependencia de la distribución con la estructura del blanco, la distribución de iones en un policristal de Hierro difiere de la distribución obtenida para el caso de un monocristal de Hierro. Se encontró que el patrón de distribución obtenido para un blanco de Hierro policristalino está de acorde con los perfiles de iones implantados experimentales obtenidos en muestra de Acero SAE1020 en el dispositivo de implantación JUPITER.

---

<sup>2</sup>Trabajo de investigación. Facultad de Ciencias. Maestría en Física. DOUGAR-JABON, Valeri.

**TITLE: COMPUTING OF NITROGEN ION IMPANTANTION  
INTO POLYCRYSTALLINE BCC-IRON: BY MOLECULAR  
DYNAMIC SIMULATION**

**Edmanuel Eduardo Torres Amaris <sup>3</sup>**

**Keywords**

1. Ion Implantation. 2. Molecular Dynamics. 3. Superficial Treatment.

**Abstract**

The final distribution of the implanted ions is an important characteristic of the superficial treatment of metals. Ions with equal initial energy are dispersed by the electrons and by the atoms' nuclei in metal materials after passing through a few atomic layers; so they will have essentially distinct energy and trajectories after a time. Due to impossibility to determine the ion trajectories and energy losses by any analytic method and by experimentally, the numerical experiment presents the only possibility for obtaining this information. There are several techniques for simulation of the implantation: the Monte Carlo method, Particle-in-Cell, Molecular Dynamics and others. In this work, the Molecular Dynamics technique is chosen for simulating of the movement of implanting ions that permits understand better the process of penetration of ions into metals. It is shown that the distribution of Nitrogen ions implanted into polycrystalline iron is very different of the distribution obtained for monocrystalline iron. It is demonstrated that the pattern of distribution of Nitrogen ions implanted for polycrystal case is well coincided with the experimental data obtained for the SAE1020 Steel on the device JUPITER.

---

<sup>3</sup>Research project. Faculty of Science. Master in Physics. DOUGAR-JABON, Valeri.

# Capítulo 1

## Introducción

Como nuestra reciente historia ha mostrado, los avances en tecnología dependen de los avances en el desempeño y el desarrollo de nuevos materiales de superiores propiedades. Normalmente estos avances han sido realizados a través de un largo proceso experimental, empleando extendidas pruebas de laboratorio sobre las muchas variantes de materiales existentes y desarrollados, recogiendo y ordenando numerosos datos obtenidos a través de la experiencia práctica. Este proceso no solo es lento para el desarrollo de materiales, sino que también está fuertemente restringido en la investigación de materiales poco estudiados<sup>1</sup>. El campo de los nuevos materiales es muy amplio y exige ser explorado de una manera sistemática y exhaustiva. Hay una vasta cantidad de materiales para ser descubiertos con potenciales aplicaciones y propiedades excepcionales, sin embargo, los métodos tradicionales no son lo suficientemente eficientes para su rápido desarrollo.

No obstante, las propiedades de los materiales pueden en última instancia ser explicadas mediante las propiedades de sus átomos y sus interacciones mutuas. Por consiguiente, las simulaciones actuales buscan describir la materia a partir de los átomos (o moléculas) que la conforman, a esta escala, regida por electrones y núcleos atómicos se le conoce como escala atómica, de aquí que podamos esperar, que el problema

---

<sup>1</sup>Aún cuando hay una buena cantidad de materiales que han sido constantemente afinados y tabulados para aplicaciones específicas, como los Aceros austeníticos y sus variantes

fundamental de la física de los materiales y su interacción con partículas pertenezca al dominio del problema de muchos cuerpos. En esta dirección, los cálculos atomísticos han alcanzado el nivel de predicción de propiedades de los materiales como: la energía de enlace, estructura cristalina, propiedades ópticas entre otras.

En esta tesis estudiamos la implantación iónica mediante simulaciones por el método de Dinámica Molecular (DM) desarrollado en siete capítulos. En los cuatro primeros capítulos se desarrollan los conceptos fundamentales de los sistemas atómicos y las consideraciones necesarias que se llevan a cabo para explicar el paso de un modelo físico de átomos interactuando a la aproximación clásica. En los capítulos quinto y sexto, se define lo que podríamos encontrar en el contexto de una simulación de DM y se describe el esquema de las simulaciones enfocadas específicamente a las simulaciones en implantación de iones, pero que pueden ser extrapoladas a otros sistemas de interés. En el capítulo séptimo se describen los resultados obtenidos mediante las simulaciones desarrolladas durante este trabajo y finalmente en el capítulo octavo se plantean las conclusiones del trabajo.

## Capítulo 2

# Fundamentos Básicos

La implantación iónica fue al principio un método para el dopaje de semiconductores en la fabricación de Circuitos Integrados, debido a la rapidéz, precisión y control del proceso esta técnica se convirtió en el estándar para este tipo de aplicaciones en la industria de dispositivos electrónicos. No fue sino hasta los años 70, cuando se encontró que la implantación iónica en ciertas condiciones podría mejorar algunas propiedades superficiales en metales, tales como: desgaste, fricción y corrosión entre otras. En la actualidad, se han encontrado muchas aplicaciones para este tipo de procesos en el tratamiento de superficies, y debido a los cambios físicos y químicos que implica, se pueden variar algunas de sus propiedades en un aspecto deseado. Hoy en día, la implantación es ampliamente usada en el tratamiento avanzado de superficies de modernos equipos (e.g., en la industria médica, aeroespacial, automovilística, entre otras).

### 2.1. Implantación Iónica

La implantación iónica es la interacción de iones con la superficie de materiales sólidos. Hay diferentes técnicas para llevar a cabo este proceso, pero la idea básica consiste en que algunos iones sean extraídos de una fuente y dirigidos hacia el blanco con energía suficiente para que pasen a través de la superficie y puedan moverse dentro de una zona cercana a esta, experimentando una gran cantidad de fenómenos durante

su trayectoria.

En implantación la mayoría de los fenómenos son en todo momento la manifestación de la interacción de muchas partículas, por lo tanto no pueden ser tratados como un caso simple de dispersión clásica entre dos partículas cargadas[1]. No hay un desarrollo teórico completo que permita explicar los diferentes fenómenos que se producen durante la implantación<sup>1</sup>. Otro problema, radica en la dificultad de obtener información concerniente a la dinámica de los iones -debido a que se encuentra en una región dentro del material- durante su movimiento. Debido a las razones mencionadas anteriormente, existe una brecha entre los estudios analíticos e investigaciones experimentales para este tipo de procesos, y solo en algunos casos particulares un número reducido de métodos teóricos han tenido aplicaciones prácticas[2].

En general hay dos métodos ampliamente conocidos de realizar implantación iónica, estos son: la Implantación de Iones por Haz (BII)<sup>2</sup> y la Implantación de Iones vía Plasma (PII)<sup>3</sup>[3]. En la práctica solo un par de técnicas son ampliamente implementadas, a saber: la Implantación Iónica por Haces Plasmicos o (PBII)<sup>4</sup> la cual es del tipo BII y la Implantación Iónica por Inmersión en Plasma o (PIII)<sup>5</sup> la cual es del tipo PII.

### 2.1.1. Implantación iónica por Haces Plasmicos

La técnica PBII ha sido el método convencional de implantación, y consiste en la generación de un plasma dentro de una cámara en condiciones de vacío, de la cual se extrae un flujo de iones para luego ser acelerado y enfocado hacia un blanco. El haz debe ser parcialmente enmascarado para prevenir que algunos iones incidan en ángulos mayores a 30 grados relativo a la dirección normal de la superficie; esto con la intención de reducir el chisporroteo. Por otra parte, en estos sistemas una muestra de forma compleja debe ser manipulada para asegurar que sea implantada sobre toda la

---

<sup>1</sup>por ejemplo, para el simple problema de una colisión entre un protón y un átomo de Hidrógeno no hay una solución teórica exacta

<sup>2</sup>por su nombre en Inglés *Beam Ion Implantation*

<sup>3</sup>por *Plasma Ion Implantation*

<sup>4</sup>por *Plasma Beam Ion Implantation*

<sup>5</sup>por *Plasma Immersion Ion Implantation*

superficie.

### 2.1.2. Implantación Iónica por Inmersión en Plasma

En la tecnología PIII, el blanco es ubicado en condiciones de vacío dentro de una cámara donde un plasma ha sido previamente generado. Luego mediante la aplicación de un pulso negativo de alto voltaje, los iones son acelerados hacia el material de modo que incidan sobre la superficie en dirección casi-normal; la dirección normal es la óptima para la implantación. Este método fue desarrollado por Conrad[4] para metales, y en este contexto se le llamó Implantación Iónica por Fuente Plásmica o PSII<sup>6</sup>.

### 2.1.3. Implantación Iónica 3-Dimensional

En años recientes fue reportado un nuevo dispositivo de implantación por V. Khevsyuk y P. Tsygankov[5], donde una descarga de alto voltaje encendida en la rama izquierda de la curva de Paschen<sup>7</sup> juega el papel principal, en otras palabras, la descarga es mantenida a baja presión. Ésta se diferencia de la técnica descrita anteriormente, en que no hay necesidad de la generación previa de un plasma dentro de la cámara. Este método se denomina Implantación Iónica 3-Dimensional o 3DII<sup>8</sup>, y entre algunas de sus características principales están la alta mono-energicidad del flujo de iones y su incidencia normal sobre toda la superficie de un objeto, aún para blancos de geometría compleja. El único equipo que actualmente funciona con base a la técnica 3DII[6] es el equipo JUPITER<sup>9</sup>.

---

<sup>6</sup>por *Plasma Source Ion Implantation*

<sup>7</sup>dependencia del voltaje de ruptura con el producto de la presión y la distancia interelectródica

<sup>8</sup>por *3-Dimensional Ion Implantation*

<sup>9</sup>Joint Universal Plasma and Ion Technology Experimental ReactorLaboratorio Física y Tecnología del Plasma , (UIS)

## 2.2. Introducción a la Estructura de los Sólidos

Generalmente cuando las sustancias líquidas sufren un proceso de solidificación forman redes cristalinas. Bajo algunas condiciones el resultado podría ser el de un único cristal donde todos los átomos están fijos dentro de una misma red o estructura cristalina, formando lo que se conoce como monocristal. En general esto no ocurre de manera natural, así que en lugar de esto, pueden ser muchas las redes que se generan simultáneamente durante la solidificación para porciones macroscópicas de materia en diferentes lugares del volumen del sólido, formando lo que se conoce como sólido policristalino, y en el caso más extremo podría resultar siendo un sólido completamente amorfo. En general, los materiales sólidos se encuentran en su gran mayoría en estado policristalinos, hallándose naturalmente en un punto intermedio entre el estado cristalino y el amorfo.

### 2.2.1. Materiales Cristalinos

Un cristal es un material sólido constituido por átomos, moléculas, o iones empacados en un estructura regular caracterizada por un arreglo periódico de sus partículas, que se extiende en todas las dimensiones espaciales (ver figura 2.1). Este ordenamiento siempre genera una dependencia de las propiedades del material en función de la dirección, a este fenómeno se conoce como anisotropía.

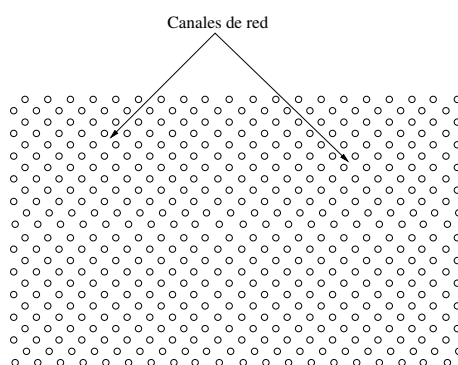


Figura 2.1: Esquema bidimensional de un material cristalino, cada círculo representa a un átomo del material

### 2.2.2. Materiales Policristalinos

Un material policristalino está formado por una estructura granular (ver figura 2.2). Los granos son definidos como pequeños volúmenes cristalinos con una orientación cristalográfica propia que cambia discontinuamente en su frontera<sup>10</sup>. Por lo tanto, el movimiento de cada ion dentro de cada grano es altamente anisotrópico debido a la estructura cristalina, pero completamente isotrópico en la región amorfa entre los límites de granos.

La anisotropía de un material policristalino depende considerablemente de la orientación y del tamaño de los granos cristalinos que lo conforman. Según esto, podemos clasificar a los policristales en dos tipos: en el caso que la mayoría de los cristales tengan la misma orientación, la anisotropía del cristal es más parecida al de un solo cristal, mientras que para el caso en que todas las direcciones tengan la misma probabilidad de ocurrir, su comportamiento es más bien isotrópico y su comportamiento macroscópico será más parecido al de un material amorfo, aún cuando, un único cristal siga mostrando un comportamiento anisotrópico.

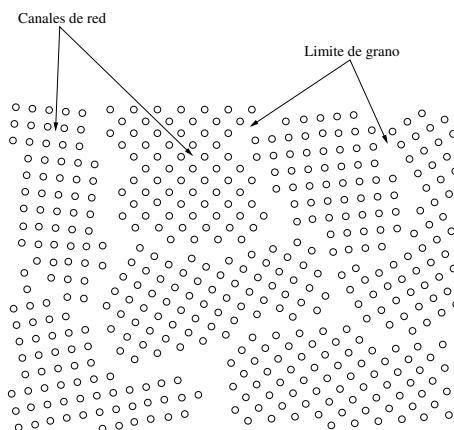


Figura 2.2: Esquema granular bidimensional de un material policristalino, donde cada círculo represente a un átomo del material

De esta manera, en general un material policristalino presenta un comportamiento intermedio debido a que comparte las propiedades conjuntas de materiales cristalinos y

<sup>10</sup>también conocida con el nombre de límite de grano.

amorfos, por lo tanto, dependiendo de las características físicas de sus granos.

## 2.3. Simulación de Materiales

En la actualidad los métodos numéricos permiten simular procesos dinámicos que involucran una gran cantidad de partículas<sup>11</sup>. En particular, en Ciencia de Materiales los métodos como el Monte Carlo (MC) y DM son ampliamente implementados.

El MC[7] es un método que se basa en la Mecánica Estadística y en la teoría de aproximación de colisiones binarias[8]. El algoritmo se basa en la selección de un parámetro de impacto aleatorio para el ion, calculando la energía y el ángulo de dispersión en cada colisión. Este proceso se repite una y otra vez, hasta que la energía del ion se reduce por debajo de un valor umbral, donde se considera detenido, ya que no cuenta con energía suficiente para desplazarse de manera apreciable.

Para entender los mecanismos dinámicos de interacción ion-sólido y los fenómenos colectivos en las interacciones, es necesario describir el movimiento de todos los átomos que actúan durante el proceso, en este caso el método de DM sería el más apropiado. En el concepto de DM, se resuelven las ecuaciones de movimiento de un conjunto de partículas, calculando las interacciones entre estas a partir de funciones que describen al potencial entre ellas. Los experimentos numéricos basados en MD proveen detalles que podrían no ser accesibles en el desarrollo experimental[9].

---

<sup>11</sup>En la actualidad la más poderosa de las supercomputadoras del mundo (BLUE GENE), The Alamos National Laboratory, USA, es empleada para llevar a cabo poderosas simulaciones en Dinámica Molecular

## Capítulo 3

# Implantación de Iones

Durante una colisión de un ion con los átomos que componen a un blanco, el ion pierde parte de su energía cinética y su dirección de movimiento puede cambiar considerablemente. En este comportamiento, resultante de las interacciones de los núcleos y los electrones del medio, los núcleos influyen en cuanto a su carga y masa pero no en cuanto a reacciones nucleares. Por lo que es usual que las colisiones atómicas estén en función de la “nube” electrónica. El problema básico podría parecer sencillo a simple vista, pero en las investigaciones se descubren cada vez más nuevos fenómenos que demuestran lo contrario. En particular, en una colisión atómica o molecular con un blanco se manifiestan -en diferencia a una colisión de dos partículas- son procesos tipo colectivo<sup>1</sup>, tales como: *formación de defectos*, *chisporroteo*, el *canaleo*, y *colisiones en cascada* entre otros.

### 3.1. Física de la Implantación de Iones

Al inicio del proceso de implantación los iones describen una trayectoria aproximadamente recta -no para iones livianos con respecto a los átomos del blanco- debido a que interactúan principalmente con los electrones “libres”, perdiendo energía principalmente por *frenado electrónico*. Cuando la energía de un ion alcanza un valor del orden de la en-

---

<sup>1</sup>en este tipo de procesos participan muchas partículas

energía vibracional de los átomos del blanco, las interacciones coulumbianas entre núcleos se hacen más efectivas, y la pérdida de energía por *frenado nuclear* se hace importante. La interacción de los iones con los núcleos produce fuerte cambios en sus direcciones de movimiento, por esto durante este dominio de frenado los iones describen una trayectoria más aleatoria. Todos los iones durante su movimiento sufren un proceso similar al descrito anteriormente, y continúa hasta cuando han perdido toda su energía <sup>2</sup> donde finalmente se detienen. Un esquema del proceso descrito anteriormente se ilustra en figura (3.1).

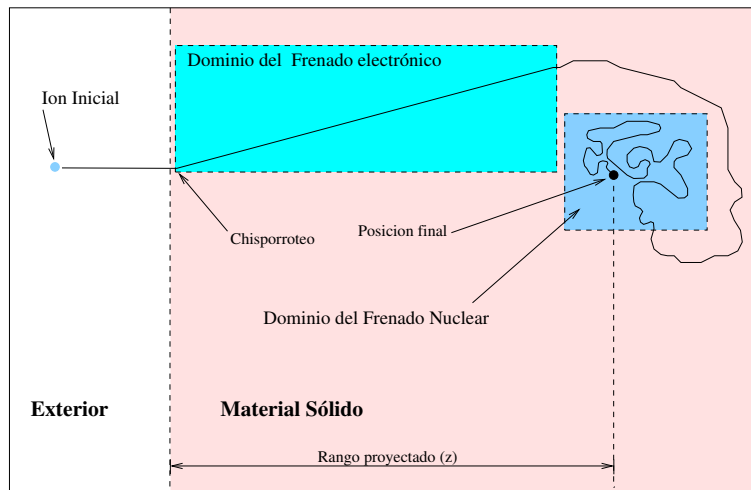


Figura 3.1: Ilustración de la trayectoria de un ion en un material sólido.

### 3.2. Frenado de iones en Sólidos

El estudio de iones atravesando por un material se logra conociendo la manera en que estos interactúan y de como cambia su energía. En un caso típico los iones van perdiendo su energía cinética constantemente debido a las interacciones que experimentan con los átomos y electrones “libres” encontrados a su paso por el material. Al proceso de *pérdida de energía* se le conoce con el nombre de *frenado*, en donde parte de la energía es cedida en forma de calor, mientras que la otra parte va produciendo defectos en la estructura

<sup>2</sup>en realidad los iones alcanzan una energía mínima o umbral en la que su posición no varía de manera importante

atómica del material.

Al inicio del proceso de frenado de un ion con energía cinética relativamente alta -dentro de rango bajas energías- va disminuyendo su velocidad principalmente por frenado electrónico, describiendo una trayectoria aproximadamente rectilínea<sup>3</sup>. A medida que la energía del ion va disminuyendo, las colisiones entre núcleos van siendo cada vez más efectivas, hasta un punto en que la pérdida de energía por frenado nuclear llega a ser la causa principal de frenado. Esta pérdida progresiva de energía se manifiesta como una fuerza de fricción (o frenado) y es comunmente expresada como una función del desplazamiento  $s$  y es definida como:

$$S = -\frac{dE}{ds}. \quad (3.1)$$

El frenado de los iones es comúnmente estudiado como la acción de dos procesos independientes. Los procesos de frenado en función del tipo de interacción son conocidos como: el *frenado electrónico* y el *frenado nuclear*. De tal modo, que el frenado total es la suma de estas dos contribuciones[10]

$$S = S_n + S_e. \quad (3.2)$$

Así mismo, hay dos rangos de energías establecidos (ver figura 3.2) para los cuales los procesos de frenados se comportan de forma diferente[11], estos son:

- **Bajas energías:** menores que  $100keV/uma$ [11], la pérdida de energía de los iones se debe al frenado electrónico y nuclear.
- **Altas energías:** mayores que los  $100keV/uma$ , la pérdida de energía se debe principalmente a la interacción con los electrones; pero en este caso el frenado es una situación compleja, donde las reacciones nucleares y emisiones de radiación<sup>4</sup>

---

<sup>3</sup>aunque para iones muy livianos las colisiones nucleares con átomos pesados son tan fuertes que la trayectoria puede ser más parecida a un camino aleatorio

<sup>4</sup>e.g., radiación de frenado ó la radiación de Cherenkov

pueden jugar un papel importante. No tenemos mayor interés en este rango de energías así que no se discutirá en adelante.

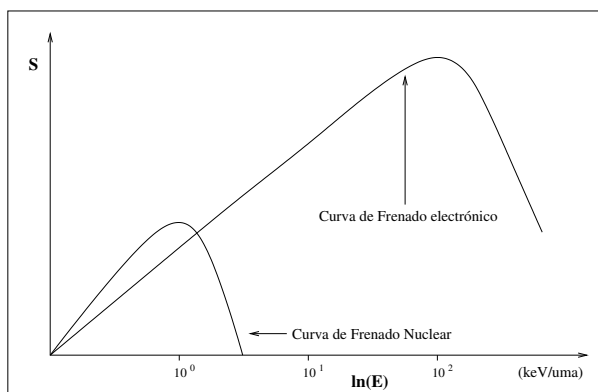


Figura 3.2: Una comparación entre el frenado electrónico y nuclear

En resumen, durante el proceso frenado a bajas energías, los núcleos influyen con la pérdida de energía de los iones debido a las interacciones coulumbianas, mientras que el frenado electrónico es la pérdida de energía debida a colisiones entre el ion y los electrones del material[12], comportándose como una fuerza de fricción que depende de la densidad electrónica local.

Para un caso típico de frenado, el máximo de pérdida de energía por colisiones entre núcleos ocurre típicamente a una energía entre  $1 - 10 \text{ keV/uma}$ , mientras que el máximo de pérdida de energía por colisiones electrónicas ocurre en el rango de  $0,1 - 1,0 \text{ MeV/uma}$ [12]. Para iones livianos en materiales pesados, el frenado nuclear es menor que el electrónico para todo el rango de energías[1].

### 3.2.1. Frenado Electrónico

Como se ha definido anteriormente, el frenado electrónico describe la pérdida de energía debido a la interacción con los electrones de un medio[13], por lo que depende de la configuración de los electrones en el material en cada instante.

En los materiales cristalinos los átomos forman una red periódica, y en consecuencia, la distribución electrónica -en las direcciones de red- presenta canales en donde las den-

sidades de núcleos atómicos y electrónica son considerablemente menores que en otros lugares. Esto significa, que los iones que se mueven por los canales pierden menos energía cinética que aquellos que no lo hacen, y por lo tanto, tienen un mayor alcance. En este caso, el frenado electrónico se convierte en un factor determinante para el cálculo de los perfiles de implantación[15]. Dentro del rango de energías de algunos kilo-electronvoltios ha sido posible derivar funciones de frenado que dependen de la posición  $r$  de la forma

$$S_e(E) = \left( \frac{dE}{dr} \right)_e . \quad (3.3)$$

Donde  $E$  es la energía del ion. Esta función se conoce como *fuerza de frenado electrónico* y describe la pérdida de energía del ion colisionante por unidad de distancia en un medio. Esta función contiene las pérdidas tanto por colisiones *elásticas* como las *inelásticas*. El término inelástico es utilizado para denotar que durante las colisiones pueden resultar excitaciones en la nube electrónica del ion. Sin embargo, en los metales -debido a que los procesos de excitación, ionización o captura no son importantes- podemos considerar el frenado electrónico producto de las colisiones elásticas. Visto de otra manera, un ion que penetra dentro de un blanco metálico va perdiendo energía en un “mar” de electrones debido las colisiones elásticas, ya que durante el frenado de iones a bajas energías solo estas juegan un papel importante[16].

El frenado electrónico para el caso de materiales sólidos, se puede parametrizar de dos maneras: local, cuando el frenado electrónico depende de la posición en el cristal, y no local, cuando el frenado electrónico es el mismo en cualquier lugar del cristal. La parametrización no local se puede emplear en materiales para los cuales el *channelling* no es significativo, pero en los materiales cristalinos el frenado electrónico es muy importante y depende estrictamente de la posición, así que la mejor manera de parametrizarlo es mediante la densidad electrónica local.

Para energías mayores que el máximo encontrado en la figura (3.2) de la curva de frenado electrónico, han sido derivadas algunas funciones de pérdida de energía por frenado electrónica, como la fórmula de Bethe-Bloch[17], con la que se puede calcular el frenado electrónico con algún razonable porcentaje de precisión. Mientras que para energías menores que algunos  $100keV/uma$  la situación es un poco distinta, y no ha sido

posible encontrar una expresión general para los diferentes materiales. Una solución a este problema fue hallado por Brandt y Kitagawa[18], quienes encontraron una función para el frenado electrónico de un ion, por medio de una expresión en función de la carga efectiva. La teoría de Brandt-Kitagawa (BK) forma las bases de muchas de las teorías modernas de frenado electrónico.

Según la teoría de carga efectiva de BK, para el caso de materiales cristalinos, el frenado electrónico de un ion pesado puede ser factorizado en dos términos, a saber, la carga efectiva del ion  $Z_{eff}$  y el frenado electrónico de un protón  $S_p$  de la forma

$$S_e = [Z_{eff}(v)]^2 S_p(\rho_e, v), \quad (3.4)$$

donde  $\rho_e$  es la densidad local de los electrones libres y  $v$  la velocidad del ion. La carga efectiva  $Z_{eff}$  se puede expresar como[10]

$$Z_{eff} = \gamma Z_1, \quad (3.5)$$

donde  $\gamma$  es la fracción efectiva de carga[18] para un ion con número atómico  $Z_1$ , que dentro de la teoría BK esta dada por la expresión

$$\gamma = q + C(1 - q) \ln \left( 1 + \left( \frac{4\Lambda}{r_s} \right) \right), \quad (3.6)$$

donde  $C$  es un factor débilmente dependiente del material y tiene un valor numérico de 0,5[19] y  $r_s = (3/(4\pi\rho))^{1/3}$  es el radio asociado de electrón libre. La *longitud de apantallamiento*  $\Lambda$  describe como los electrones apantallan el núcleo es función de la fracción de ionización  $q$  definida como[18]

$$\Lambda = \frac{2ka_0(1 - q)^{2/3}}{Z_1^{1/3} [1 - (1 - q)/7]}, \quad (3.7)$$

donde  $k = 0,24005$  y  $a_0$  es el radio de Bohr, y  $q$  es la fracción de ionización que para un ion en reposo depende del número de electrones ligados. Pero para un ion en movimiento la fracción de ionización varía en función de su estado de movimiento, por lo que ha sido usual expresarla como una función de la velocidad reducida relativa  $y_r$ [20],

$$q = 1 - e^{-0,95(y_r - 0,07)}, \quad (3.8)$$

donde  $y_r$  se define como

$$y_r = \frac{v_r}{v_0 Z_1^{2/3}} \quad (3.9)$$

con  $v_0$  la velocidad de Bohr.

La velocidad de los iones relativa a los electrones  $v_r$  es una función de la relación entre la velocidad del ion y a la velocidad de Fermi  $v_F$  de los electrones definida como[19],

$$\begin{aligned} v_r &= v \left( 1 + \frac{v_F^2}{5v^2} \right), & \text{si } v \geq v_F \\ v_r &= \frac{3v_F}{4} \left( 1 + \frac{2v^2}{3v_F^2} - \frac{v^4}{15v_F^4} \right), & \text{si } v < v_F \end{aligned} \quad (3.10)$$

La velocidad de Fermi de los electrones en un lugar dentro del material es la velocidad de los electrones correspondiente al nivel de Fermi dada como

$$v_F = \frac{1}{\alpha r_s}, \quad (3.11)$$

donde  $\alpha = (4/(9\pi))^{1/3}$ .

A partir de la ecuación (3.5) se puede obtener la carga efectiva, así que solo faltaría calcular el término de frenado para protones. Un resultado que se puede emplear fue obtenido por Yu Gott[13] para iones con energías menores que  $100keV$ , encontrando una función que solo depende de dos parámetros para cada par ion-átomo, de la siguiente forma

$$S_p = \alpha v(1 - \beta v)eV/\text{\AA}, \quad (3.12)$$

donde  $v$  es la velocidad del ion dada en unidades de  $10^8 cm/s$ ,  $\alpha$  y  $\beta$  son dos constantes, que para el caso de Nitrógeno-Hierro tienen los valores de 7,300 y 0,076 respectivamente[13].

### 3.2.2. Frenado Nuclear

El frenado nuclear se debe a la transferencia clásica de energía por colisiones coulombianas entre el ion y los átomos del material. Así que, mientras para partículas a altas energías,  $S \approx S_e$ , el frenado se debe principalmente a las interacciones con los electrones[20], a bajas energías, la contribución de las interacciones coulombianas con

los núcleos son más fuertes y frecuentes. De esta manera frenado nuclear  $S_n(E)$  se hace más importante y la pérdida total de energías se debe en mayor parte a estas colisiones.

$$S_n(E) = \left( \frac{dE}{dr} \right)_n. \quad (3.13)$$

En las simulaciones de DM no es necesario emplear una función de frenado nuclear, sino que esta transferencia de energía se produce de manera natural durante las colisiones con los nucleos. Así que para cada paso del tiempo se puede calcular el frenado nuclear como la cantidad de la energía transmitida al material en cada interacción como:

$$S_n(E) = \left( \frac{E_o - E_f}{\Delta s} \right)_{\Delta t} \quad (3.14)$$

donde  $S_n(E)$  es la pérdida de energía para una interacción de duración  $\Delta t$  para un desplazamiento  $\Delta s$ .

### 3.3. Física de las colisiones en cascada

Cuando un ion energético colisiona con un átomo le entrega parte de su energía. Si esta energía transmitida, es suficiente para que este átomo se pueda desplazar y colisionar con otro, y así sucesivamente de tal manera que esto le suceda a un gran número de átomos, iniciando toda una serie de colisiones atómicas iniciadas por el impacto inicial de un solo ion. A este fenómeno se le conoce como *colision en cascada*.

Una colisión en cascada puede ser dividida en tres etapas: *i)* durante el estado inicial los iones colisionan fuertemente contra átomos, esta etapa es llamada *fase colisional* y tarda aproximadamente alrededor de  $0,1 - 1,0ps$ . *ii)* Como resultado de estas colisiones los átomos cercanos a la trayectoria del ion inicial adquieren un movimiento de alta temperatura, para la cual sigue una etapa de dispersión de calor, que da lugar a la fase de *dispersión térmica* que tarda rigurosamente  $1ns$ . *iii)* Una vez la fase de dispersión térmica termina, normalmente quedan una gran cantidad de defectos distribuidos por todo el volumen del material. Estos defectos pueden ser de varias formas, desde vacancias y átomos intersticiales hasta complejas combinaciones y defectos volumétricos. A esta última fase se le llama *fase de relajación*.

### 3.4. Canaleo de Iones

En algunos materiales los átomos (o las moléculas) se distribuyen formando redes periódica, y es de esperar que el frenado electrónico no sea uniforme en todas las direcciones. Y de echo, los materiales cristalinos tienen direcciones bien definidas conocida como canales, en donde las densidades atómicas y electrónicas son considerablemente menores, permaneciendo más o menos constante a lo largo de alguna dirección específica. Esto significa que los iones que entran por un canal, sufren un frenado más débil que en otras direcciones de la red, y por lo tanto tienen mayor posibilidades de alcanzar una profundidad superior[14], a este fenómeno se le conoce con el nombre de *canaleo* o *channelling* en Inglés. El canaleo es un factor muy importante para tener en cuenta durante el desarrollo de simulaciones de implantación de iones en materiales cristalinos o policristalinos.

### 3.5. Rango de Implantación

Es un importante objetivo en el tratamiento superficial por implantación el de controlar la distribución de los iones implantados, ya que aún teniendo el mayor control posible del proceso no sería suficiente para predecir la posición final de los iones. Si las energías de los iones son inicialmente iguales, al entrar en contacto con la superficie de un material y luego de atravesar algunas capas atómicas sus energías serán en su mayoría distintas, por lo tanto, el camino recorrido por los iones que atraviesan una sustancia no será necesariamente el mismo. Así que cada ion implantado experimenta dispersión con los electrones y átomos del blanco sufriendo un número arbitrario de colisiones durante su recorrido, y por consiguiente describiendo una trayectoria diferente.

Por lo anterior, es conveniente caracterizar los iones implantados teniendo únicamente en cuenta la distancia total  $z$  alcanzada por un ion medida desde la superficie del blanco -la cual se conoce como rango proyectado-, sin tener en cuenta los corrimientos laterales relativos al punto de entrada en la superficie (ver figura 3.1). De esta manera, la distribución de iones implantados en función de la profundidad o perfil de iones

implantados  $C(z)$ , puede ser descrita en función de  $z$  de la siguiente forma[2].

$$C(z) = \Phi f(z) \quad (3.15)$$

con  $f(z)$  la distribución normalizada de iones y la dosis  $\Phi$  como,

$$\Phi = \int_0^{\infty} C(z) \cdot dz \quad (3.16)$$

tomando  $z = 0$  en la superficie del blanco.

De igual manera, es normalmente conveniente caracterizar el perfil de iones implantados por el rango medio proyectado  $R_P$  de las partículas; este es la distancia promedio que un grupo de iones que pasan a travéz de la superficie, y es definido como:

$$R_P = \int_0^{\infty} z \cdot f(z) \cdot dx \quad (3.17)$$

obteniendo para su desviación estándar,  $\sigma_P$ ,

$$\sigma_P^2 = \int_0^{\infty} (z - R_P)^2 \cdot f(z) \cdot dz \quad (3.18)$$

para un caso simple la distribución puede ser en primera aproximación una función de distribución Gaussiana. La concentración de iones implantados como una función de la posición esta dada por la siguiente

$$n(z) = \frac{\Phi}{\sqrt{2\pi} \sigma_P} e^{\left[ \frac{-(z-R_P)^2}{2\sigma_P^2} \right]} \quad (3.19)$$

donde el máximo de concentración ocurre a  $z = R_P$ , y  $\sigma_P$  es la desviación estándar de la distribución. La integral de  $n(z)dz$  da la dosis  $\Phi$ , y el máximo de concentración  $n(R_P)$  puede ser escrito como:

$$n(R_P) = \frac{\Phi}{\sqrt{2\pi} \sigma_P} \cong \frac{0,4\phi}{\sigma_P} \quad (3.20)$$

El rango de proyección de una distribución Gaussiana da una descripción de primer orden para los iones implantados en substratos amorfos.

## Capítulo 4

# Aproximación Clásica

Posiblemente la mejor aproximación al modelamiento de un sistema de partículas, consiste en la solución discreta de las ecuaciones de movimiento para todos sus átomos, en este caso, las simulaciones más precisas serán las basadas directamente en la Mecánica Cuántica. El principal impedimento para emplear un modelo mecánico cuántico se encuentra en el excesivo poder computacional que estas simulaciones exigen, a tal punto, que la representación para algunos sistemas complejos -de algunas partículas- es de hecho impracticable.

### 4.1. Aproximación de Born-Oppenheimer

Desde el punto de vista mecánico cuántico un sistema de átomos (o moléculas) está realmente formado por un grupo de núcleos y electrones que interactúan los unos con los otros. En este caos el hamiltoniano para el sistema puede ser escrito como[21]:

$$\hat{H} = \sum_i \frac{P_i^2}{2M_i} + \sum_k \frac{p_k^2}{2m} + \frac{1}{2} \sum_{ij} \frac{Z_i Z_j e^2}{|\mathbf{R}_i - \mathbf{R}_j|} + \frac{1}{2} \sum_{kk'} \frac{e^2}{|\mathbf{r}_k - \mathbf{r}_{k'}|} + \sum_{ik} \frac{Z_i e^2}{|\mathbf{R}_i - \mathbf{r}_k|} \quad (4.1)$$

donde los índices  $i, j$  y  $k$  y  $k'$  corresponden a los núcleos y los electrones respectivamente,  $\mathbf{R}_i$  y  $\mathbf{P}_i$  son las posiciones y los momentos correspondiente a los núcleos,  $\mathbf{r}_k$  y  $\mathbf{p}_k$  a los electrones,  $Z_i$  es el número atómico del núcleo  $i$ ,  $M_i$  su masa y  $m$  la masa de los electrones.

En principio, si se resuelve la ecuación de Schrödinger  $\hat{H}\Psi = E\Psi$  y se encuentran las funciones  $\Psi(\mathbf{R}_i, \mathbf{r}_a)$  que describen el sistema -lo cual en la práctica es un muy complicado y solo sistemas sencillos y aproximados han sido resueltos- se tendría a mano toda la información necesaria para el estudio de sus propiedades.

Un hecho importante de la Física Molecular, reside en la diferencia de masa que hay entre los núcleos atómicos y electrones, por este hecho, los núcleos mucho más pesados que los electrones, se mueven con una velocidad cuyo orden de magnitud es tres ordenes de magnitud menor que la velocidad de estos últimos. Esta característica condujo a Born y Oppenheimer[22] a considerar los movimientos de núcleos y electrones de manera independiente, sugiriendo, que se puede considerar el movimiento de los electrones relativo a los núcleos, suponiendo que los núcleos se encuentran en posiciones fijas, lo que a su vez, permite factorizar la función de onda de la siguiente manera

$$\Psi(\mathbf{R}_i, \mathbf{r}_a) = \Xi(\mathbf{R}_i)\Phi(\mathbf{r}_a; \mathbf{R}_i), \quad (4.2)$$

donde la función  $\Xi(\mathbf{R}_i)$  describe los estados correspondientes de los núcleos, y  $\Phi(\mathbf{r}_a; \mathbf{R}_i)$  de los electrones. Se observa como la función que describe los electrones solo depende paramétricamente de la posición de los núcleos. Con esta suposición, el problema puede ser reformulado en términos de dos ecuaciones de Schrödinger

$$\hat{H}_{el}(\mathbf{r}_a; \mathbf{R}_i)\Phi(\mathbf{r}_a; \mathbf{R}_i) = V(\mathbf{R}_i)\Phi(\mathbf{r}_a; \mathbf{R}_i), \quad (4.3)$$

donde,

$$\hat{H}_{el} = \sum_k \frac{p_k^2}{2m} + \frac{1}{2} \sum_{ij} \frac{Z_i Z_j e^2}{|\mathbf{R}_i - \mathbf{R}_j|} + \frac{1}{2} \sum_{nn'} \frac{e^2}{|\mathbf{r}_n - \mathbf{r}_{n'}|} + \sum_{in} \frac{Z_i e^2}{|\mathbf{R}_i - \mathbf{r}_n|} \quad (4.4)$$

y

$$\left[ \sum_i \frac{P_i^2}{2M_i} + V(\mathbf{R}_i) \right] \Xi(\mathbf{R}_i) = E \Xi(\mathbf{R}_i). \quad (4.5)$$

La ecuación (4.3) describe el sistema electrónico considerando a los núcleos en posiciones fijas. Los valores propios de la energía  $V(\mathbf{R}_i)$  dependerán paramétricamente de las coordenadas de los núcleos, y podemos llamar a estos valores *potenciales interatómicos* o *potenciales efectivos*. Para cada función  $V(\mathbf{R}_i)$  se puede obtener la función de onda que describe el estado de movimiento de los núcleos ignorando los detalles de la distribución

electrónica, a la vez que sus movimientos podrían ser descritos por ecuaciones relativamente más simples. Pero aún este continúa siendo un problema difícil de solución, lo cual implica que para realizar simulaciones de muchas partículas como es necesario en implantación de iones, habrá que aplicar algunas otras aproximaciones a fin poder realizar cálculos computacionales eficientemente.

## 4.2. Ecuaciones de Newton-Lorentz y Potenciales Interatómicos

Si las interacciones entre átomos pueden ser descritas mediante potenciales efectivos, significa que si por lo menos en teoría conocemos la forma del potencial entre las partículas, se podrían emplear las ecuaciones de Newton-Lorentz para describir su movimiento. Pero en la revisión anterior, se encontró que los sistemas a niveles atómicos obedecen más bien a las leyes cuánticas que a las clásicas, por lo cual, se requiere de algún criterio adicional para asegurar que el sistema se encuentra en las condiciones apropiadas para ser considerado dentro de la aproximación clásica. Con el objetivo de verificar si la aproximación clásica es válida, se puede recurrir a la definición de la longitud de onda térmica de Broglie[23], definida como:

$$\Lambda = \sqrt{\frac{2\pi\hbar^2}{mk_B T}}, \quad (4.6)$$

donde  $m$  es la masa atómica y  $T$  la temperatura. De esta manera, la aproximación clásica -pobre para sistemas muy livianos tales como  $H_2$ ,  $He$ ,  $Ne$ , entre otros, pero muy efectiva para átomos pesados con energías de rango intermedio- es justificada si  $\Lambda \ll a$ , donde  $a$  es la separación media entre los átomos más cercanos.

En todo caso, en la aproximación clásica se calculan las coordenadas y velocidades de todos los átomos para un instante del tiempo, resolviendo sus ecuaciones de movimiento, basándose en la mayoría de los casos en complejos potenciales efectivos debido a la naturaleza de las interacciones atómicas, y que muy a menudo restringen a que las ecuaciones no tengan una solución analítica, pero suficientemente simples para ser resueltas

por medio de métodos numéricos[24], por lo que desde un punto de vista computacional resultan ser muy prácticas.

#### 4.2.1. Potencial en metales

Los metales son elementos que tienen energías de ionización relativamente pequeñas, debido a que los electrones en las capas atómicas externas están débilmente ligados. Estos electrones pueden ser arrancados con la energía que se libera al momento de formarse los cristales. De esta manera, los metales son una red de iones positivos dentro de un “gas” electrónico constituido por los electrones liberados. Los electrones se mueven más o menos libremente por toda la red, y por lo tanto no están localizados[25], creando así, una densidad electrónica distribuida por todo el material responsable en parte de la unión del material, y que en la aproximación clásica puede ser descrita por medio una función de *distribución de densidad electrónica* en función de la posición de los núcleos.

Para el caso de  $\alpha$ -Hierro -con estructura cristalina cubica centrada en el cuerpo (bcc)- aparece una energía de cohesión adicional comparada con un sistema donde se consideren solo las interacciones binarias entre los átomos. Además, por no ser una estructura compacta, no puede ser estabilizado por medio de potenciales simples (e.g., como los potenciales de dos cuerpos). Teniendo en cuenta estas consideraciones, entre otras, se desarrolló para los metales y otros sólidos el *método de átomo embebido*[26] o EAM<sup>1</sup> basado en un potencial de la forma:

$$V = \sum_{i>j}^N \phi(r_{ij}) + \sum_i^N U_i(\rho_i), \quad (4.7)$$

donde  $\phi(r)$  es un potencial de interacción de dos cuerpos, y  $U(\rho)$  es una función embebida que proporciona una parte de la energía como función de la densidad electrónica  $\rho_i$ . Para un átomo dado  $\rho_i$  se calcula como una superposición de la contribución electrónica debido a sus átomos vecinos

$$\rho_i = \sum_{j \neq i}^N \rho_{ij}(r_{ij}) + \rho_0, \quad (4.8)$$

---

<sup>1</sup>por su nombre en ingles “*Embedded Atom Method*”

siendo  $\rho_i$  la densidad electrónica en el átomo  $i$  debido a todos los átomos  $j$  que lo rodean,  $\rho_{ij}$  la densidad electrónica del átomo  $j$ -ésimo en el lugar donde se encuentra  $i$  y  $\rho_0$  es una densidad electrónica de fondo. Así que la energía total del sistema metálico es la suma de la energía necesaria para embeber cada átomo en un fondo de electrones creado por los átomos vecinos mas la contribución debido a las interacciones coulombianas entre núcleos.

#### 4.2.2. Potencial de interacción ion-átomo

Las interacciones  $N^+$ -Fe, a pequeñas distancias experimentan una repulsión esencialmente coulombiana, correspondiente a la interacción entre los núcleos, pero a distancias relativamente grandes, la nube electrónica apantalla un núcleo del otro. De esta manera, el potencial de interacción puede ser descrito en forma general, por el producto de la repulsión coulombiana entre núcleos por una función de atenuación resultante del apantallamiento producido por la nube electrónica. En el caso de que solo halla dependencia de la distancia  $r$  de separación, el potencial se puede definir como[20]

$$V_{ij}(r) = \frac{Z_1 Z_2 e^2}{4\pi\epsilon_0 r_{ij}} \Phi_e(r_{ij}) \quad (4.9)$$

donde el término  $(Z_i Z_j e^2)/r$  denota la repulsión coulombiana entre los núcleos, con  $Z_i$  el número atómico del ion,  $Z_j$  el numero atómico de las partículas del blanco, y  $\Phi_e(r)$  es una función de atenuación; que describe el apantallamiento electrónico entre núcleos.

$c_i$	$d_i$
0,181750	-3,19980
0,509860	-0,94229
0,280220	-0,40290
0,028171	-0,20162

Cuadro 4.1: Coeficientes de la formula (4.10)

Un gran número de potenciales repulsivos de la forma (4.9) y funciones de apantallamiento  $\Phi_e(r_{ij})$  han sido propuestos. Entre estas la función de apantallamiento  $\Phi_U$

derivada por ZBL<sup>2</sup>[20] conocida como *función universal de apantallamiento*, la cual ha demostrado ser apropiada para describir las interacciones de los iones a energías medias.

$\Phi_U$  es una función del radio  $r$  definida de la forma

$$\Phi_U(x) = \sum_{i=0}^4 c_i e^{d_i x}, \quad (4.10)$$

donde  $x = r/a_U$ , y  $a_U$  es

$$a_U = \frac{0,8854a_0}{Z_i^{2/3} + Z_j^{2/3}}, \quad (4.11)$$

aquí  $a_0$  es el radio de Bohr, y los valores de  $c_i$  y  $d_i$  están dados en el cuadro (4.1)

---

<sup>2</sup>por los nombres de Ziegler-Biersack-Littmark

## Capítulo 5

# Modelamiento Computacional

En el contexto científico la palabra “modelamiento” -del Latin model- se relaciona comúnmente con dos significados: formulación de un modelo y modelamiento numérico. Este último término es frecuentemente asociado como sinónimo de simulación -del Latin simulare- numérica. Los actuales avances en métodos de predicción numérica encontrados en todos los dominios actuales de las ciencias de materiales, han establecido la disciplina de “ciencia computacional de materiales”, la cual es un resultado del trabajo en conjunto de la ciencia de materiales, la física, las ciencias computacionales, química, e ingeniería.

El dominio clásico de la *ciencias de materiales* estaba dirigida a las investigaciones de defectos de red y a sus interacciones a meso-escala. Hoy en día, la *ciencia computacional de materiales* se concentra principalmente en la investigación de redes cristalinas y las propiedades dinámicas a escala atómica, empleando métodos como el DM o MC. Siguiendo una jerarquía de escala, los niveles de modelamiento microestructurales pueden ser rigurosamente agrupados según su tamaño en los siguientes regímenes: nanoscópico, microscópico, mesoscópico, y macroscópico. En este contexto, el término nanoscópico se refiere al nivel atómico, microscópico a los defectos de red por debajo de la escala de granos, mesoscópico a los defectos de red a escala de grano, y macroscópico a la geometría de la muestra.

El empleo de métodos numéricos en ciencias de materiales es promovido por la creciente capacidad de los sistemas de computación -ya sea en la velocidad de procesamiento o de capacidad de almacenamiento- y a la creciente demanda de predicciones cuantitativas en la investigación y en la industria. En este contexto la solución numérica de un problema se le refiere como la simulación.

## 5.1. Ideas básicas del Modelamiento

Parte de abstracción científica consiste en reemplazar parte del mundo real bajo observación, por un modelo conceptual que permita reproducir sintéticamente las propiedades en investigación, pero preservando las características esenciales a pesar que se describe el sistema empleando una estructura más simple. Durante el modelamiento de un problema bajo investigación, se deben establecer condiciones de frontera apropiadas y (o) los valores iniciales de las variables. Esto permite convertir el modelo inicial en un problema con formulación matemática bien definida.

Para cualquier modelo, lo primero es precisar un conjunto de variables independientes y un conjunto dependientes. Las típicas variables independientes son el espacio y el tiempo, pero las variable dependientes son más específicas, y están relacionadas con las variables de estado. Estas últimas se deben escoger de manera que permitan una cuantificación lo suficientemente precisa de las propiedades que se encuentran bajo investigación. Una vez definido el sistema de variables, se fórmula un modelo computacional usualmente en forma de ecuaciones diferenciales, que básicamente son: un conjunto de ecuaciones de estado que expresan las propiedades en términos del conjunto de variables dependientes y un conjunto de ecuaciones de movimiento que definen la evolución del sistema en función de las variables independientes.

## 5.2. Dinámica Molecular (DM)

Llamamos Dinámica Molecular, a la técnica de simulación que describe la evolución en el tiempo de un conjunto de átomos (o moléculas) interactuantes por solución de sus

ecuaciones de movimiento. Un tratamiento exacto de este problema requiere la formulación y solución de la ecuación (4.1) de Schrödinger para todos los átomos considerados. Sin embargo, como se demostró en las ecuaciones (4.5) y (3.7), en un tratamiento cuasi-estático esto no es necesario, y se puede considerar la interacción de las partículas en términos de potenciales efectivos y las ecuaciones clásicas de movimiento[27].

Para el caso de un cuerpo sólido, la idea consiste en construir un modelo en función de las partículas que lo conforman. Desde el punto de vista teórico esto parecería una tarea muy complicada. Pero dentro del contexto de la DM clásica en donde se supone que cada átomo interactúa con cada uno de los demás, originando fuerzas que afectan las posiciones y movimientos de cada uno, dando como resultado un cambio de sus posiciones para cada instante de tiempo. Por consiguiente, para estas simulaciones se debe tener un especial cuidado en la selección de los potenciales a emplear, puesto que cuanto más detallada sea su descripción de las interacciones interatómicas menor será la necesidad de emplear parámetros de ajuste empírico dentro del modelo.

En las simulaciones DM los átomos son típicamente representados como masas puntuales, interactuando a través de potenciales que dependen de la distancia de separación. De esta manera, la fuerza  $F_i$  que actúa sobre cada ion  $i$  con los átomos  $j$  del material se puede calcular como

$$\mathbf{F}_i(r_i) = \sum_{j \neq i} \mathbf{F}_{ij}(r_{ij}) = - \sum_{j \neq i} \nabla V_{ij}(r_{ij}). \quad (5.1)$$

De manera que la evolución del sistema bajo la acción de estas fuerzas se obtiene mediante la solución numérica de las ecuaciones de movimiento, así que, para calcular la trayectoria de cada ion  $i$  en el sistema constituido por  $N$  átomos del material, es necesario resolver las ecuaciones de movimiento,

$$\frac{d\mathbf{P}_i}{dt} = \mathbf{F}_i \quad (5.2)$$

y

$$\frac{d\mathbf{r}_i}{dt} = \frac{\mathbf{P}_i}{m_i}. \quad (5.3)$$

Aquí  $m_i$  es la masa del ion  $i$  y  $F_i$  la fuerza sobre éste.

De manera que por medio de un método eficiente de integración y un adecuado paso del tiempo, se calcula la trayectoria del sistema en el espacio de fases  $6N$  dimensional -aunque tal trayectoria usualmente no sea relevante- por un algoritmo, compuesto en esencia, de las siguientes tres etapas principales:

- **Condiciones iniciales:** Se asignan los valores iniciales de posiciones y velocidades para todos los átomos, para un mismo instante de tiempo  $t$ .
- **Evolución en el espacio de fase:** Se resuelven las ecuaciones de movimiento y se obtiene la trayectoria del sistema en dicho espacio.
- **Resultados de la simulación:** Se calculan los resultados de la simulación.

### 5.2.1. Cómputo de las Fuerzas

Este es el cálculo que consume la mayor parte del tiempo dentro de una simulación. Si en particular aplicamos el modelo de interacción entre todas partículas como se ha planteado anteriormente, y consideramos la contribución de la fuerza sobre una partícula  $i$  debido a todas las demás (ver figura 5.1). Sería imposible calcular dicha fuerza debido a que el número de interacciones aumenta rápidamente con el radio y la cantidad de operaciones crecen proporcional a  $O(N^2)$ ; donde  $N$  es el número de partículas dentro del radio de acción. Por esto, en la práctica es normal establecer un radio de corte  $r_c$  de acción para la fuerza e ignorar todas las interacciones entre átomos separados por una distancia mayor a ésta[23], lo que se puede escribir como un nuevo potencial de la forma

$$V(r) = \begin{cases} V(r) & \text{si } r \leq r_c \\ 0 & \text{si } r > r_c \end{cases} . \quad (5.4)$$

La consideración anterior se basa en la rápida disminución de los potenciales con la distancia interatómica, de manera que, después de cierta distancia se encuentra que la contribución es muy pequeña, y por lo tanto este aporte puede ser despreciado. Una consecuencia de esta aproximación, es que la implementación del algoritmo se hace un poco más complicada, pero se ahorran enormes recursos computacionales.

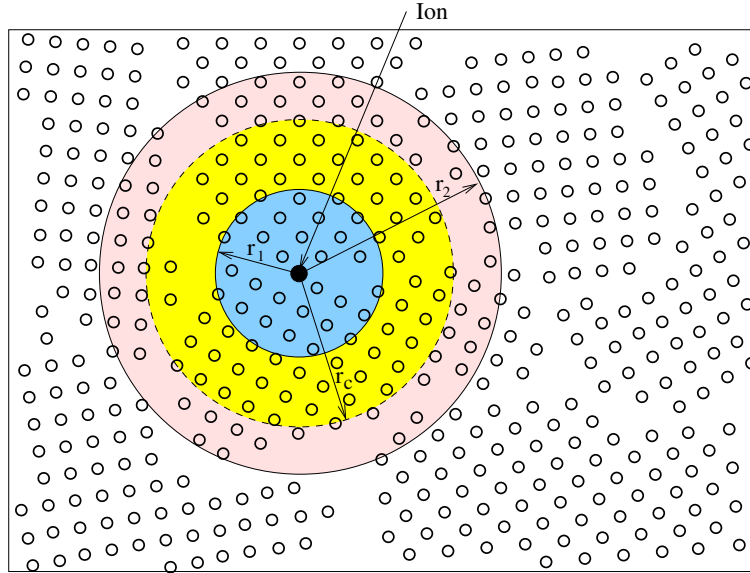


Figura 5.1: Interacción de todas las partículas, en este esquema se pueden apreciar la diferencia del número de partículas para dos radios de acción.

Por otro lado, el simple echo de un truncamiento en potencial trae consigo un nuevo problema, debido a que cada vez que una partícula “cruza” la distancia de corte, el valor de su energía hace un pequeño salto discontinuo. Un gran número consecutivo de estos eventos, podría probablemente arruinar la ley de conservación de la energía, y por lo tanto, producir resultados falsos e inesperados. Para prevenir ese tipo de problema, los potenciales deben tender suavemente hacia cero cerca del radio de corte, lo cual se logra comúnmente empleando una función de suavizado.

### Fuerza en Metales

Usualmente los potenciales para metales y semiconductores son diseñados con un radio de corte  $r_c$ ; es decir que el potencial tiende a cero a medida que se acerca a  $r_c$  junto con su primera y segunda derivada al menos y cero desde  $r_c$  en adelante. Las fuerzas derivadas de una función de energía que depende de las coordenadas, se expresan como

$$\mathbf{F}_i = -\nabla V_i(r_1, \dots, r_N). \quad (5.5)$$

Partiendo del hecho que los átomos de Hierro interactúan con un potencial del tipo Método Átomo Embebido (EAM), la fuerza se obtiene desarrollando (5.5), y queda expresada de la siguiente forma

$$\mathbf{F}_i = - \sum_{j \neq i} \left\{ \phi'(r_{ij}) + [U'(\rho_i) + U'(\rho_j)\rho'(r_{ij})] \right\} \frac{\mathbf{r}_{ij}}{r_{ij}}. \quad (5.6)$$

Donde  $\phi'(r)$  es la derivada del potencial, y  $U'(\rho)$  es la derivada de la función embebida, como están explicadas para la ecuación (4.7). Este cálculo es ligeramente más complejo que el requerido para un potencial de un sistema de dos cuerpos.

### Fuerza entre Ion-Átomo

La fuerza sobre un ion se obtiene de forma similar a la anterior, ya que posee una función de potencial que depende solo de la distancia entre átomos, lo cual se puede apreciar de la ecuación (4.9). De manera que podemos calcular la fuerza de la forma sobre el ion  $i$  como

$$F_i = \frac{Z_1 Z_2}{4\pi\epsilon_0} \sum_{j \neq i} \left[ \frac{\Phi'_U(r_{ij})}{r_{ij}} - \frac{\Phi_U(r_{ij})}{r_{ij}^2} \right] \frac{\mathbf{r}_{ij}}{r_{ij}}, \quad (5.7)$$

en donde  $\Phi_U(r_{ij})$  se calcula por medio de la ecuación (4.10), y  $\Phi'_U$  como

$$\Phi'_U(x) = \frac{1}{a_U} \sum_{i=0}^4 c'_i e^{d'_i x}, \quad (5.8)$$

donde  $x = r_{ij}/a_U$ ,  $a_U$  esta dado por la ecuación (4.11), y los parámetros  $c'_i$  y  $d'_i$  en la tabla (5.1).

$c'_i$	$d'_i$
-0,581560	-3,19980
-0,480430	-0,94229
-0,112900	-0,40290
-0,005679	-0,20162

Cuadro 5.1: Coeficientes de la formula (5.8)

Como puede observar, la función universal de ZBL de la ecuación (4.10) decrece monótonamente con la distancia, pero no se anula por completo aún para grandes distancias. En este caso es necesario introducir un radio de corte  $r_c$  suavemente como se mencionó anteriormente para evitar oscilaciones en la energía, por lo cual, se emplea una función de corte de tercer orden definida como,

$$S(r) = \begin{cases} 1 & (r < a) \\ \frac{(2r-3a+b)(r-b)^2}{(b-a)^3} & (a < r < b). \\ 0 & (b < r) \end{cases} \quad (5.9)$$

En este caso, los valores de  $a$  y  $b$  son de  $0,8r_c$  y  $r_c$  respectivamente, y  $r_c$  es del orden de la constante de red.

### 5.2.2. Unidades adimensionales

En simulación es conveniente expresar cantidades tales como energía, temperatura, tiempo, entre otras en unidades adimensionales[28]. Esto significa que debemos elegir unas convenientes unidades de escalamiento para energía, longitud y de masa, y entonces expresar todas las otras cantidades en términos de estas unidades básicas. Una elección natural de las unidades de escalamiento para un sistema compuesto de átomos, son las siguientes:

- Unidad de escalamiento de longitud  $\sigma = 1\text{\AA}$
- Unidad de escalamiento de energía  $\epsilon = 1k_B$ <sup>1</sup>
- Unidad de escalamiento de masa  $\mu = 1uma$

Para estas unidades de escalamiento básicas, la unidad de escalamiento para el tiempo esta dada por

$$\tau = \sigma \sqrt{\mu/\epsilon} \quad (5.10)$$

---

<sup>1</sup>Numéricamente igual a la constante de Boltzmann

Para transformar las cantidades reales a cantidades en unidades reducidas, solo hay que dividir las por su correspondiente unidad de escalamiento como sigue, para la energía

$$E_{red} = \frac{E}{\epsilon}, \quad (5.11)$$

para la masa

$$m_{red} = \frac{m}{\mu}, \quad (5.12)$$

y para la distancia queda

$$x_{red} = \frac{x}{\sigma}. \quad (5.13)$$

De manera similar para cualquier otra.

Una razón muy importante para emplear unidades reducidas en una simulación, es que cuando se utilizan cantidades reales directamente en el sistema de unidades internacionales, se encuentra que algunas de las variables implicadas dentro de los cálculos (e.g., la energía promedio de una partícula o su aceleración) podrían ser muy grandes o muy pequeñas comparados con la unidad; por lo tanto, luego de varias operaciones aún empleando valores estándar de coma flotante de doble precisión, enfrentamos el peligroso riesgo de obtener resultados erróneos debido a los *errores de truncamiento*<sup>2</sup>, y en el peor de los casos se podrían obtener resultados que creen un *overflow* o un *underflow* en la capacidad numérica de los registros de la computadora<sup>3</sup>. Mientras por el contrario, en unidades reducidas casi todas las cantidades de son del orden de la unidad, es decir se encuentran entre  $10^{-3}$  a  $10^3$  aproximadamente y todos los errores disminuyen enormemente, ya que una computadora de 32 bits tiene una precisión de punto flotante de  $10^{-16}$ . Además el empleo de unidades adimensionales no representa ningún nuevo inconveniente, ya que los resultados de una simulación obtenidos en unidades reducidas siempre pueden ser convertidos a sus valores en unidades reales con un simple escalamiento.

---

<sup>2</sup>los tipos de errores se analizarán más adelante

<sup>3</sup>esto significa que no es posible almacenar la cantidad en ninguno de los casos

### 5.2.3. Algoritmos de Integración en el tiempo

El motor de las simulaciones de DM es un algoritmo de integración en el tiempo, requerido para resolver las ecuaciones de movimiento de las partículas, y así calcular las trayectorias. Un algoritmo basado en el *método de diferencias finitas* temporal, tiene un intervalo de tiempo  $\Delta t$  entre puntos consecutivos. De esta manera, si se conocen las posiciones y sus derivadas temporales para el tiempo  $t$ , se pueden calcular las posiciones y las velocidades para el tiempo  $t + \Delta t$ .

Ya que los métodos de integración son aproximados, por lo tanto hay errores asociados a ellos. En particular, se puede distinguir entre dos tipos[30]:

- *Errores de truncamiento*, relacionados con la precisión del método de diferencia finita con respecto a la solución. Los métodos de diferencias finitas usualmente se basan en expansiones en series de Taylor truncadas en algún término.
- *Errores de redondeo*, relacionados con los errores asociados a una implementación en particular del algoritmo. Causados por los límites de las cifras decimales que pueden ser manejados en operaciones aritméticas por las computadoras.

Estos errores se pueden controlar variando el valor de  $\Delta t$ . Para grandes valores de  $\Delta t$  los errores de truncamiento dominan, pero decrecen rápidamente si el valor  $\Delta t$  disminuye. Por otro lado, los errores de redondeo decrecen más lentamente con la disminución de  $\Delta t$ , y dominan en los límites de pequeños valores de  $\Delta t$ , estos errores son más difíciles de controlar, debido a que dependen del número de bits empleados en el almacenamiento de las variables durante las operaciones. El uso de 64-bit -correspondiente a *dobles precisión* en algunos compiladores- ayuda a mantener los errores de redondeo a un valor mínimo.

Muchos de los algoritmos de integración, están basados simplemente en expansiones de Taylor truncadas. La serie de Taylor para las coordenadas de las partículas es

$$r(t + \Delta t) = r(t) + r'(t)\Delta t + \frac{r''(t)}{2}\Delta t^2 + \dots \quad (5.14)$$

Si truncamos esta expansión más allá del término en  $\Delta t^2$ , obtenemos el conocido algoritmo de Euler el cual tiene una precisión muy pobre. En particular, el algoritmo de

Euler no es reversible en el tiempo y sufre de pésimas oscilaciones de la energía, por consiguiente no es recomendado. Revisemos brevemente algunas alternativas.

### Algoritmo de integración Verlet

Este es uno de los algoritmos de integración en el tiempo comúnmente empleados en Dinámica Molecular, debido a su poca exigencia de memoria y su buena estabilidad para pequeños pasos del tiempo[31]. La idea básica parte de escribir dos expansiones de Taylor de tercer orden para las posiciones  $\mathbf{r}(t)$ , una un paso adelante y la otra un paso atrás en el tiempo, se obtiene:

$$\begin{aligned}\mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 + \frac{1}{6}\mathbf{b}(t)\Delta t^3 + O(\Delta t^4) \\ \mathbf{r}(t - \Delta t) &= \mathbf{r}(t) - \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 - \frac{1}{6}\mathbf{b}(t)\Delta t^3 + O(\Delta t^4),\end{aligned}\tag{5.15}$$

siendo  $\mathbf{v}$  la velocidad,  $\mathbf{a}$  la aceleración, y a  $\mathbf{b}$  la tercera derivada de  $\mathbf{r}$  con respecto al tiempo  $t$ . Sumando las dos expresiones y reordenando se llega a

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \mathbf{a}(t)\Delta t^2 + O(\Delta t^4).\tag{5.16}$$

Inmediatamente se puede ver que el error de truncamiento del algoritmo cuando el sistema avanza con un paso de tiempo  $\Delta t$  es del orden de  $\Delta t^4$ . Lo que finalmente se obtiene, es un algoritmo que esencialmente hace

$$\{\mathbf{r}(t), \mathbf{a}(t), \mathbf{r}(t - \Delta t)\} \longrightarrow \{\mathbf{r}(t + \Delta t), \mathbf{a}(t + \Delta t)\}.\tag{5.17}$$

Sin embargo hace falta una expresión explícita para el cálculo de las velocidades. Para esto, se encuentra que las velocidades pueden ser calculadas de las posiciones de la siguiente manera

$$v(t) = \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t - \Delta t)}{2\Delta t}.\tag{5.18}$$

Sin embargo, el error asociado a esta expresión, es del orden de  $\Delta t^2$  en lugar de  $\Delta t^4$ , lo cual es un problema para la estabilidad de la energía cinética. Para sobreponerse a esta dificultad, se han desarrollado algunas variantes del algoritmo Verlet.

### Algoritmo de integración Verlet Velocidad

Una mejor implementación del algoritmo Verlet básico, es conocido con el nombre de *Verlet Velocidad*[32], donde las posiciones, velocidades y aceleraciones son obtenidas al mismo tiempo de la siguiente manera

$$\begin{aligned}\mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \frac{1}{2}[\mathbf{a}(t) + \mathbf{a}(t + \Delta t)]\Delta t.\end{aligned}\tag{5.19}$$

En resumen, el algoritmo se puede representar

$$\{\mathbf{r}(t), \mathbf{a}(t), \mathbf{v}(t)\} \longrightarrow \{\mathbf{r}(t + \Delta t), \mathbf{a}(t + \Delta t), \mathbf{v}(t + \Delta t)\}.\tag{5.20}$$

### Algoritmo de integración Leap-Frog

El algoritmo Leap-Frog[31] es también muy empleado en la simulación de sistemas compuestos por muchas partículas, es un algoritmo equivalente al Verlet. En este algoritmo se evalúan las velocidades en un paso intermedio de tiempo y se emplean para el cálculo de las nuevas posiciones. Para derivar el algoritmo de Leap-Frog del Verlet, se inicia definiendo las velocidades en un paso intermedio de tiempo como sigue:

$$\mathbf{v}(t - \Delta t/2) \equiv \frac{\mathbf{r}(t) - \mathbf{r}(t - \Delta t)}{\Delta t}\tag{5.21}$$

y

$$\mathbf{v}(t + \Delta t/2) \equiv \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t},\tag{5.22}$$

De estas ecuaciones y la ecuación (5.16) se obtiene la siguiente expresión para las velocidades

$$\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t + \Delta t/2) + \mathbf{a}(t)\Delta t,\tag{5.23}$$

Así la ecuación (5.16) se puede reescribir en la siguiente forma

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t + \Delta t/2)\Delta t + O(\Delta t^4),\tag{5.24}$$

Obteniendo un algoritmo que en resumen hace

$$\{\mathbf{r}(t), \mathbf{a}(t), \mathbf{v}(t - \Delta t/2)\} \longrightarrow \{\mathbf{r}(t + \Delta t), \mathbf{a}(t + \Delta t), \mathbf{v}(t + \Delta t/2)\}.\tag{5.25}$$

A pesar de que se obtiene una expresión para las velocidades, estas son obtenidas para un paso intermedio de tiempo, significando una diferencia de tiempo de  $\Delta t/2$  entre el cálculo de las velocidades y de las posiciones. Esto resulta ser un problema al calcular la energía total del sistema en un tiempo  $t$  durante la simulación. Una expresión para la velocidad en el tiempo  $t$  es

$$\mathbf{v}(t) = \frac{1}{2}[\mathbf{v}(t - \Delta t/2) + \mathbf{v}(t + \Delta t/2)]. \quad (5.26)$$

La cual puede ser empleada en cálculos de energía entre otras.

### Algoritmo de integración de Smith & Harrison

El algoritmo Smith & Harrison[33] es un preciso algoritmo predictor-corrector empleado en problemas de implantación de iones. En este esquema es apropiado emplear un intervalo de tiempo variable, porque, a pesar que se mejora notablemente la estabilidad de las soluciones, se gasta mayor tiempo y memoria en los cálculos de las posiciones y velocidades. Está basado en la expansión en serie de Taylor truncada después del término  $\Delta t^3$

$$r(t + \Delta t) = r(t) + r'(t)\Delta t + \frac{r''(t)}{2}\Delta t^2 + \frac{r'''(t)}{6}\Delta t^3 + O(\Delta t^4). \quad (5.27)$$

Empleando la siguiente nomenclatura

$$\mathbf{r}_n = \mathbf{r}(t_n); \mathbf{r}_{n+1} = \mathbf{r}(t_n + \Delta t). \quad (5.28)$$

La derivada de la aceleración es estimada como

$$\mathbf{a}'_n = \frac{\mathbf{a}_n - \mathbf{a}_{n-1}}{\Delta t_{n-1}} + O(\Delta t_{n-1}). \quad (5.29)$$

Reemplazando (5.29) en la serie dada por (5.27), se encuentra que la predicción para las posiciones es

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n\Delta t_n + \frac{1}{6}[(3 + R)\mathbf{a}_n - R\mathbf{a}_{n-1}]\Delta t_n^2, \quad (5.30)$$

donde

$$R = \frac{\Delta t_n}{\Delta t_{n-1}} \quad (5.31)$$

La serie de Taylor para la velocidad esta dada por

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t_n + \mathbf{a}'_n \frac{\Delta t_n^2}{2} + \mathbf{a}''_n \frac{\Delta t_n^3}{6} + O(\Delta t_n^4) \quad (5.32)$$

La aceleración y su derivada calculadas de (5.30)

$$\mathbf{a}'_n = \frac{\mathbf{a}_{n+1} - R^2 \mathbf{a}_{n-1} + (R^2 - 1) \mathbf{a}_n}{\Delta t_n (1 + R)} \quad (5.33)$$

y

$$\mathbf{a}''_n = 2R \left[ \frac{\mathbf{a}_{n+1} - R \mathbf{a}_{n-1} + (R + 1) \mathbf{a}_n}{\Delta t_n^2 (1 + R)} \right]. \quad (5.34)$$

Reemplazado (5.33) y (5.34) en la serie (5.32), se obtiene la expresión para el cálculo de las velocidades

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \left[ \frac{(3 + 2R) \mathbf{a}_{n+1}}{1 + R} + (3 + R) \mathbf{a}_n - \frac{R^2 \mathbf{a}_{n-1}}{1 + R} \right] \frac{\Delta t_n}{6}. \quad (5.35)$$

La corrección en las posiciones se obtiene con

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n \Delta t_n + \left[ \frac{(2 + R) \mathbf{a}_{n+1}}{1 + R} + (4 + R) \mathbf{a}_n - \frac{R^2 \mathbf{a}_{n-1}}{1 + R} \right] \frac{\Delta t_n^2}{12}. \quad (5.36)$$

lo cual demanda dos evaluaciones de la fuerza por paso de tiempo. Para emplear este método se emplea el siguiente algoritmo:

- (a) Se calculan las nuevas posiciones empleando la ecuación (5.30)
- (b) Se calculan las aceleraciones  $a_{n+1}$
- (c) Se calculan las nuevas velocidades empleando la ecuación (5.35)
- (c) Se corrigen las posiciones empleando la ecuación (5.36)

Si en lugar de emplear un paso de tiempo variable, se emplea un paso de tiempo  $\Delta t$  constante, las ecuaciones (5.30) y (5.35) se reducen a la forma

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n \Delta t_n + [4 \mathbf{a}_n - \mathbf{a}_{n-1}] \frac{\Delta t_n^2}{6} \quad (5.37)$$

y

$$\mathbf{v}_{n+1} = \mathbf{v}_n + [5 \mathbf{a}_{n+1} + 8 \mathbf{a}_n - \mathbf{a}_{n-1}] \frac{\Delta t_n^2}{12}. \quad (5.38)$$

#### 5.2.4. Condiciones de Frontera Periódica

No importa que tan grande es el tamaño de un sistema en una simulación, este siempre tiene un tamaño finito. Así que, aún cuando el número  $N$  de átomos dentro de la simulación sea muy grande, este podría ser muy pequeño comparado con el número de átomos contenido en un fragmento macroscópico -del orden de  $10^{23}$  átomos- de materia, por lo que es necesario definir unas condiciones de fronteras convenientes en los límites del sistema.

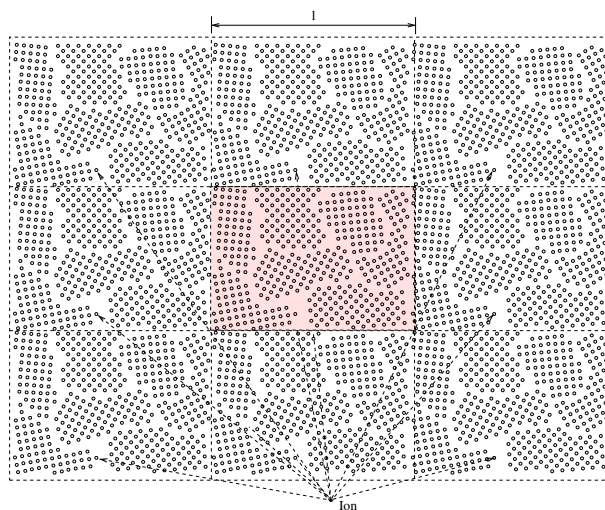


Figura 5.2: Esquema bidimensional de Condiciones de Fronteras Periódicas

Una posibilidad es dejar las fronteras libres, así que los átomos cerca a la “superficie” tendrán menos átomos vecinos que los átomos situados en un lugar del interior. Esta condición podría ser útil si lo que se necesita es simular un grupo (o cluster) de átomos. Pero si el objetivo es simular materiales macroscópicos, entonces sería apropiado definir unas condiciones en la frontera que hagan parecer al sistema de mayor tamaño que su tamaño real.

En este caso la solución está en emplear *condiciones de fronteras periódicas*[28], donde las partículas distribuidas dentro de un cubo de lado  $l$  es copiado infinitas veces por traslaciones en las tres direcciones cartesianas llenando completamente el espacio, en otras palabras, si una de las partículas esta localizada en una posición  $r$  del cubo, asumimos que esta partícula realmente representa un conjunto infinito de partículas,

que se hallan a la distancia correspondiente a la separación  $l$  entre cubos contiguos (ver la figura 5.2).

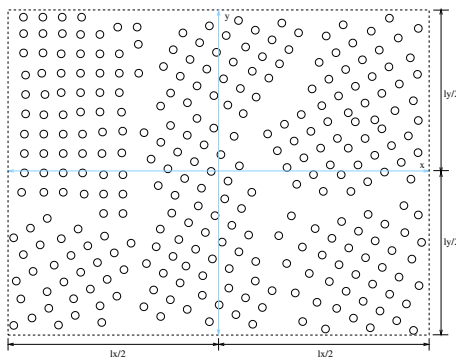


Figura 5.3: Condiciones de fronteras periódicas para un sistema de lado  $l$ .

Para llevar a cabo el efecto de fronteras periódica, se debe examinar antes de cada integración las coordenadas de todas las partículas del sistema, y en caso que alguna se encuentre por fuera de la región se debe corregir sus coordenadas y en este sentido traerla de regreso. De esta manera, si la coordenada  $r_x$  se debe encontrar en la región del espacio delimitada entre  $-l_x/2$  y  $l_x/2$ , como en la figura 5.3, donde  $l_x$  es el tamaño del sistema en la dirección del eje  $x$ , entonces la coordenada  $r_x$  debe cumplir la siguiente condición

$$r_x = \begin{cases} r_x - l_x, & \text{si } r_x \geq l_x/2 \\ r_x + l_x, & \text{si } r_x < -l_x/2 \end{cases} \quad (5.39)$$

Esta misma consideración se debe tener en cuenta cuando se está realizando el cálculo de la fuerza, en este caso, el efecto de periodicidad aparece cuando se está calculando la interacción con un átomo que se encuentra a una distancia aparente mayor que  $l_x/2$ . En la figura 5.2 se puede observar el caso de fronteras periódicas para dos dimensiones.

### 5.3. Métodos de optimización

En vista de los enormes costos computacionales que implica el desarrollo de simulaciones basadas en la DM hasta aquí explicada. En el estudio de la DM se han desarrollado métodos de optimización que aumentan el rendimiento y capacidades a niveles importantes, permitiendo la implementación de sistemas macroscópicos aún en computadores

personales, por esta razón, las optimizaciones son un tema de significativa importancia. Con esta intención, la búsqueda de parámetros que ayuden a mejorar el rendimiento, también requiere de identificar las ventajas que puedan obtenerse a partir de las propias condiciones del sistema y emplearlas en la elaboración de algoritmos más eficientes[29].

### 5.3.1. Método Lista-Celda

Lo primero es aprovechar el radio de corte  $r_c$ , ya que este garantiza, que no haya la necesidad de buscar partículas más allá de esta distancia de separación. Debido a esto, el espacio puede ser por lo tanto, dividido en celdas cúbicas de lado como mínimo de longitud  $r_c$ , organizando a todas las partículas por medio de las celdas, por lo que comúnmente a esta técnica se le conoce como *lista-celda*.

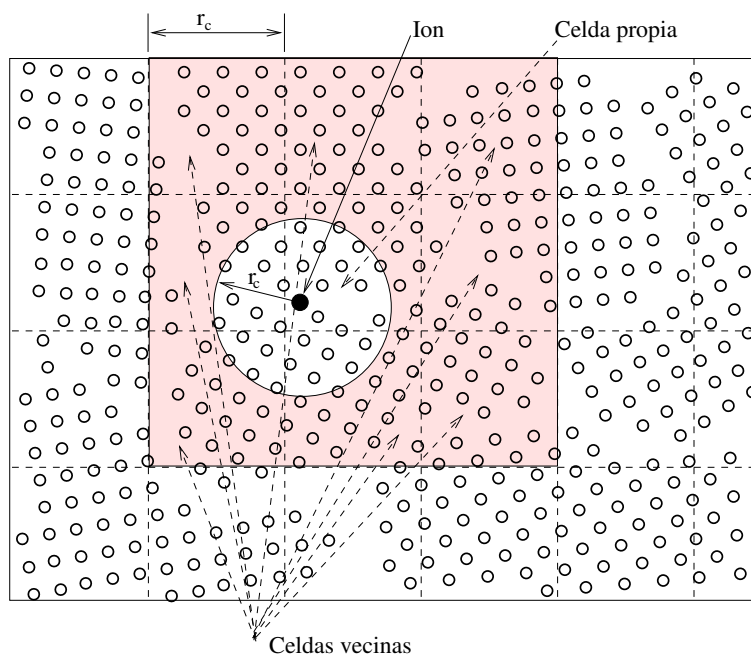


Figura 5.4: Esquema del método lista-celda para la búsqueda de los átomos vecinos.

De la figura (5.4) se puede ver que el espacio está dividido en  $M = N_x \times N_y$  celdas, donde  $N_x = l_x/r_c$  y  $N_y = l_y/r_c$ , lo que puede fácilmente extenderse para las tres dimensiones. De manera, que para cada celda se puede construir una lista de las partículas en

su interior como

$$M_{mn} \leftarrow (i, j) \quad , \text{ si } \begin{cases} n(l/2) \leq r_{ij,x} < (n+1)(l/2) \\ m(l/2) \leq r_{ij,y} < (m+1)(l/2) \end{cases} \quad (5.40)$$

Obteniendo la matriz  $M_{mn}$ , que contiene la partículas de cada celda  $mn$ . Este es uno de los métodos más importantes de optimización, disminuyendo de manera considerable el tiempo de búsqueda de los átomos vecinos<sup>4</sup> dentro de la esfera de corte, ya que solo hay que buscar a los vecinos dentro de la celda propia<sup>5</sup> y en las celdas inmediatamente adyacentes, tal como se muestra en la figura 5.4 y no por todo el volumen, como sería necesario para un potencial sin un radio límite.

### 5.3.2. Método Lista-Capa

Otra técnica de optimización muy importante, es la de *lista-capa*. Aquí el concepto consiste en agregar una delgada capa adicional de partículas de grosor  $d_{sh}$  alrededor de la esfera de corte de radio  $r_c$ , resultando una nueva esfera de radio  $r_{sh} = r_c + d_{sh}$  (ver figura 5.5) adicionando a la lista de átomos vecinos un pequeño número adicional de partículas, que se encuentran por fuera de la esfera de corte, pero que están muy próximos a entrar en ella.

La idea de emplear este método, está en que no sea necesario actualizar la lista de vecinos para cada paso del tiempo, sino que, podamos esperar hasta que el valor de la acumulación del máximo desplazamiento en cada paso del tiempo  $i$  no exceda la distancia  $d_{sh}$ , esto en formulación matemática es:

$$\sum_{pasos} (max_i\{v_j\}) > \frac{\Delta r}{2\Delta t}. \quad (5.41)$$

Donde la función *max* encuentra la partícula  $j$  más veloz dentro del sistema en el paso de tiempo  $i$  y *pasos* es el número de pasos de tiempo sin tener que actualizar la lista de las partículas. Así que antes de realizar una actualización en de la lista de las partículas

<sup>4</sup>Se le llaman átomos vecinos a aquellos que se encuentran dentro de la esfera de interacción dada por el radio de corte del potencial.

<sup>5</sup>Celda se encuentra el ion.

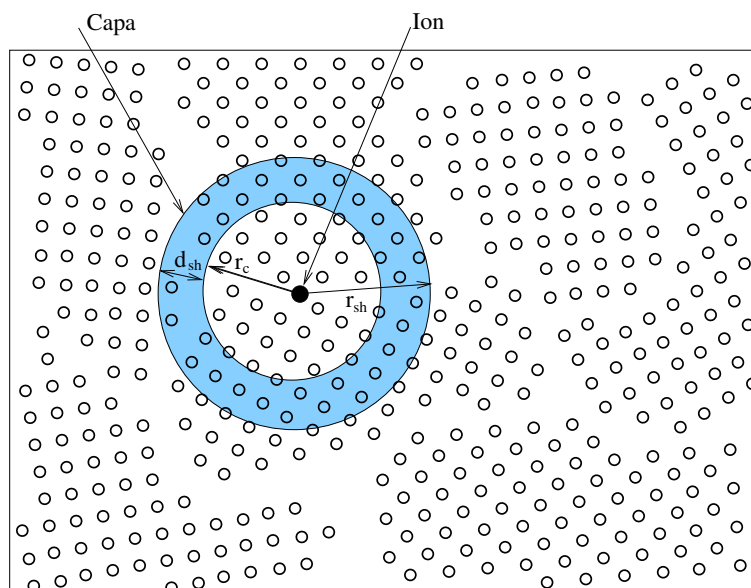


Figura 5.5: Esquema del método lista-capa para la búsqueda de los átomos vecinos.

vecinas, podemos primero, chequear si el criterio de desplazamiento ha sido excedido. El éxito de este método se sustenta en el lento cambio del entorno microscópico, lo cual implica finalmente que la lista de vecinos permanezca válida sobre varios pasos del tiempo para todo el sistema -típicamente de 10 a 20 pasos- aún para valores pequeños de  $d_{sh}$ .

El resultado de aplicar la técnica descrita anteriormente y la técnica lista-celda, da como resultado, un algoritmo muy eficiente, que permite realizar simulaciones de sistemas con un número enorme de partículas, aún disponiendo solo de computadoras personales. Un esquema completo de las técnicas de optimización propuestas se puede observar en la figura (5.6).

El efecto total de la utilización de estos métodos de optimización, es una reducción del orden de  $O(N^2)$  -de interacción de todas las partículas- a un orden de  $O(N)$  -de interacción con las partículas dentro de la esfera corte- en el cálculo de la fuerza en la simulación.

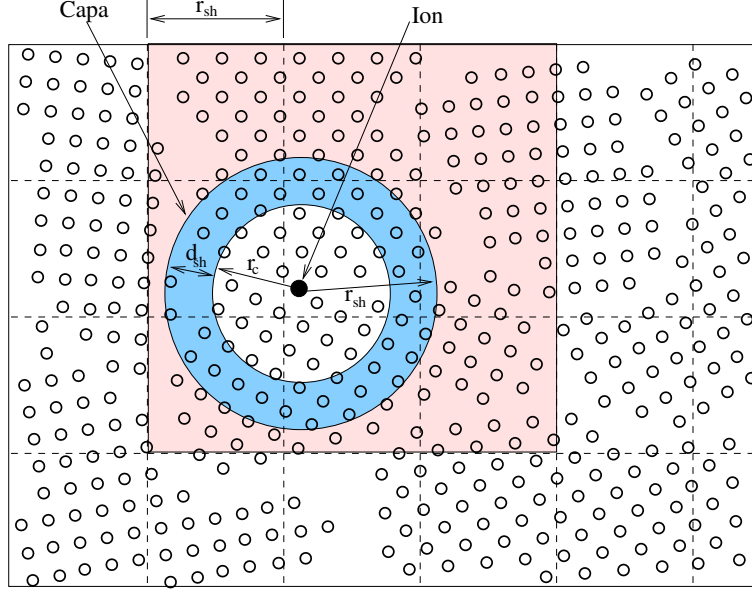


Figura 5.6: Esquema completo del método de optimización.

### 5.3.3. Paso de Tiempo Variable

El movimiento de las partículas se hace a través de pasos discretos en el espacio, la distancia máxima que podría desplazarse una partícula está determinada por el paso del tiempo  $\Delta t$  y por su velocidad  $v_i$ . Entonces, la distancia recorrida  $\Delta r_i$  se puede calcular como

$$\Delta r = v_i \Delta t. \quad (5.42)$$

Como es usual, en las simulaciones de DM de partículas interactuando con superficies sólidas, el paso del tiempo  $\Delta t$  puede ser determinado a partir de la energía de las partículas incidentes, las cuales se mueven a velocidades extremadamente altas comparadas con las del blanco.

Un cálculo apropiado de paso de tiempo variable durante la simulación con la intención de acelerar el cálculo del proceso colisional, se puede obtener a partir de las velocidades de las partículas, ya sea, teniendo en cuenta a cada una independientemente, o a todas en conjunto. Una condición de actualización del paso de tiempo podría aplicarse si limitamos la partícula más veloz a un valor de desplazamiento constante, consiguiendo que ninguna supere este valor desplazamiento máximo. En acuerdo con

lo anterior definimos a  $\Delta r_{max}$  como el máximo desplazamiento, entonces el paso del tiempo apropiado para cada paso del tiempo sería dado por la siguiente relación

$$\Delta t = \frac{\Delta r_{max}}{v_{max}}. \quad (5.43)$$

donde  $v_{max}$  es la velocidad de la partícula más veloz en el sistema, y que corresponde a uno de los iones incidentes. De esta manera, el paso del tiempo es recalculado durante la simulación manteniendo a todas los átomos moviéndose a un desplazamiento más o menos constante, lo cual es verdaderamente importante para energías bajas, debido que para este rango energías no habría un movimiento apreciable, si se emplea un paso del tiempo constante.

## Capítulo 6

# Simulación de Implantación de Iones

Para realizar el estudio de la implantación de iones, debemos modelar el material sólido y los iones a implantar. Para el caso de esta tesis, se trata de una simulación de implantación de Iones de Nitrógeno en Hierro policristalino, por lo cual, es necesario generar antes que todo la porción del material a implantar.

La metodología seguida para la construcción del blanco fue la de crecimiento de polí-cristales por nucleación, ya que es apropiada para ser desarrollada bajo los lineamientos de la DM. Para esto, inicialmente se generan algunas semillas<sup>1</sup> de bcc-Fe en direcciones aleatorias, como se aprecia en la figura (6.1) y luego se llena el resto de espacio con átomos de Hierro en estado líquido, es decir, con una energía un poco mayor que la energía del estado enlazado, de modo que los átomos se puedan mover más o menos libremente por todo el volumen.

El crecimiento de los cristales, se produce a medida que los átomos libres van perdiendo su energía por múltiples colisiones con los átomos en las semillas. La temperatura en las semillas se mantiene constante así que la energía transmitida se va extrayendo por enfriamiento, esto sucede hasta que la energía de los iones no es suficiente para escapar

---

<sup>1</sup>se le llaman semillas a cada pequeña porción de material cristalino.

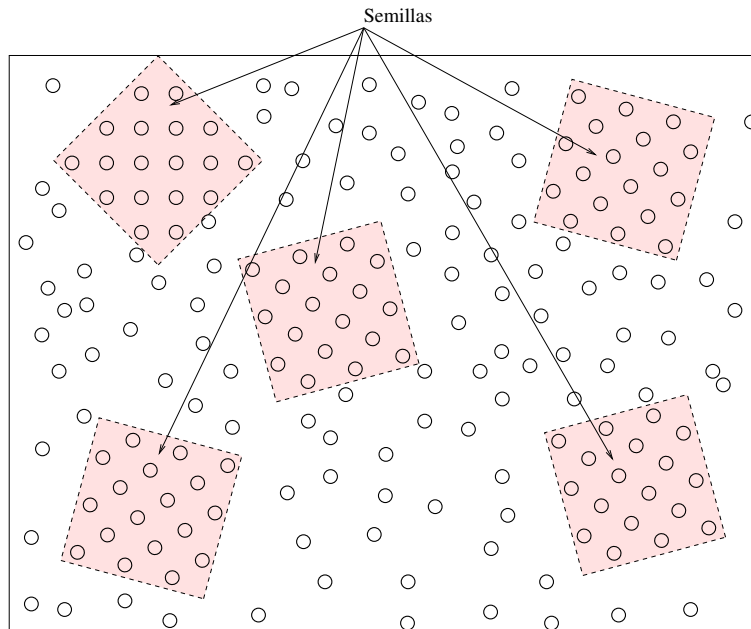


Figura 6.1: Esquema de semillas y átomos libres.

del enlace, y finalmente quedan atrapados. De esta manera, debido a que la forma de la estructura cristalina está implícita en el potencial como la configuración de mínima energía, los cristales empiezan a crecer en todas las direcciones (ver la figura 6.2) de manera aleatoria, formando finalmente una estructura granular.

Una vez que contamos con el blanco, solo es necesario incluirlo dentro del modelo de implantación.

## 6.1. Algoritmo para Implantación de Iones

A continuación se plantea el algoritmo básico basado en el método de integración de Smith & Harrison, utilizado por el programa de simulación de implantación con el que fueron llevadas a cabo las simulaciones durante el desarrollo de este trabajo.

### 1. Condiciones iniciales

- Elección del Tiempo total  $t_{max}$  de la simulación y paso de tiempo  $\Delta t$ .



- Rango proyectado de cada ion.
- Concentración de iones.

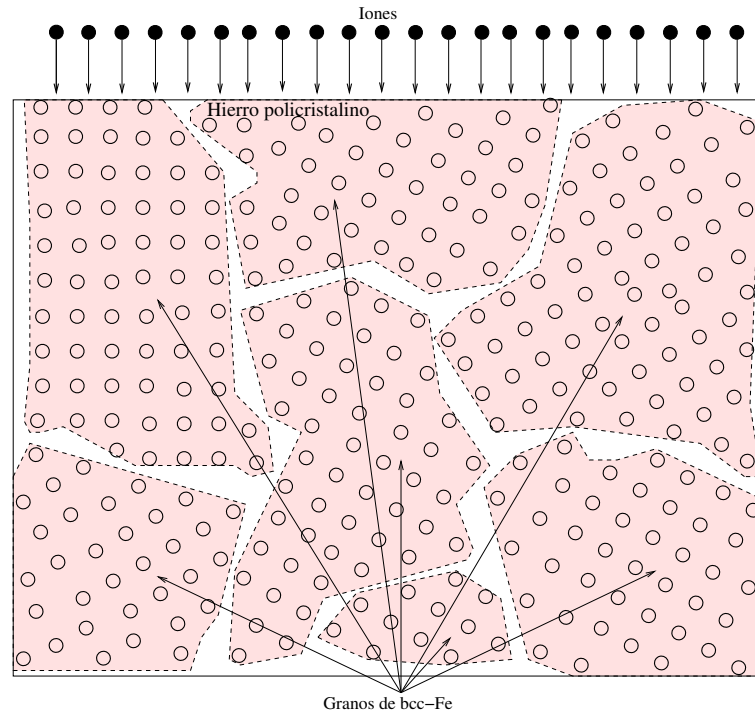


Figura 6.3: Esquema bidimensional del estado inicial de una simulación en implantación.

En la figura (6.3) se resume en un esquema bidimensional el estado inicial de la simulación, en donde se observa que inicialmente los iones se encuentran a una pequeña distancia sobre el blanco y poseen una velocidad con dirección normal a la superficie. La implementación de este algoritmo, dió como resultado un código desarrollado en C/C++ (algunas de las funciones se pueden ver en al anexo A). El empleo de C/C++ se debe principalmente, a la necesidad de manejar complejas estructuras de datos y la asignación dinámica de memoria.

## 6.2. Parámetros Físicos y Computacionales

El tamaño real del material fue un cubo de lado  $28,664\text{\AA}$  para un total de 2385 átomos de Fe. El número de granos crecidos en el volumen fue de 5 granos con un

tamaño promedio de  $4,3904nm^3$ . Otros parámetros están especificados en la siguiente tabla.

Constante de red del bcc-Fe	2,8664 Å
Masa atómica del Fe	55,847uma
Masa atómica del Ion	14,007uma
Radio de corte	4,095Å

Cuadro 6.1: Parámetros empleados durante las simulaciones.

## Capítulo 7

# Resultados

Se realizaron simulaciones de implantación para bcc-Fe policristalino y para bcc-Fe monocristalino, con la intención de observar el comportamiento de los iones de Nitrógeno debido a la estructura del material. Así se le hizo el seguimiento a la trayectoria de unas 100 partículas para cada caso, y para una energía inicial de  $10\text{keV}$ . La temperatura del blanco se mantuvo a  $240\text{K}$  durante toda la simulación, de esta manera la longitud de De Broglie es del orden de  $0,151\text{Å}$ , lo cual garantiza que podamos aplicar la aproximación clásica.

En el la gráfica 7.1 se presentan las trayectorias resultantes para el caso de un blanco de bcc-Fe monocristalino. Se puede ver como la estructura de la red cristalina afecta las trayectorias de los iones. Los iones que entran por los canales del monocristal tienen menos obstáculos en su camino y por lo tanto gastan menos energía debido a las colisiones con los núcleos, además, debido a que en los canales la densidad electrónica es menor también pierden menos energía por frenado electrónico, y por otro lado al tener una trayectoria más rectilínea recorren un una mayor distancia promedio.

Para el caso de un blanco de bcc-Fe en estado policristalino las trayectorias encontradas son -como se puede ver de la gráfica 7.2- más aleatorias, describiendo un movimiento rectilíneo para distancias cortas pero cambiando abruptamente de dirección debido a su estructura granular. Así que los iones se desplazan en promedio menos

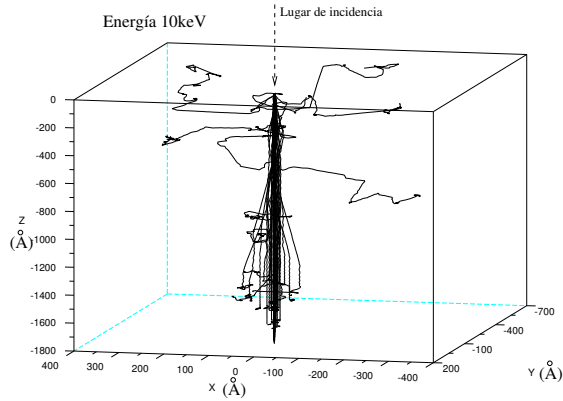


Figura 7.1: Gráfica de la trayectoria de 100 iones implantados en bcc-Fe monocristalino con energía inicial de  $10keV$ .

que en el caso de un material cristalino puro, y por lo tanto describen un desplazamiento medio menor.

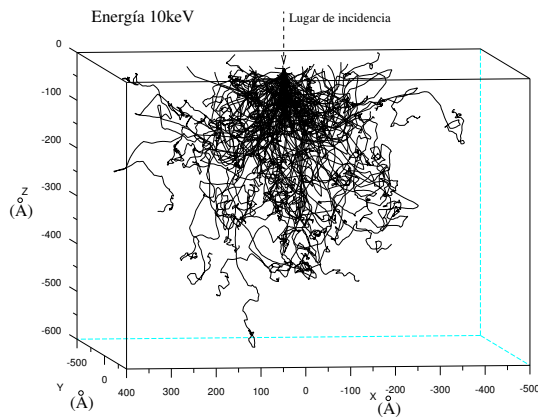


Figura 7.2: Gráfica de la trayectoria de 100 partículas implantadas en bcc-Fe policristalino con energía inicial de  $10keV$ .

Inicialmente se realizó el cálculo de los perfiles de iones implantados en bcc-Fe puro para las energías de  $10keV$  y  $20keV$ , empleando una dosis de 11000 iones. Como se encontró en los cálculos, una dosis mayor a 10000 iones es suficiente para obtener una buena descripción de un perfil de implantación. Inicialmente los iones parten de una misma distancia - de algunos Å- de separación con respecto a la superficie (ver la figura (6.3)) del blanco, pero distribuidos aleatoriamente en un área cuadrada de lado  $28,664\text{Å}$

sobre ésta. Al final de la simulación, se obtiene los perfiles mostrados en las figuras (7.3) y (7.4), donde se aprecia como en ambos casos la mayoría de los iones tienden a ubicarse en la parte final del rango máximo alcanzado.

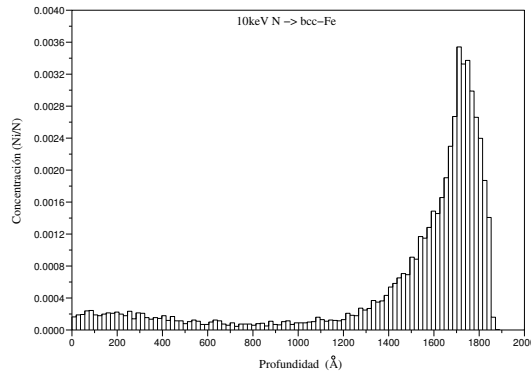


Figura 7.3: Perfil de implantación para partículas con energía inicial de  $10\text{keV}$  en bcc-Fe monocristalino.

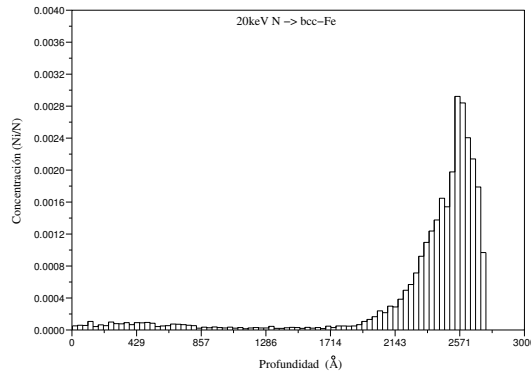


Figura 7.4: Perfil de implantación para partículas con energía inicial de  $20\text{keV}$  en bcc-Fe monocristalino.

Los siguientes perfiles de implantación por profundidad fueron calculados para las energías de 10, 20 y  $30\text{keV}$ , (ver figuras (7.5), (7.6) y (7.7 respectivamente), aquí también las posiciones iniciales de los iones fueron generadas aleatoriamente y el ángulo de incidencia fue de 0 grados con respecto a la dirección normal de la superficie. El número total de iones implantados fue de  $N = 11000$  en cada caso. La profundidad esta dada en

Å y la concentración está normalizada con el número total de partículas implantadas. El tiempo de computo de cada simulación fue de aproximadamente 12h, pero el tiempo de real que tardo la de implantación fue de alrededor de  $1 - 2ps$ .

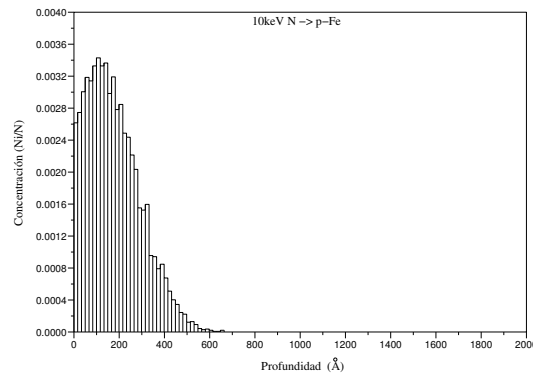


Figura 7.5: Perfil de implantación para partículas con energía inicial de  $10keV$  en bcc-Hierro policristalino.

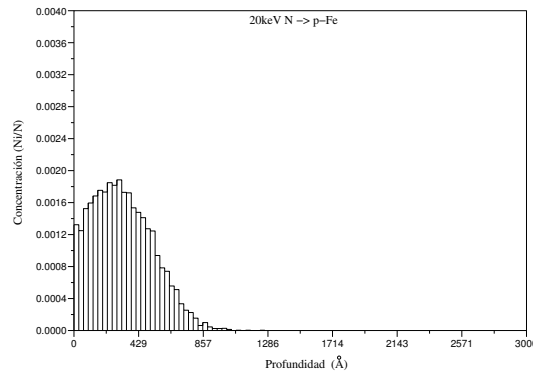


Figura 7.6: Perfil de implantación para iones de Nitrógeno con energía inicial de  $20keV$  en bcc-Hierro policristalino.

La comparación por superposición de los perfiles encontrados en la figura (7.8), permite ver como varía el comportamiento de los perfiles en función de la energía. En esta figura -teniendo en cuenta que todos tienen el mismo número de partículas- se aprecia como disminuye el máximo encontrado del perfil, y como se va corriendo hacia dentro del material y se encuentra que a medida que aumenta la energía inicial de los iones el

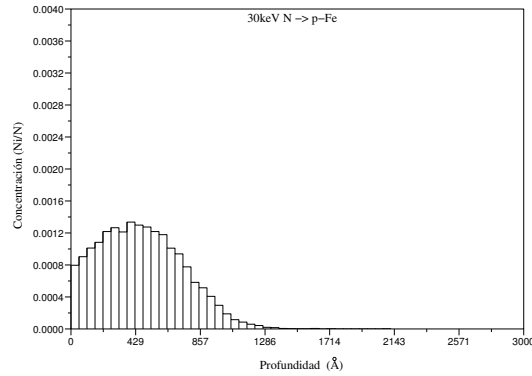


Figura 7.7: Perfil de implantación para iones de Nitrógeno con energía inicial de  $30keV$  en bcc-Hierro policristalino.

máximo se aleja de la superficie, pero el máximo de la concentración va disminuyendo si la dosis (número) de iones permanece constante.

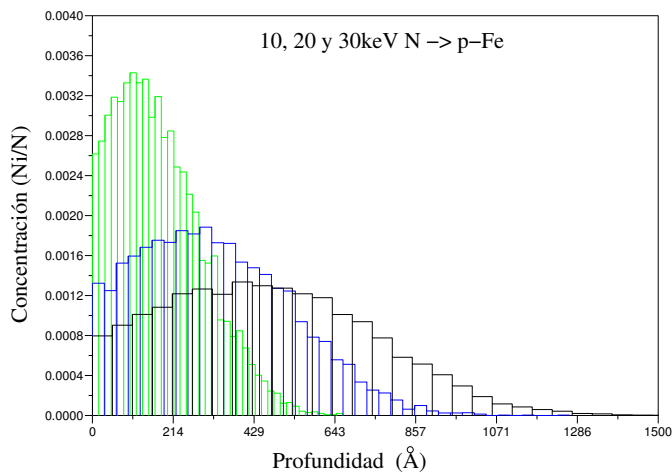


Figura 7.8: Superposición de los perfiles de implantación en bcc-Fe policristalino a 10, 20 y  $30keV$ .

Los rangos medios proyectados  $R_p$  para las energías de 10, 20, 30, 40 y  $50keV$ , se encontrarán registrados en la tabla número (7.1).

En la gráfica (7.9) se puede observar la linealidad del rango proyectado  $R_p$  en función

$E_0(keV)$	$R_p(\text{\AA})$
10	194,029
20	342,182
30	483,800
40	633,259
50	785,037

Cuadro 7.1: Rangos medios proyectados para las diferentes energías.

de la energía inicial de los iones  $E_0$ . Este resultado es muy importante para las investigaciones y es una excelente guía para poder realizar estimaciones de la distribución de los iones para otras energías.

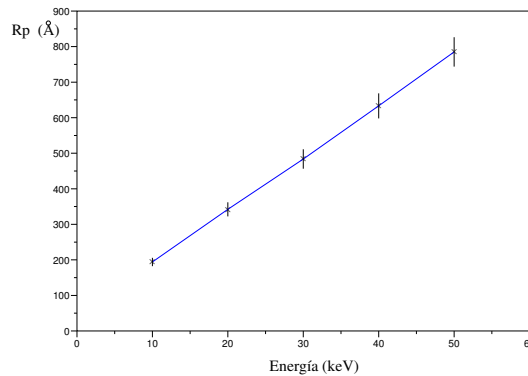


Figura 7.9: Gráfica de los rangos proyectados en función de la energía inicial de implantación para bcc-Fe policristalino.

Los resultados de los perfiles encontrados en la figura (7.10) fueron obtenidos por medio de resultados experimentales de análisis SIMS<sup>1</sup> de muestras de Acero SAE1020<sup>2</sup> implantado con Nitrógeno a energías de 10 y 20 keV. Estos datos son un buen punto de comparación para los resultados obtenidos por las simulaciones.

En la gráfica (7.11) se comparan los perfiles experimentales con los obtenidos por

<sup>1</sup>Secondary Ion Mass Spectrometry

<sup>2</sup>estas muestras de acero son un 98% de Hierro en estado policristalino

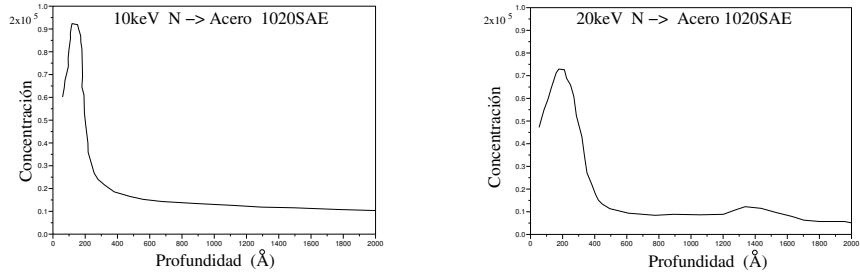


Figura 7.10: Perfiles SIMS de implantación para iones de Nitrógeno con energías de  $10keV$  y  $20keV$  en Acero.

simulación por medio de una superposición de los datos normalizados. Podemos observar que hay un buen ajuste entre el perfil encontrado para la simulación a  $10keV$  con el encontrado por medios experimentales, sin embargo se encuentra una discrepancia para los cálculos a  $20keV$ . Es interesante notar que las estructura del acero produce una distribución similar a la del Hierro policristalino. Se aprecia como el algoritmo de

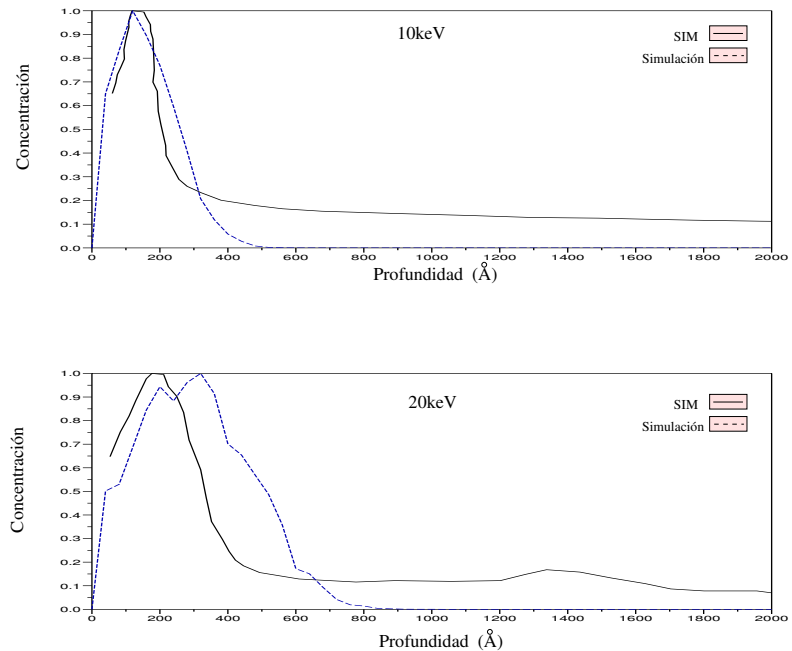


Figura 7.11: Superposición de los perfiles de implantación experimentales con los calculados a  $10$  y  $20keV$ .

simulación provee de muy buenos resultados, pero sin embargo, hay la necesidad de seguir desarrollando el modelo y tener en cuenta algunas otras consideraciones para realizar simulaciones de materiales compuestos, en particular para los aceros.

Como se pudo ver, las simulaciones de DM producen resultados muy próximos a los encontrados experimentalmente, y es de esperarse ya que el blanco real es un poco más complejo y aun hay seguir desarrollando el modelo, de manera, que se pueden considerar los resultados de las simulaciones en la toma de decisiones en nuevos experimentos a otras energías. Con estas simulaciones se obtienen perfiles de implantación para hierro policristalino, pero se puede aumentar el campo de acción para otros materiales y para el rango de altas energías.

## Capítulo 8

# Conclusiones

Se desarrolló un eficiente programa basado en técnicas avanzadas en Dinámica Molecular, las cuales permiten realizar simulaciones de implantación de iones con un considerablemente grande número de partículas.

Se encontró la importancia que tiene la estructura del blanco en la distribución final de los iones. Como se pudo observar de las simulaciones, hay una fuerte dependencia de los perfiles de los iones implantados debido a la red de los átomos del blanco. Esto se debe principalmente a que las trayectorias tienen un mayor desplazamiento libre medio para las estructuras más organizadas donde hay canales de red.

Se encontró algo muy importante en el comportamiento del rango medio proyectado  $R_p$ . Para el caso de un blanco policristalino la dependencia del  $R_p$  tiene un comportamiento lineal en función de la energía inicial de los iones para el rango  $10 - 50keV$ . En este caso se demostró que el material nano-cristalino implantado se comporta como un material amorfo, debido a que el desplazamiento de los iones es muy grande comparado con el tamaño de los granos. Esto sugiere que hay una interesante relación entre el tamaño del grano y la energía de implantación.

Se demostró la importancia de realizar simulaciones basadas en Dinámica Molecular en el campo de la implantación iónica, ya que aquí el blanco es regularmente un material sólido. En este tipo de simulaciones es muy sencillo tener en cuenta la estructura

microscópica de un material que como se ha encontrado es muy importante para estos estudios.

Las simulaciones en Dinámica Molecular son una alternativa para las investigaciones en implantación iónica, siendo una herramienta que provee abundante información que no es accesible por ningún otro medio, además, para este tipo de simulaciones es muy sencillo cambiar las propiedades de las partículas y la estructura del blanco, lo que es muy útil para futuras investigaciones.

El programa resultó ser muy eficiente en el cálculo de los perfiles en un tiempo realmente corto. Se espera seguir avanzando hacia mejores modelos que permitan realizar estudios en materiales más complejos, por lo tanto continua su constante desarrollo.

## ANEXO A

```
// Programado por: Edmanuel E. Torres Amaris
// Programa para realizar simulaciones de
// dinámica molecular de alpha-Hierro policristalino.
// Grupo de Física y Tecnología del Plasma
// Universidad Industrial de Santander (UIS).

void CMDSim::mdTargetTargetNeighbors(int i){
    int j;
#ifdef MD_MSG
    printf("\nEnd mdTargetTargetNeighbors (%i)",i);
#endif
    mdrc = fVAdd(target[i].r,_box_middle);
    mdcc = iVMul(mdrc,mdinvwcell);
    for (int _m=0; _m<14; _m++){
        mdtmp = iVAdd(mdcc,mdNbCells[_m]);
// Free _DIM here with PCB
        for (coord=0; coord<_DIM; coord++){
            //if(_md_pbc[coord])
            if(mdtmp[coord] >= mdcells[coord]){
                mdtmp[coord]= 0;
                shift[coord]= _box_size[coord];
            }else if(mdtmp[coord] < 0){
                mdtmp[coord]= mdcells[coord]-1;
                shift[coord]=-_box_size[coord];
            }else{shift[coord]=0;}
        }
        _icell = iVLinear(mdtmp,mdcells);
        j = mdCellHead[_icell];
        while(1){
            if(j<0) break;
            if(_m!=0 || j>i){
                r2 = 0;
                for (coord=0; coord<_DIM; coord++){
                    dx = target[j].r[coord] - target[i].r[coord];
                    if (dx < -_box_middle[coord]){
                        dx += _box_size[coord];
                    }else if(dx > _box_middle[coord]){
                        dx -= _box_size[coord];
                    }
                    r2 += (dx*dx);
                }
            }
            if(r2 < _ngb_rc2){
                target[i].neighbor.push_back(j);
                //target[j].neighbor.push_back(i);
                if(r2 < _target_rc2){
                    rsqrt = sqrt(r2);
                    ri = (rsqrt-eamp[0][0])*_inv_delta[0];
                    rk = int(ri);
                    if (rk < 0) rk = 0;
                    weight = ri - (double)rk;
                    //ELECTRONIC DENSITY
                    rho = weight* eam[rk+1][2] + (1.0-weight)* eam[rk][2];
                    target[i].n_i += rho;
                    target[j].n_i += rho;
                    // PAIR POTENTIAL
                    pot = weight* eam[rk+1][0] + (1.0-weight)* eam[rk][0];
                    target[i].pot_par += pot;
                    target[j].pot_par += pot;
                }
            }
        }
    }
}
```

```

}
}
j = mdCellList[j];
}
}
//POTENTIAL EMBED
if(target[i].n_i<eamp[1][1]){
ri = (target[i].n_i-eamp[0][1])*_inv_delta[1];
rk = int(ri);
if (rk < 0) rk = 0;
weight = ri - (double)rk;
pot = weight* eam[rk+1][1] + (1.0-weight)* eam[rk][1];
target[i].pot_emb = pot;
}else{target[i].pot_emb = 0;}
#ifdef MD_MSG
printf("\nEnd mdTargetTargetNeighbors");
#endif
}

void CMDSim::mdTargetNeighbors(void){
mdCellsList();
#ifdef MD_MSG
printf("\nEnd mdTargetNeighbors");
#endif
for(unsigned int i=0; i<_NUM_TARGET; i++){
target[i].neighbor.clear();
mdTargetTargetNeighbors(i);
}
#ifdef MD_MSG
printf("\nEnd mdTargetNeighbors");
#endif
}

void CMDSim::mdIonsTargetNeighbors(int i){
int j;
real force, elec_dens = 0, v, Se, v2=0, vp, vi, F;
mdrc = fVAdd(ions[i].r,_box_middle);
mdcc = iVMul(mdrc,mdinvcwcell);
for (int _m=0; _m<27; _m++){ // all 26
mdtmp = iVAdd(mdcc,mdNbCells[_m]);
for (coord=0; coord<_DIM; coord++){
if(_md_pbc[coord])
if(mdtmp[coord] >= mdcells[coord]){
mdtmp[coord]= 0;
shift[coord]= _box_size[coord];
}else if(mdtmp[coord] < 0){
mdtmp[coord]= mdcells[coord]-1;
shift[coord]=-_box_size[coord];
}else{shift[coord]=0;}
}
_icell = iVLinear(mdtmp,mdcells);
if(_icell>=0 && _icell < _NUM_CELLS)
j = mdCellHead[_icell];
else j = -1;
while(1){
if(j<0) break;
r2 = 0;
for (coord=0; coord<_DIM; coord++){
dx = target[j].r[coord] - ions[i].r[coord];
while(dx < -_box_middle[coord])
dx += _box_size[coord];
while(dx > _box_middle[coord])
dx -= _box_size[coord];
rpos[coord]=dx;
r2 += (dx*dx);
}
}
}

```

```

    }
    if(r2 < _ion_shl_rc2){
        ions[i].neighbor.push_back(j);
    }
    if(r2 < _ion_rc2){
        rsqrt = sqrt(r2);
#ifdef _BK_ELEC_STP_
        if(r2 < _target_rc2){
            ri = (rsqrt-eamp[0][0])*_inv_delta[0];
            rk = int(ri);
            if (rk < 0) rk = 0;
            weight = ri - (double)rk;
//ELECTRONIC DENSITY
            elec_dens += weight* eam[rk+1][2] + (1.0-weight)* eam[rk][2];
        }
#endif
        ions[i].pot_scr += mdIonScreenPot(rsqrt);
        force = mdIonScreenForce(rsqrt)/rsqrt;
        for(coord=0; coord<_DIM; coord++){
            F = force*rpos[coord];
            ions[i].a[coord] += (F*_am_im); //ions[i].a[coord] += (F*mdGetTargetAtomMass())/mdGetIonsAtomMass();
            target[j].a[coord] -= F;
        }
    }
}
j = mdCellList[j];
}
}
#ifdef _BK_ELEC_STP_
for(j=0; j<(int)_DIM; j++){
    v2 += (ions[i].v[j]*ions[i].v[j]);
    ions[i].energ=v2;
    v = sqrt( 2.0*(v2*K_B)/(_ions_mass*AMU) );
    vi = v;
    vp = vi*1e-6;
    Se = mdGetIonElectronicStop(elec_dens, vi , vp);
    for(j=0; j<(int)_DIM; j++){
        ions[i]._e_stp[j] = (Se * ions[i].v[j])/vi;
    }
    ions[i].a[j]+=ions[i]._e_stp[j];
}
#endif
}

void CMDSim::mdIonsNeighbors(void){
    mdCellsList();
    for(unsigned int i=0; i<_NUM_IONS; i++){
        if(ions[i]._sw){
            ions[i].neighbor.clear();
            mdIonsTargetNeighbors(i);
        }
    }
    //cout<<endl;
}

real CMDSim::mdGetTargetElectronicDensity(real _r){
    double _rho_;
    _rho_ = ( ((real)_ions_atom_num-1.0) / (4.0 * PI * _r * pow(_target_lamb,2.0) ))\
    * exp(- _r / _target_lamb);
    return _rho_;
}

real CMDSim::mdGetIonElectronicStop(real _rho, real _vi, real _vp){
    double yr, q, lamb_i, gamma, Zeff, Sp, Se, Vr, d, rs, _Vf;
    d = 3.0/(4.0*PI*_rho);
    rs = pow(d,p1);
}

```

```

        _Vf = k*pow((3.0*PI*PI*_rho*1e30),p1); ///  

        if( _vi >= _Vf)  

Vr = _vi * ( 1.0 + 1.0/5.0 * pow(_Vf/_vi,2.0) );  

        else  

Vr = (3.0*_Vf/4.0) * ( 1.0 + ( (2.0/3.0) * pow(_vi/_Vf,2.0) )\  

- (1.0/15.0)* pow(_vi/_Vf,4.0) );  

        yr = Vr / (Vo * pow((double)_ions_atom_num,p2) );  

        q = 1.0-exp(-0.95*(yr - 0.07));  

        lamb_i = (2.0 * ao * pow(1.0-q,2.0/3.0) ) / \  

( pow((double)_ions_atom_num,1.0/3.0) * (1.0-(1.0-q)/7.0));  

        gamma = q + C*(1.0-q) * log( 1.0 + pow((4.0*lamb_i)/rs,2.0));  

        Zeff = gamma * (double)_ions_atom_num;  

        Sp = (7*1e23/1.013e23)*(7.3*_vp*(1.0-0.076*_vp));  

        Se = - Sp*(Zeff*Zeff)*eV/K_B; // _epsilon/A  

        return Se;  

}

void CMDSim::mdIonsScreenConst(void){  

    a_u = (0.8854 * A_0 )/(pow((double)mdGetTargetAtomicNum()\  

0.23)+pow((double)mdGetIonsAtomNum(),0.23));  

    a_u_0= (0.8854 * A_0 )/(pow((double)mdGetTargetAtomicNum(),0.23)\  

+pow((double)mdGetIonsAtomNum(),0.23));  

    k_p = (double(mdGetTargetAtomicNum())\  

*double(mdGetIonsAtomNum())*E*E)/(4.0*PI*E_0*K_B);  

}

// ZBL repulsive potential --- V(r) = (1/(4*PI*E_0)) * ((Z_1*Z_2)/r) * phi(r)  

real CMDSim::mdIonScreenPot(real r){  

    double Vr, phi, x;  

    x = r/a_u;  

    phi = 0.18175*exp(-3.1998*x)+0.50986*exp(-0.94229*x)+0.28022*\  

exp(-0.40290*x)+0.028171*exp(-0.20162*x);  

    Vr = (k_p * phi)/r;  

    return Vr;  

}

void CMDSim::mdScreenPot(void){  

    unsigned int i;  

    double du;  

    scrpot.resize(_tabledat);  

    scrforc.resize(_tabledat);  

    smax=_ion_rc*1e-10;  

    smin=1e-12;  

    _sa = 0.8*smax;  

    _sb = smax;  

    sdelt = (smax-smin)/(_tabledat-1);  

    for(i=0; i<_tabledat; i++)  

scrpot[i]=mdIonScreenPot(smin+double(i)*sdelt);  

    smax=_ion_rc;  

    smin=1e-2;  

    sdelt = (smax-smin)/(_tabledat-1);  

    for(i=0; i<2; i++){  

du=(2.0*scrpot[i+3]-9.0*scrpot[i+2]+18.0*scrpot[i+1]-11.0*scrpot[i])/(6.0*sdelt);  

scrforc[i]=du;  

    }  

    for(i=2; i<(_tabledat-2); i++){  

du=(scrpot[i-2]-8.0*scrpot[i-1]+8.0*scrpot[i+1]-scrpot[i+2])/(12.0*sdelt);  

scrforc[i]=du;  

    }  

    for(i=_tabledat-2; i<_tabledat; i++){  

du=(11.0*scrpot[i]-18.0*scrpot[i-1]+9.0*scrpot[i-2]-2.0*scrpot[i-3])/(6.0*sdelt);  

scrforc[i]=du;  

    }  

    for(i=0; i<_tabledat; i++){  

du=(smin+double(i)*sdelt)*1e-10;  


```

```

if(du>_sa) scrpot[i]*=mdIonScreenSpline(du);
    }
}

double CMDSim::mdIonScreenSpline(double _r){
    real S;
    S = ((2.0*_r-3.0*_sa+_sb)*(_r-_sb)*(_r-_sb))/pow((_sb-_sa),3.0);
    return S;
}

// dV(r)/r = ((Z_1*Z_2)/(4*Pi*E_0)) * [r^(-2) * phi(r)] + [r^(-1) * Phi(r)]
double CMDSim::mdIonScreenForce(double _r){
    double Fr, ri, weight;//, phi, dphi, x=0;
    int rk;
    ri = (_r-smin)/sdelt;
    rk = int(ri);
    if (rk < 0) rk = 0;
    weight = ri - (double)rk;
    Fr = (weight* scrforc[rk+1] + (1.0-weight)* scrforc[rk]);
    return Fr;
}

void CMDSim::mdTargetTargetEmbEner(int i){
    for(unsigned int j=0; j<target[i].neighbor.size(); j++){
    r2 = 0;
    for (coord=0; coord<_DIM; coord++){
        dx = target[target[i].neighbor[j]].r[coord] - target[i].r[coord];
        if (dx < -_box_middle[coord]){
            dx += _box_size[coord];
        }else if(dx > _box_middle[coord]){
            dx -= _box_size[coord];
        }
        r2 += (dx*dx);
    }
    if(r2 < _target_rc2){
        rsqrt = sqrt(r2);
        ri = (rsqrt-eamp[0][0])*_inv_delta[0];
        rk = int(ri);
        if (rk < 0) rk = 0;
        weight = ri - (double)rk;
        //POTENTIAL DENS
        rho = weight* eam[rk+1][2] + (1.0-weight)* eam[rk][2];
        target[i].n_i += rho;
        target[target[i].neighbor[j]].n_i += rho;
        //POTENTIAL PAIR
        pot = weight* eam[rk+1][0] + (1.0-weight)* eam[rk][0];
        target[i].pot_par += pot;
        target[target[i].neighbor[j]].pot_par += pot;
    }
    }
    //POTENTIAL EMBED
    if(target[i].n_i<eamp[1][1]){
    ri = (target[i].n_i-eamp[0][1])*_inv_delta[1];
    rk = int(ri);
    if (rk < 0) rk = 0;
    weight = ri - (double)rk;
        pot = weight* eam[rk+1][1] + (1.0-weight)* eam[rk][1];
    target[i].pot_emb = pot;
        }else{target[i].pot_emb = 0;}
    }

void CMDSim::mdTargetEmbEner(void){
#ifdef MD_MSG
    printf("\nBegin mdTargetEmbEner");
#endif
}

```

```

        for(unsigned int i=0; i<_NUM_TARGET; i++)
mdTargetTargetEmbEner(i);
#ifdef MD_MSG
    printf("\nBegin mdTargetEmbEner");
#endif
}

void CMDSim::mdComputeForce(int i){
    if(target[i].n_i<eamp[1][1]){
        ri = (target[i].n_i-eamp[0][1])*_inv_delta[1];
rk = int(ri);
if (rk < 0) rk = 0;
weight = ri - (double)rk;
dpot = weight* deam[rk+1][1] + (1.0-weight)* deam[rk][1];
dpot_emb_i = dpot;
    }else{dpot_emb_i = 0;}
    for(unsigned int j=0; j<target[i].neighbor.size(); j++){
f=0;
r2 = 0;
for (coord=0; coord<_DIM; coord++){
    dx = target[target[i].neighbor[j]].r[coord] - target[i].r[coord];
    if (dx < -_box_middle[coord])
dx += _box_size[coord];
    else if (dx > _box_middle[coord])
dx -= _box_size[coord];
    rpos[coord]=dx;
    r2 += dx*dx;
}
if(r2 < _target_rc2){
    rsqrt = sqrt(r2);
    ri = (rsqrt-eamp[0][0])*_inv_delta[0];
    rk = int(ri);
    if (rk < 0) rk = 0;
    weight = ri - (double)rk;
    drho = weight* deam[rk+1][2] + (1.0-weight)* deam[rk][2];
    dn_i = drho;
    dpot = weight* deam[rk+1][0] + (1.0-weight)* deam[rk][0];
    dpot_par = dpot;
    if((target[target[i].neighbor[j]].n_i)<eamp[1][1]){
ri = (target[target[i].neighbor[j]].n_i-eamp[0][1])*_inv_delta[1];
rk = int(ri);
if (rk < 0) rk = 0;
weight = ri - (double)rk;
dpot = weight* deam[rk+1][1] + (1.0-weight)* deam[rk][1];
dpot_emb_j = dpot;
    }else{dpot_emb_j = 0;}
    f = (dpot_par + (dpot_emb_i + dpot_emb_j) * dn_i)/rsqrt;
    for(unsigned int k=0; k<_DIM; k++){
dpot = rpos[k]*f;
target[i].a[k] += dpot;
target[target[i].neighbor[j]].a[k] -= dpot;
    }
}
}

void CMDSim::mdComputeForces(void){
#ifdef MD_MSG
    printf("\nBegin mdComputeForces");
#endif
for(unsigned int i=0; i<_NUM_TARGET; i++){
mdComputeForce(i);
}
#ifdef MD_MSG
    printf("\nBegin mdComputeForces");
#endif
}

```

```

#endif
}

void CMDSim::mdIonsTargetForce(int i){
    real force, elec_dens = 0, v, Se, v2=0, vp, vi, F;
    unsigned int j;
    for(j=0; j<ions[i].neighbor.size(); j++){
        //f=0;
        r2 = 0;
        for (coord=0; coord<_DIM; coord++){
            dx = target[ions[i].neighbor[j]].r[coord] - ions[i].r[coord];
            if (dx < -_box_middle[coord])
                dx += _box_size[coord];
            else if (dx > _box_middle[coord])
                dx -= _box_size[coord];
            rpos[coord]=dx;
            r2 += dx*dx;
        }
        if(r2 < _ion_rc2){
            ions[i].neighbor.push_back(j);
            rsqrt = sqrt(r2);
#ifdef _BK_ELEC_STP_
            if(r2 < _target_rc2){
                ri = (rsqrt-eamp[0][0])*_inv_delta[0];
                rk = int(ri);
                if (rk < 0) rk = 0;
                weight = ri - (double)rk;
                //ELECTRONIC DENSITY
                elec_dens += weight* eam[rk+1][2] + (1.0-weight)* eam[rk][2];
            }
#endif
            ions[i].pot_scr += mdIonScreenPot(rsqrt);
            force = mdIonScreenForce(rsqrt)/rsqrt;
            for(coord=0; coord<_DIM; coord++){
                F = force*rpos[coord];
                ions[i].a[coord] += (F*_am_im);
                target[j].a[coord] -= F;
            }
        }
    }

#ifdef _BK_ELEC_STP_
    for(j=0; j<_DIM; j++){
        v2 += (ions[i].v[j]*ions[i].v[j]);
        ions[i].energ=v2;
        v = sqrt( 2.0*(v2*K_B)/(_ions_mass*AMU) );
        vi = v;
        vp = vi*1e-6;
        Se = mdGetIonElectronicStop(elec_dens, vi , vp);
        for(j=0; j<_DIM; j++){
            ions[i]._e_stp[j] = (Se * ions[i].v[j])/vi;
        }
        ions[i].a[j]+=ions[i]._e_stp[j];
    }
#endif
}

void CMDSim::mdIonsForce(void){
    for(unsigned int i=0; i<_NUM_IONS; i++)
        if(ions[i]._sw) mdIonsTargetForce(i);
}

void CMDSim::mdComputeStartMotionTarget(void){
    unsigned int i, j;
    real x;
#ifdef MD_MSG
    printf("\nBegin mdComputeStartMotionTarget");
#endif
}

```

```

#endif
    mdResetParticles();
#ifdef _MD_TARGET_
    mdTargetNeighbors();
    mdComputeForces();
#endif
#ifdef _MD_IONS_
    mdIonsNeighbors();
#endif
    for(i=0; i<_NUM_TARGET; i++){
if(target[i]._sw){
    for(j=0; j<_DIM; j++){
target[i].r[j] += (target[i].v[j]*_time_step)+0.5*target[i].a[j]*_time_step*_time_step;
    while(target[i].r[j] < -_box_middle[j]) target[i].r[j] += _box_size[j];
    while(target[i].r[j] > _box_middle[j]) target[i].r[j] -= _box_size[j];
target[i].v[j] += 0.5*target[i].a[j]*_time_step;
    }
}
}

#ifdef _MD_IONS_
    for(i=0; i<ions.size(); i++){
for(j=0; j<_DIM; j++){
x = (ions[i].v[j]*mdGetTimeStep()+0.5*ions[i].a[j]*_time_step*_time_step;
    ions[i].r[j] += x;
    if(_md_pbc[j]){
if (ions[i].r[j] < -_box_middle[j])
    ions[i].r[j] += _box_size[j];
if (ions[i].r[j] > _box_middle[j])
    ions[i].r[j] -= _box_size[j];
    }
    ions[i].v[j] = (ions[i].v[j] + 0.5*ions[i].a[j]*_time_step);
    ions[i]._rg[j] += x;
}
}
    //mdAdjustIonsTemp();
#endif

#ifdef _MD_TARGET_
    mdAdjustTargetTemp();
#endif
mdResetParticles();
#ifdef _MD_TARGET_
    mdTargetNeighbors();
    mdComputeForces();
#endif
#ifdef _MD_IONS_
    mdIonsNeighbors();
#endif
}

// B. Smith and Harrison (Computer in Physics 3 (1989) - 68)
void CMDSim::mdComputeMotionTarget(void){
    unsigned int i, j;
    real x;
#ifdef MD_MSG
    printf("\nBegin mdComputeMotionTarget");
#endif
#ifdef _MD_TARGET_
    for(i=0; i<_NUM_TARGET; i++){
if(target[i]._sw){
    for(j=0; j<_DIM; j++){
target[i].r[j] += (target[i].v[j]*_time_step)\
+ (4.0*target[i].a[j]-target[i]._a[j])*_tt_6;
while(target[i].r[j] < -_box_middle[j])

```

```

        target[i].r[j] += _box_size[j];
while(target[i].r[j] > _box_middle[j])
    target[i].r[j] -= _box_size[j];
target[i].v[j] += (8.0*target[i].a[j]-target[i]._a[j])*_t_12;
    }
}
    mdAdjustTargetTemp();
#endif
#ifdef _MD_IONS_
    for(i=0; i<_NUM_IONS; i++){
if(ions[i]._sw){
    for(j=0; j<_DIM; j++){
x = (ions[i].v[j]*_time_step)+(4.0*ions[i].a[j]-ions[i]._a[j])*_tt_6;
ions[i].r[j] += x;
if(_md_pbc[j]){
    while(ions[i].r[j] < -_box_middle[j])
ions[i].r[j] += _box_size[j];
    while(ions[i].r[j] > _box_middle[j])
ions[i].r[j] -= _box_size[j];
}
#ifdef _BK_ELEC_STP_MOT_
//ions[i].v[j] += (8.0*ions[i].a[j]-ions[i]._a[j])*_t_12 +\
ions[i]._e_stp[j]*_t_12;//_time_step;
#else
ions[i].v[j] += (8.0*ions[i].a[j]-ions[i]._a[j])*_t_12;
#endif
ions[i]._rg[j] += x;
//cout<<"X["<<j<<"]="<<ions[i].proy[j]<<endl;
    }
}
}
#endif
    mdResetParticles();

    if(_dis_stp>0.5*_ngb_shll) _do_ngb=1;
    if(_dis_istp>0.5*_ingb_shll) _do_ingb=1;

#ifdef _MD_TARGET_
    if(_do_ngb){
mdTargetNeighbors();
_do_ngb = 0;
_dis_stp= 0;
    }else{
mdTargetEmbEner();
    }
    mdComputeForces();
#endif

#ifdef _MD_IONS_
#ifdef _MD_IONS_NEIGH_
    if(_do_ingb){
mdIonsNeighbors();
_do_ingb = 0;
_dis_istp= 0;
    }else{
mdIonsForce();
    }
#else
    mdIonsNeighbors();
#endif
#endif
#endif

#ifdef _MD_TARGET_
    _maxvv = 0;

```

```

        for(i=0; i<_NUM_TARGET; i++){
//if(target[i]._sw){
    _vv=0;
    for(j=0; j<_DIM; j++){
target[i].v[j] += (5.0*target[i].a[j])*_t_12;
_vv += target[i].v[j]*target[i].v[j];
    }
    _maxvv = max(_maxvv, _vv);
//}
    }
    _dis_stp += sqrt(_maxvv)*_time_step;
    _MXV = sqrt(_maxvv);
#endif

#ifdef _MD_IONS_
    _maxvv = 0;
    for(i=0; i<_NUM_IONS; i++){
if(ions[i]._sw){
    _vv=0;
    for(j=0; j<_DIM; j++){
#ifdef _BK_ELEC_STP_MOT_
ions[i].v[j] += (5.0*ions[i].a[j])*_t_12 + ions[i]._e_stp[j]*_t_12;
#else
ions[i].v[j] += (5.0*ions[i].a[j])*_t_12;
#endif
    }
    _vv += (ions[i].v[j]*ions[i].v[j]);
    }
    _maxvv = max(_maxvv, _vv);
}
    }
    _dis_istp += sqrt(_maxvv)*_time_step;
#endif

#if defined(_MD_TARGET_) && _MD_IONS_
    _MXV = max(_MXV, sqrt(_maxvv));
#else
    _MXV = sqrt(_maxvv);
    printf("\n Only Atoms or Ions...!");
#endif

#ifdef _TIME_STEP_
    mdSetTimeStep(0.1/_MXV);
#endif
    mdAdjustTargetTemp();
#ifdef MD_MSG
    printf("\nEnd mdComputeMotionTarget");
#endif
}

void CMDSim::mdEvalProp(void){
    real _vx=0, _vy=0, _vz=0, _ke, _ue, _temp;
    for(unsigned int i=0; i<_NUM_TARGET; i++){
_vx += target[i].v[0];
_vy += target[i].v[1];
_vz += target[i].v[2];
    }
    _ue = mdEvalSysUEnergy();
    _ke = mdEvalSysKEnergy();
    _temp = mdEvalSysTemp();
#ifdef _MD_DEBUG_ENER
    printf("Energy=%f\n", _ke+_ue);
    printf("Temp=%f\n", _temp);
#endif
}

```

```

real CMDSim::mdEvalSysKEnergy(void){
    real _k_ener;
    _k_sys_ener = 0;
    for(unsigned int i=0; i<_NUM_IONS; i++){
if(ions[i]._sw){
    _k_ener = 0;
        for(unsigned int j=0; j<_DIM; j++)
_k_ener += (ions[i].v[j]*ions[i].v[j]);
        _k_ener *= (0.5*mdGetIonsAtomMass()/mdGetTargetAtomMass());
        if(_k_ener<K_0 && _num_ions_off<_NUM_IONS){
ions[i]._sw=false;
_num_ions_off++;
        }
        _k_sys_ener += _k_ener;
    }
    }
    _k_sys_ener *= (K_B/eV);
    //if(_k_sys_ener==0)
    //mdWriteRange("range.xyz", 6);
    return _k_sys_ener/(_NUM_IONS-_num_ions_off);
}

real CMDSim::mdEvalSysTemp(void){
    real sysTemp=mdEvalSysKEnergy();
    return (sysTemp/3.0);
    //return (sysTemp*1.3806503e-23)/(_NUM_TARGET*1.60217733e-19);
}

```

# Bibliografía

- [1] K. Nordlund, *Molecular Dynamics Simulations of Atomic Collisions for Ion Irradiation Experiments*, Thesis Doctoral, Department of Physics (Accelerator Laboratory), University of Helsinki, 1995.
- [2] J. F. Ziegler, *Handbook of Ion Implantation Technology*, (Elsevier Science Publishers, Amsterdam, 1992).
- [3] M. A. Lieberman, A. J. Lichtenberg, *Principles of Plasmas, Discharges and Material Processing*, (Elsevier Science Publishers, Amsterdam, 1992).
- [4] J. R. Conrad, *Plasma source ion implantation: a new approach to ion beam modification of materials*, mater Sci. Eng. 116A (1987) 197-203.
- [5] V. I. Khvesyuk, P. A. Tsygankov, *The use a high-voltage discharge at low pressure for 3D ion implantation*, Surface & Coatings Technology 96 (1997) 68-74.
- [6] V. D. Dougar-Jabon, J. Dulce Moreno, P. A. Tsygankov, *Review of Scientific Instruments*, 73 (2004) 828.
- [7] D. W. Heermann, *Computer Simulation Methods*, (Springer-Verlag, New York, 1990).
- [8] M. T. Robinson and I. M. Torrens, *Computer Simulation of atomic-displacement cascades in solids in the binary-collision approximation*, Phys. Rev. B 9, 5008 (1974).

- [9] S. J. Zhou, D. M. Beazley, P. S. Lomdahl, and B. L. Holian, *Large-Scale Molecular Dynamics Simulations of Three-Dimensional Ductile Failure*, Phys. Rev. Lett. 78 (1997) 479-482. 3.
- [10] J. Sillanpää, *Phenomenological model for electronic stopping of low-velocity ions in crystalline solids*, Thesis Doctoral, Department of Physics (Accelerator Laboratory), University of Helsinki, 2000.
- [11] K. Nordlund, *Molecular dynamics simulation of ion ranges in the 1-100 keV energy range*, Comp. Mat. Sci. 3, 448 (1995).
- [12] K. Nordlund, *Molecular Dynamics Simulations of Atomic Collisions for Ion Irradiation Experiments*, PhD Thesis, Department of Physics (Accelerator Laboratory), University of Helsinki, 1995.
- [13] Y. V. Gott, *Interaction of Particles with Substances in Plasma Research*, Moscú, 1978.
- [14] E. Torres, *Estudio de la Implantación de Iones de Nitrógeno en  $\alpha$ -Hierro en Aproximación de Colisiones Elásticas*, Tesis, Universidad Industrial de Santander, Bucaramanga, 2002.
- [15] Sillanpää Jussi and Nordlund Kai and Keinonen J., *Electronic stopping of Silicon from a 3D Charge Distribution*, Phys. Rev. B, 62 3109 (2000).
- [16] Chimi Y., Iwase A., Ishikawa N., Kudora N., Kambara T., *Radiation annealing induced by electronic excitation in iron*, Nucl. Intr. and Meth. in Phys. Res. B 164-165, (2000) 408-414.
- [17] Burcham W. E., *Elements of nuclear physics*, (Longman, London, p. 76, 1979).
- [18] Brandt W. and Kitagawa M., *Effective stopping-power charges of swift ions in condensed matter*, Phys. Rev. B 25, 5631 (1982).
- [19] Cai David, Grønbech-Jensen Niels, Snell Charles, Beardmore Keith, *Phenomenological electronic stopping-power model for molecular dynamics and Monte Carlo simulation of ion implantation into silicon*, Phys. Rev. B, 54, 17147 (1996).

- [20] Ziegler J. F., Biersack J. P. and Littmark U., *The Stopping and Range of Ions in Matter* (Pergamon, New York, 1985).
- [21] Bethe A. Hans and Raman W. Jackiw, *Intermediate Quantum Mechanics*, (W. A. Benjamin, Inc., New York , 1968).
- [22] Born Max and Oppenheimer J. Roberth, *Ann. Phys.* 84 457 (1927).
- [23] Ercolessi Furio, *A molecular dynamics primer*, (International School for Advanced Studies, Trieste, 1997).
- [24] D. C. Rapaport. *The Art of Molecular Dynamics Simulation* Cambridge, United Kingdom, 2004.
- [25] Alonso Marcelo, Finn Edward J., *Fundamentos Cuánticos y Estadísticos*, (Fondo Educativo Interamericano S. A., Mexico, 1976).
- [26] Daw M. S. and Baskes M. I., *Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals*, *Phys. Rev. B* 29, 6443 (1984).
- [27] F. Jensen. *Introduction to Computational Chemistry*, Wiley, United Kingdom, 1999.
- [28] D. Frenkel, B. Smit, *Understanding Molecular Simulation*, (Academic Press, Boston, 1996).
- [29] Beardmore Keith and Grønbech-Jensen Niels. *Efficient molecular dynamics scheme for the calculation of dopant profiles due to ion implantation*, *Phys. Rev. E*, 57, 7278 (1997).
- [30] Landau Rubin H. and Paéz Manuel J., *Computational Physics, Problem Solving with Computers*, (Jhon Wiley & Sons, Inc., USA, 1997).
- [31] Aoki Takaaki, *Molecular Dynamics Simulation of Cluster Ion Impact on Solid Surface*, Doctoral Thesis, Kyoto University, Japan, 2000.
- [32] Verlet L., *Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*, *Phys. Rev.* 159 98 (1967).

- [33] Smith Richard J. and Harrison Don E., *Algorithms for molecular dynamics simulations of keV particle bombardment*, Comput. in Phys. 3 68 (1989).