

Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Implementación computacional del modelo armónico
de un potencial tipo *Tight Binding* para el grafeno

Autor: Pedro García Saavedra

Tutor: Pilar Ariza Moreno

Dpto. de Mecánica de Medios Continuos y
Teoría de Estructuras
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

**Implementación computacional del modelo
armónico de un potencial tipo *Tight Binding* para
el grafeno**

Autor:

Pedro García Saavedra

Tutor:

Pilar Ariza Moreno

Catedrático de Universidad

Dpto. de Mecánica de Medios Continuos y Teoría de Estructuras

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado: Implementación computacional del modelo armónico de un potencial tipo Tight Binding para el grafeno

Autor: Pedro García Saavedra

Tutor: Pilar Ariza Moreno

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A mi familia

A mi segunda familia

Agradecimientos

Quiero dar las gracias a mis familiares y amigos por siempre animarme a seguir adelante y mostrarme las distintas formas de afrontar los problemas personales y académicos a lo largo de estos años. Gracias a mis compañeros por enseñarme lo que hay más allá de los libros durante todos los interminables días compartidos frente a éstos. Agredecer a mi tutora Pilar toda la paciencia y compromiso que me ha mostrado a lo largo de todo el proceso de este proyecto. Pero sobre todo agradecer a mi madre por días y días de recordarme el valor de las cosas que hemos y no hemos conseguido.

Pedro García Saavedra

Sevilla, 2019

Resumen

El presente trabajo tiene como objetivo la definición y resolución computacional de un modelo armónico de una red de grafeno del tipo *Tight Binding* en el entorno MATLAB a partir de uno ya definido en lenguaje Mathematica.

Previo planteamiento teórico del modelo, se plasman las diferentes soluciones adoptadas para poder expresar las ecuaciones que conforman el sistema sin afrontar pérdida de información a lo largo del proceso de resolución, los resultados alcanzados y las diferencias entre ambos programas en su ejecución.

Abstract

The present project seeks to achieve the computational definition and resolution of a harmonic model of a Tight Binding-type graphene lattice in the MATLAB environment starting by an already defined model in the Mathematica coding language.

Firstly, a theoretical approach is carried out, then all the different solutions assumed in order to express the model equations without any information lost during the solving process are shown. Thirdly, the achieved results are demonstrated and finally it is provided an explanation of the differences between both softwares throughout its execution.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Tablas	xvi
Índice de Figuras	xviii
Notación	xx
1 Introducción	1
1.1 <i>Grafeno y su historia</i>	1
1.2 <i>Descripción y propiedades del grafeno</i>	3
1.3 <i>Objeto y motivo del trabajo</i>	5
2 Complejo Simplicial del Grafeno	7
3 Modelo Teórico	12
3.1 <i>Potenciales interatómicos. Definición y contexto histórico</i>	12
3.2 <i>Principios físicos del Tight Binding</i>	14
3.2 <i>Modelo Tight Binding</i>	15
4 Aplicación Práctica	20
4.1 <i>Introducción a la simulación computacional</i>	20
4.2 <i>Descripción del entorno Mathematica</i>	21
4.3 <i>Descripción del entorno MATLAB</i>	22
4.4 <i>Soluciones a las diferencias de sintaxis</i>	25
4.5 <i>Proceso de resolución</i>	25
4.6 <i>Resultados</i>	28
5 Conclusiones y trabajos futuros	33
5.1 <i>Análisis de los resultados</i>	33
5.2 <i>Trabajos futuros</i>	33
Apéndice A. Transformada de Fourier	36
A.1 <i>Definición matemática</i>	36
A.2 <i>Propiedades fundamentales</i>	36
A.3 <i>Funciones periódicas</i>	36
Bibliografía	37

ÍNDICE DE TABLAS

Tabla 1. Comparativa entre las propiedades físicas de grafeno, acero y aluminio	3
Tabla 2. Parámetros <i>On-Site</i>	17
Tabla 3. Coeficientes para $S(r_{ij})$	17
Tabla 4. Coeficientes de f	18
Tabla 5. Parámetros para $\Phi(r_{ij})$	18
Tabla 6. Tiempos de resolución en segundos	31

ÍNDICE DE FIGURAS

Figura 1. <i>Timeline</i> del grafeno desde 1560 hasta 2017	1
Figura 2. Tipos de hibridación del carbono	2
Figura 3. Alótropos del carbono	2
Figura 4. Red de grafeno	3
Figura 5. Prototipos de <i>smartphones</i> flexibles basados en grafeno	4
Figura 6. Exfoliación mecánica del grafeno	4
Figura 7. Celdas-p. Tipos de átomos, enlaces y celdas de una red de grafeno	8
Figura 8. Red de Bravais del grafeno. Traslaciones de átomos, enlaces y celdas	8
Figura 9. Orientación de las celdas de la red	9
Figura 10. Compromiso entre tiempo computacional y precisión de los modelos interatómicos	13
Figura 11. Átomos vecinos en una red de grafeno (átomos tipo 1: blanco, átomos tipo 2: negro)	15
Figura 12. Algoritmo para la resolución y mejora de un modelo científico	21
Figura 13. Logo Mathematica	21
Figura 14. Disposición sobre una misma interfaz de texto, <i>script</i> matemático y gráficas	22
Figura 15. Logo de MATLAB	23
Figura 16. <i>Apps</i> disponibles en MATLAB 2015	23
Figura 17. Captura de la ventana de comandos	24
Figura 18. Captura del editor de código	24
Figura 19. Diagrama de flujo correspondiente al proceso de resolución	26
Figura 20. Croquis del redimensionamiento de la matriz Ψ_{ij}	27
Figura 21. Relación entre índices por medio del Teorema de Tales	27
Figura 22. Matriz Φ_{ij} en Mathematica para $\theta_1 = 0.5$, $\theta_2 = 0.25$	28
Figura 23. Matriz Φ_{ij} en MATLAB para $\theta_1 = 0.5$, $\theta_2 = 0.25$	28
Figura 24. Representación de la parte real de Φ_{ij} para $\theta_1 = 0.5$, $\theta_2 = 0.25$ en ambos programas	29
Figura 25. Representación de la parte imaginaria de Φ_{ij} para $\theta_1 = 0.5$, $\theta_2 = 0.25$ en ambos programas	29
Figura 26. Vista lateral. Mostrando los resultados para los valores reales con una x constante	30
Figura 27. Vista lateral. Mostrando los resultados para los valores reales con una y constante	30
Figura 28. Tiempos de resolución, en segundos, de ambos métodos	31

Notación

e	número e
CVD	Deposición Química de Vapor
DFT	Transformada Directa de Fourier
FCC	Cúbica centrada en las caras
MPa	Megapascales
TPa	Terapascales
W/mK	Vatios por metro Kelvin
S	Ohmio a la menos 1 o Siemens
$S \cdot m^{-1}$	Siemens por metro
\mathbb{Z}	Conjunto de números enteros
\mathbb{R}	Conjunto de números reales
<i>html</i>	Hypertext Markup Language

1 INTRODUCCIÓN

*Si has construido castillos en el aire,
tu trabajo no se pierde;
ahora coloca las bases debajo de ellos
-Henry David Thoreau -*

A continuación se van a exponer algunos conceptos e ideas básicas relacionadas con el presente trabajo, desde la definición del material sobre el que se centra el estudio, **grafeno**, hasta la descripción del modelo seleccionado, pasando por la filosofía y técnicas empleadas para abordarlo. Dicha descripción será muy somera dado el carácter introductorio y sin vocación académica de este capítulo.

1.1 Grafeno y su historia

Pese a ser motivo de estudio desde el siglo pasado, la palabra grafeno no empezó a sonar con fuerza hasta que en 2004 Novoselov y Geim encontraron que, tras un proceso consecutivo de exfoliación mecánica sobre un bloque de grafito, era posible aislar una película de espesor monoatómico cuyas propiedades supondrían una revolución en el campo de la nanotecnología en la última década.

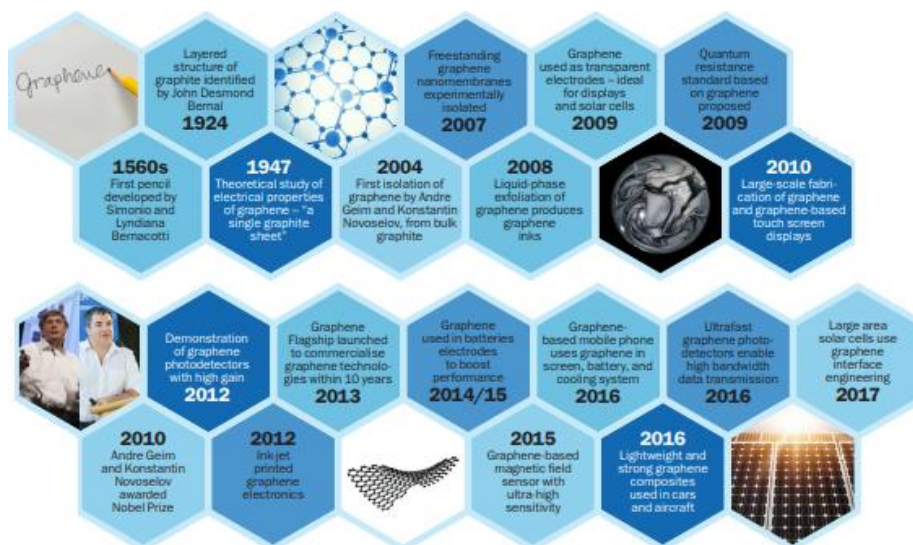


Figura 1. *Timeline* del grafeno desde 1560 hasta 2017

Las potenciales propiedades que se le han estado atribuyendo hasta la fecha son cuanto menos motivo de curiosidad en la comunidad científica y cada vez más en el mundo ingenieril. Sus futuribles aplicaciones en tan diversos campos como la electrónica, termomecánica o materiales compuestos, hacen del grafeno uno de los principales objetos de investigación mundial.

Es de recibo identificar previamente al **carbono**, elemento químico primordial del grafeno, como responsable de gran parte de sus propiedades. Motivo principal de muchas de estas fascinantes cualidades es su estructura electrónica ($1s^2 2s^2 2p^2$ para el estado basal), que origina una vasta variedad de formas alotrópicas aparte del propio grafeno, entre ellas:

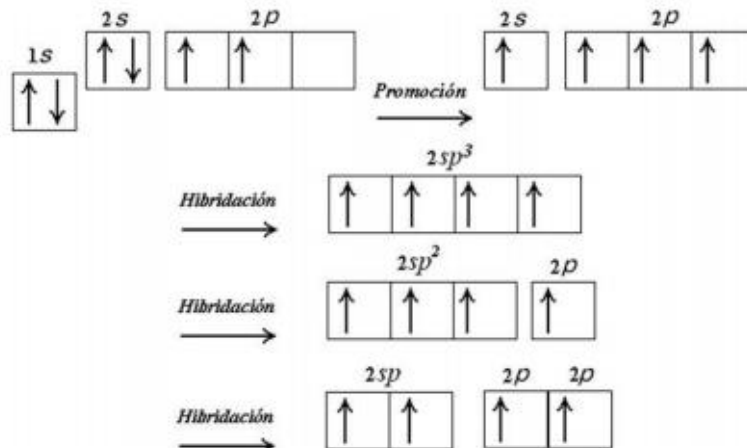


Figura 2. Tipos de hibridación del carbono

- *Grafito*: sólido consistente en una consecución de láminas bidimensionales de patrón hexagonal unidas mediante enlaces dobles. Las uniones entre capas son realizadas por Fuerzas de Van der Waals, muy inferiores a los enlaces covalentes formados sobre cada una de las capas de grafito. Al separar en repetidas ocasiones estas capas de sus vecinas se obtiene, eventualmente, una lámina de grafeno.
- *Nanotubos de carbono*: alótropo de estructura tubular conformada por láminas de grafito enrolladas sobre sí mismas.
- *Fulerenos*: estructuras en forma esférica compuesta por celdas tanto hexagonales y como pentagonales, siendo sus vértices átomos de carbono. Ejemplos son el C_{60} , C_{70} o C_{540} .
- *Diamante*: alótropo cristalino de red cúbica centrada en las caras (FCC) lo cual le proporciona una serie de propiedades físicas sin igual. Dureza, tenacidad y conductividad térmica son algunos ejemplos.

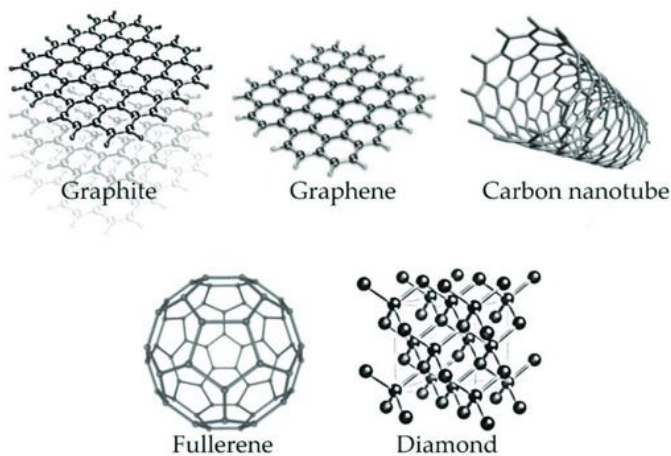


Figura 3. Alótropos del carbono

1.2 Descripción y propiedades del grafeno

Como ya se ha comentado, la estructura del carbono facilita el origen de distintos compuestos; esto se da gracias al proceso de **hibridación** entre orbitales atómicos. En el caso del grafeno, el resultado es un material estrictamente bidimensional en el que los tres enlaces híbridos de tipo sp^2 permanecen en un plano formando un ángulo de 120° , dando lugar así a una red hexagonal plana de un átomo de grosor.

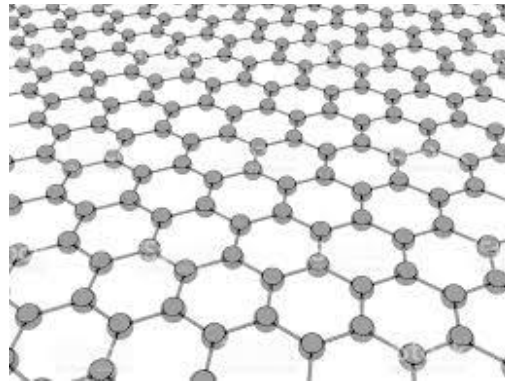


Figura 4. Red de grafeno

Son precisamente esta estructura bidimensional y las conexiones interatómicas que en ella se producen las causantes de unas peculiares, aunque útiles, propiedades tanto mecánicas (resistencia, rigidez y flexibilidad entre otras) como eléctricas, siendo el material más resistente medido hasta la fecha.

Algunas de dichas cualidades en las que destaca son aplastantemente superiores a las de otros materiales habitualmente empleados en la industria como son el acero y el aluminio. Es posible apreciarlo en la siguiente tabla:

Tabla 1. Comparativa entre las propiedades físicas de grafeno, acero y aluminio

Propiedades	Grafeno	Acero	Aluminio
Tensión de rotura (MPa)	130000	686	50
Módulo de Young (TPa)	42	0.21	0.07
Conductividad térmica (W/mK)	4000	58	209.3
Resistividad (S)	30	$20 \cdot 10^{-8}$	$2.75 \cdot 10^{-8}$
Conductividad electrónica ($S \cdot m^{-1}$)	$9.87 \cdot 10^7$	$3 \cdot 10^6$	$3.78 \cdot 10^7$

Es notable la abrupta diferencia entre los valores alcanzados entre el grafeno y aluminio y acero. Propiedades puramente mecánicas como las dos primeras de la tabla son órdenes de magnitud muy superiores a los de los metales; a lo que ha de añadirse que el grafeno cuenta con una densidad de 1 g/cm^3 por lo que una menor cantidad de grafeno cumpliría con facilidad solicitudes para las que sería necesario grandes proporciones de acero o aluminio.

Estas propiedades hacen al grafeno ideal para una gran cantidad de aplicaciones como:

- *Transistores de efecto de campo* que permitirán la creación de circuitos integrados más rápidos y con menos ruido.
- *Dispositivos móviles* de ergonomía adaptable gracias a la flexibilidad y ligereza del material
- Mejoras en *estructuras* de medios de transporte terrestre, marítimo y aéreo. Sus ya demostradas capacidades mecánicas junto a una ligereza extrema hacen que sea un material atractivo a la hora de reducir consumo sin perjudicar a la integridad estructural.

- Nuevos tratamientos para *combustibles* al permitir una fácil destilación de etanol a temperatura ambiente.
- Sensores para uso *médico* dada su sensibilidad para detectar cantidades ínfimas de otras sustancias, así como su empleo en un método eficiente de secuenciación del ADN.
- *Blindajes militares* gracias a su alta tenacidad, flexibilidad e incluso capacidad de regeneración si se nutre a la malla de átomos de carbono tras la creación de un orificio.



Figura 5. Prototipos de *smartphones* flexibles basados en grafeno

Sin embargo, cabe preguntarse por qué un material tan beneficioso para la sociedad no está siendo empleado ya en todo tipo de industria. La respuesta se encuentra en sus métodos de **producción**. Existen principalmente dos grupos de métodos para la obtención del grafeno:

- *Exfoliación*: o método de la cinta adhesiva, consiste en ir extrayendo capas de grafito repetidamente hasta que se alcance una capa monoatómica. Una vez obtenida dicha capa, se retira la cinta adhesiva mediante un proceso químico.
- *Epitaxia*: deposición de partículas de grafeno sobre un sustrato, típicamente carburo de silicio, níquel, rutenio o iridio, de forma controlada para la generación de la estructura cristalina. Una forma habitual de realización es mediante deposición química de vapor (CVD).

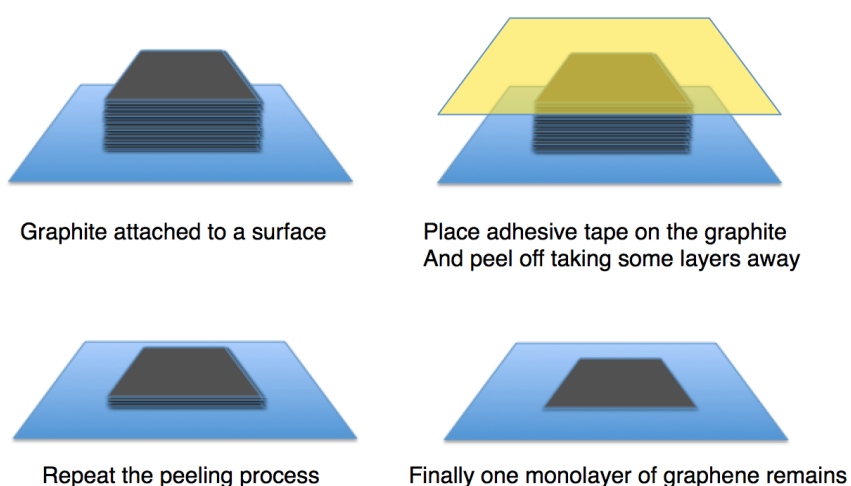


Figura 6. Exfoliación mecánica del grafeno

Estos métodos suponen, hoy en día, unos costes no asumibles para las cantidades generadas. Existen otros argumentos para asegurar que el grafeno no reemplazará al silicio en los dispositivos electrónicos, ni es la panacea tecnológica con la que a menudo se presenta, como puede ser su carencia de **resistividad**. Esta falta de

resistencia eléctrica hace que el grafeno no pueda dejar de conducir electricidad, lo que puede ser un gran inconveniente. Igual que sucedió en los años 20 con el plástico, el grafeno ha sido el protagonista de un *boom* mediático que le exige lograr unos niveles tecnológicos que la industria no puede alcanzar a tan corto plazo.

1.3 Objeto y motivo del trabajo

El alcance de todas estas “características prometidas” no se pueden determinar ni cuantificar si no se modela el grafeno para conocer su comportamiento real. Han existido a lo largo de la historia una serie de modelos que permiten definir la energía del carbono y sus derivados con mayor o menor precisión, entre ellos el potencial *Tight Binding*, sobre el cual se centra el esfuerzo del presente trabajo y se analizará con exhaustividad más adelante.

Dichos modelos han sido simulados computacionalmente durante años en aras de conseguir resultados aplicables en la práctica. Es por ello por lo que se busca optimizar el proceso todo lo posible y poder simular dichas implementaciones para recursos de tiempo y hardware limitados.

De aquí que nazca la necesidad del proyecto. En él se va a llevar a cabo una traducción del código armónico del *Tight Binding* implementado en Mathematica a su equivalente en MATLAB de modo que se pueda sacar partido de paradigmas de la computación numérica para la obtención de resultados experimentales.

2 COMPLEJO SIMPLICIAL DEL GRAFENO

*En el mundo del eterno retorno descansa,
sobre cada gesto, el peso de una
insoportable responsabilidad*

- Milan Kundera-

Existen distintos caminos para modelar el grafeno entre los que se pueden encontrar descripciones basadas en argumentos de la Teoría de la Elasticidad, sin embargo, esta descripción no será objeto de este capítulo. Es posible emplear otra sistemática, las técnicas de simulación de la Dinámica Molecular siempre y cuando sea posible contar con grandes tiempos de simulación.

Éstas últimas pueden ser combinadas con la Topología Algebraica. Esta rama de las Matemáticas posee herramientas (entre ellas la Homología) que permiten caracterizar las redes atómicas como conjuntos topológicos sobre los cuales definir unas **funciones**.

Bajo este marco, la red de grafeno puede ser considerada como **complejo simplicial** (CS); o lo que es lo mismo, un conjunto de celdas hexagonales las cuales incluyen átomos interconectados mediante enlaces atómicos. Una manera de simplificar esta nomenclatura es llamando a cada uno de estos elementos mediante la notación *celdas- p* , donde p indica la dimensión de la “celda” en cuestión:

- *Celdas-0* (e_0): hacen referencia a los átomos de la red.
- *Celdas-1* (e_1): hacen referencia a los enlaces unidimensionales interatómicos de la red.
- *Celdas-2* (e_2): hacen referencia a las celdas hexagonales generadas por dichos enlaces.

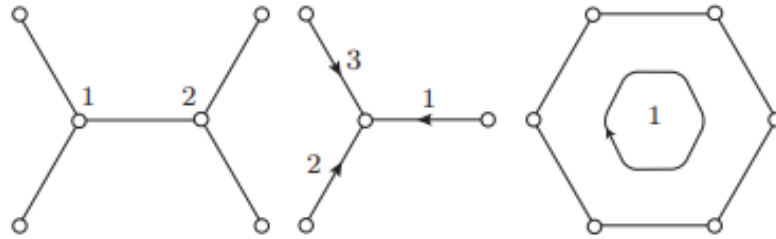


Figura 7. Celdas-p. Tipos de átomos, enlaces y celdas de una red de grafeno

Tal y como se muestra en la figura superior, pueden existir distintos tipos de celdas de la misma dimensión. Por definición, las celdas del mismo tipo son recursivas a lo largo de la red, es decir son translaciones las unas de las otras. Esta propiedad permite definir una base de Bravais en la red, la cual se basa en los vectores: $\mathbf{a}_1 = \left(\frac{3a}{2}, -\frac{\sqrt{3}a}{2}, 0\right)$ y $\mathbf{a}_2 = \left(\frac{3a}{2}, \frac{\sqrt{3}a}{2}, 0\right)$, siendo a el parámetro de red. La red es, por tanto, periódica y existirá una **invariancia ante traslaciones**.

Es, por tanto, posible hacer referencia a cualquier miembro de la red mediante dos coordenadas $l \in \mathbb{Z}^2$ en la base $(\mathbf{a}_1, \mathbf{a}_2)$.

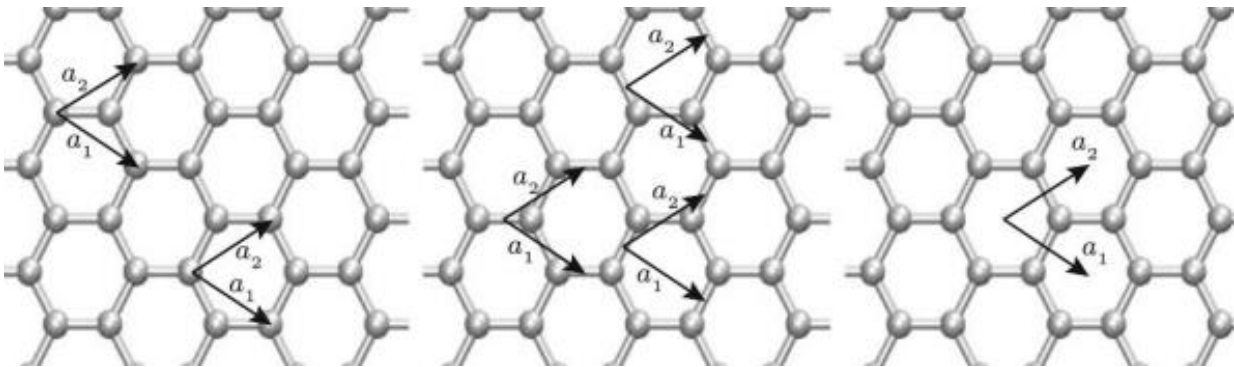


Figura 8. Red de Bravais del grafeno. Traslaciones de átomos, enlaces y celdas

Para cada celda-p se define un entorno local como la colección de complejos que inciden sobre e_p . De esta forma se dice que dos celdas-p e_p y e'_p son del mismo tipo si sus entornos locales $C(e_p)$ y $C(e'_p)$ son uno una traslación del otro.

Para mayor sencillez en la notación se denota como $e_p(\mathbf{l}, \alpha)$ a la celda-p del tipo α y coordenadas $(\mathbf{l}_1, \mathbf{l}_2) \in \mathbb{Z}^2$ dentro de la red de Bravais y como $E_p(C)$ al conjunto de celdas de dimensión p (con $p = 0, 1, 2$) en el complejo

diferencial C . De tal modo que, teniendo en cuenta la definición de entorno local que se acaba de proponer, es posible diferenciar los siguientes conjuntos:

- Átomos tipo 1 y 2: $E_0 = \{e_0(\mathbf{l}, \alpha), \mathbf{l} \in \mathbb{Z}^2, \alpha = 1, 2\}$
- Enlaces tipo 1, 2 y 3: $E_1 = \{e_1(\mathbf{l}, \alpha), \mathbf{l} \in \mathbb{Z}^2, \alpha = 1, 2, 3\}$
- Un solo tipo de celda: $E_2 = \{e_2(\mathbf{l}, \alpha), \mathbf{l} \in \mathbb{Z}^2\}$

Completando la definición anterior es preciso establecer la orientación de cada una de las “celdas” de modo que sea posible definir la estructura diferencial del CS. En la figura se muestra el criterio tomado:

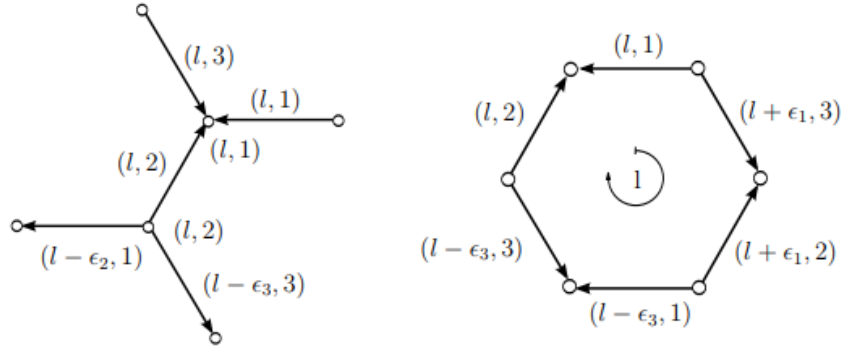


Figura 9. Orientación de las celdas de la red

Es de utilidad definir las direcciones ϵ_1, ϵ_2 , y ϵ_3 , coordenadas enteras de la red de Bravais, que se emplearán como base para las traslaciones. Estas cumplen ser $\epsilon_1 = (1,0)$, $\epsilon_2 = (0,1)$ y $\epsilon_3 = \epsilon_2 - \epsilon_1 = (-1,1)$.

Con objeto de definir los operadores diferenciales para la estructura homónima, se parte de que ω es una forma-0 definida sobre los átomos y e_{ab} es un enlace definido por los átomos a y b y suponiendo además que e_{ab} está definido de a a b . Entonces, el diferencial $d\omega(e_{ab})$ es:

$$d\omega(e_{ab}) = \omega(e_b) - \omega(e_a) \quad (2.1)$$

Suponiendo ahora que ω es una forma-1 definida sobre los enlaces y e_{abcdef} una celda hexagonal formada por los enlaces $e_{ab}, e_{bc}, e_{cd}, e_{de}, e_{ef}$ y e_{fa} , el operador diferencial toma la forma:

$$d\omega(e_{abcdef}) = \omega(e_{ab}) + \omega(e_{bc}) + \omega(e_{cd}) + \omega(e_{de}) + \omega(e_{ef}) + \omega(e_{fa}) \quad (2.2)$$

Teniendo la orientación de las celdas en mente, es posible desarrollar las expresiones anteriores para operadores diferenciales de las formas-p que correspondan:

$$d\omega_1(l, 1) = \omega_0(l, 1) - \omega_0(l + \epsilon_2, 2) \quad (2.3)$$

$$d\omega_1(l, 2) = \omega_0(l, 1) - \omega_0(l, 2) \quad (2.4)$$

$$d\omega_1(l, 3) = \omega_0(l, 1) - \omega_0(l + \epsilon_3, 2) \quad (2.5)$$

$$d\omega_2(l) = \omega_1(l, 2) - \omega_1(l, 1) + \omega_1(l + \epsilon_1, 3) - \omega_1(l + \epsilon_1, 2) + \omega_1(l - \epsilon_3, 1) - \omega_1(l - \epsilon_3, 3) \quad (2.6)$$

En este momento, conviene recordar que para operar con redes que cumplen con la invariancia traslacional, es de gran utilidad trabajar con la Transformada de Fourier Discreta (DFT), la cual se define de la forma:

$$\hat{\omega}(\theta, \alpha) = \sum_{l \in \mathbb{Z}^2} \omega(l, \alpha) e^{-i\theta l} \quad (2.7)$$

No está de más conocer que la antitransformada se expresa como:

$$\omega(l, \alpha) = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\omega}(\theta, \alpha) e^{i\theta l} d\theta_1 d\theta_2 \quad (2.8)$$

Para una explicación más exhaustiva de la Transformada de Fourier consultar Apéndice A. Aplicando estas definiciones a las formas-p se obtiene:

$$d\hat{\omega}(\theta, \alpha) = \sum_{\beta=1}^{N_p} Q \begin{pmatrix} \theta \\ \alpha \\ \beta \end{pmatrix} \hat{\omega}(\theta, \beta) \quad (2.9)$$

donde Q representa la estructura diferencial de la red en representaciones mediante la Transformada de Fourier. A través esta expresión es posible extender los operadores diferenciales una dimensión, pasando las formas-0 correspondientes a los átomos a formas-1; las formas-1, definidas en los enlaces atómicos, a formas -2; y las formas-2 de las celdas hexagonales, a vectores.

Dados los operadores diferenciales definidos desde (2.3) hasta (2.6), se obtienen:

$$Q_1(\theta) = \begin{pmatrix} 1 & -e^{i\theta_2} \\ 1 & -1 \\ 1 & -e^{i\theta_3} \end{pmatrix} \quad (2.10)$$

$$Q_2(\theta) = \begin{pmatrix} e^{i\theta_3} - 1 & 1 - e^{i\theta_1} & e^{i\theta_1} - e^{i\theta_3} \end{pmatrix} \quad (2.11)$$

Donde $\theta_3 = \theta_2 - \theta_1$. Q_1 define el diferencial de las formas-0, relacionando los elementos E_0 y E_1 , mientras que Q_2 es el diferencial de las formas-1 y relaciona elementos de E_1 y E_2 . En adelante también se empleará la matriz $P_i = Q_i^*$, que denota la conjugada de Q_i , con $i = 1, 2$. Su cálculo es idéntico al ya realizado, pero partiendo de las expresiones codiferenciales de los operadores:

$$\delta\omega^0(l, 1) = \omega^1(l, 1) + \omega^1(l, 2) + \omega^1(l, 3) \quad (2.12)$$

$$\delta\omega^0(l, 2) = -\omega^1(l, 2) - \omega^1(l' \epsilon_3, 3) - \omega^1(l - \epsilon_2, 1) \quad (2.13)$$

$$\delta\omega^1(l, 1) = -\omega^2(l) + \omega^2(l + \epsilon_3) \quad (2.14)$$

$$\delta\omega^1(l, 2) = \omega^2(l) + \omega^2(l - \epsilon_1) \quad (2.15)$$

$$\delta\omega^1(l, 3) = \omega^2(l - \epsilon_1) - \omega^2(l + \epsilon_3) \quad (2.16)$$

Resultando unas matrices P_i :

$$P_1(\theta) = \begin{pmatrix} 1 & 1 & 1 \\ -e^{i\theta_2} & -1 & -e^{i\theta_3} \end{pmatrix} \quad (2.17)$$

$$P_2(\theta) = \begin{pmatrix} e^{i\theta_3} - 1 \\ 1 - e^{i\theta_1} \\ e^{i\theta_1} - e^{i\theta_3} \end{pmatrix} \quad (2.18)$$

Que relacionarán los entornos locales E_i de manera inversa a como lo hacen las matrices Q_i .

3 MODELO TEÓRICO

*Cada descubrimiento abre un nuevo campo
para la investigación de los hechos,
nos muestra la imperfección de nuestras teorías.*

*Se ha dicho oportunamente,
que cuanto mayor es el círculo de luz,
mayor es el límite de la oscuridad de que está rodeado*

- Sir Humphry Davy-

Es completamente imprescindible tener la capacidad de cálculo de la energía total de un sistema molecular o atómico a fin de poder predecir propiedades físicas de los materiales. La energía de un sistema es esencial para un completo conocimiento que nos permita abordar cualquier problema propuesto por la dinámica molecular.

3.1 Potenciales interatómicos. Definición y contexto histórico

Una de las soluciones para este problema podría ser obtenida mediante argumentos cuánticos, sin embargo, las soluciones derivadas de este modo son computacionalmente costosas además de numéricamente inestables.

Un resultado clásico de los sistemas conservativos que permite el cálculo energético del problema es que las fuerzas actuantes sobre una partícula pueden obtenerse de un potencial. Partiendo de este pretexto, es posible simular el comportamiento de un material a nivel atómico según dos tipos de modelos teóricos: *empírico* y *ab initio*.

Los modelos **empíricos**, también conocidos como campos de fuerza o potenciales analíticos, expresan las fuerzas atómicas mediante funciones del tipo:

$$U(r_1, r_2, r_3) = \sum_i U_1(r_i) + \sum_i \sum_{j>i} U_2(r_i, r_j) + \sum_i \sum_{j>i} \sum_{k>j} U_3(r_i, r_j, r_k) + \dots \quad (3.1)$$

Siendo U_n la energía de orden n del átomo i . U_1 representa aquella energía asociada a las condiciones de contorno, U_2 a la interacción entre un par de átomos y U_3 añade términos a la anterior que reflejan interacciones entre tres átomos, de igual modo que lo harían los subsecuentes términos ($n > 3$). Es decir, la energía potencial es calculable mediante la construcción de un potencial que dependa sólo de las coordenadas de los átomos que conforman el sistema.

Existen dos clases de modelos empíricos:

- *Potenciales por pares*: de gran simpleza, pero fallan al representar defectos en la red atómica. Algunos

modelos analíticos tipo par son los de Lennard-Jones, Morse o Buckingham.

- *Potenciales reactivos*: describen de manera satisfactoria uniones atómicas y, por tanto, reacciones químicas. Existen varios modelos con estas características: Tersoff, Brenner, Finnis & Sinclair, Stuart, *ReaxFF*, *Embedded Atom Method* (EAM), *Reactive Empirical Bond-Order* (REBO), así como sus extensiones.

Los modelos **ab initio** en cambio, son aquellos basados en axiomas y leyes aceptadas de la física que se emplea en la resolución del problema, excluyendo cualquier valor tabulado o modelo simplificado. Como es lógico, este tipo de modelos proporcionan unos resultados precisos al minimizar los errores de definición de modelo; sin embargo, el tiempo y complejidad de computación ascienden, no pudiéndose abarcar hoy en día sistemas que superen cantidades del orden de los cien átomos. Algunos ejemplos son la *Density Functional Theory* (DFT), *Quantum Monte Carlo*, o la *Perturbation Theory*.

Con la idea de aprovechar las ventajas de ambos modelos, nace el potencial **Tight Binding**. Este potencial presenta un compromiso entre los dos anteriores, ya que permite un menor tiempo de computación (de 2 a 3 órdenes de magnitud más rápido que los modelos ab initio) junto con una precisión superior a la que puede dar un modelo empírico. Tiene también a su favor la capacidad de representar propiedades eléctricas del material. Se procederá a describir con mayor detenimiento este último a lo largo de este capítulo, dándose ahora una visión general de algunos de los demás modelos para una idea completa del concepto.

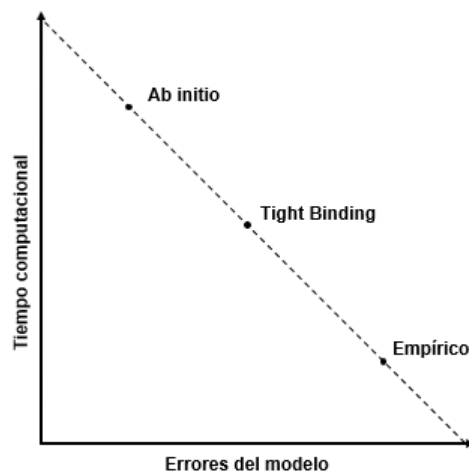


Figura 10. Compromiso entre tiempo computacional y precisión de los modelos interatómicos

Algunos de los modelos interatómicos más empleados dentro y fuera del ámbito de sistemas moleculares basados en el carbono son los siguientes:

- *Abell-Tersoff y Brenner*: Tersoff (1988) centró su trabajo en la construcción de un nuevo potencial interatómico que fuese capaz de reproducir enlaces y moléculas de mayor complejidad, así como los efectos de enlaces de mayor intensidad, como es el caso del enlace covalente en el Silicio.
- *Lennard-Jones*: potenciales de mínima complejidad como el 6-12, aplicables a cualquier configuración pero que sólo representan con eficacia sistemas de complejidad también simple.
- *REBO (Reactive Energy Bond-Order)*: la definición a base de la expresión de la dinámica clásica (1.1) limita el estudio a ciertas geometrías, dado que, si se quisiese aumentar la potencia del modelo, habría que retener los términos de cuatro y/o cinco cuerpos, algo computacionalmente muy costoso. La solución obtenida es la de emplear funciones que recojan los efectos multicuerpo (tan lejos como se pueda permitir) sobre los enlaces en términos b_{ij} conocidos como *términos orden del enlace* y dando lugar a un tipo de potencial conocido como *potencial multicuerpos*. Este tipo de potenciales desarrollados por Brenner (1990) y (2002) se les conoce por el nombre de REBO.
- *AIREBO (Adaptative Intermolecular Reactive Energy Bond-Order)*: su desarrollo surge con el objetivo de mejorar las prestaciones del modelo REBO, mediante la incorporación de fenómenos intermoleculares (vitales para la descripción y definición de algunos hidrocarburos), un radio de corte

mayor a 2Å permitiendo interacciones a más larga distancia y la inclusión de términos de torsión los cuales tienen un papel protagonista en la generación energética de muchos hidrocarburos.

$$E^{AIREBO} = E^{REBO} + E^{LJ} + E^{tors} \quad (3.2)$$

3.2 Principios físicos del *Tight Binding*

Como ya se ha comentado en el apartado anterior, el punto de partida de los potenciales interatómicos tratados son métodos mecánicos clásicos. Para el *Tight Binding* (TB), es preciso partir desde la ecuación de Schrödinger, en la cual se expresan las funciones de onda como combinaciones lineales de orbitales atómicos escritos en términos de funciones de salto y de repulsión (*hopping* y *repulsive*).

Para materiales basados en carbono como el grafeno, Xu et al. propusieron en 1992 un modelo de base ortogonal con la aproximación de dos centros, lo que viene a decir que las funciones *hopping* y *repulsive* dependen sólo de las distancias interatómicas. Este modelo ha sido usado de manera satisfactoria en varios sistemas de grafeno junto al TB para explicar comportamientos tanto mecánicos como electrónicos como se explicará posteriormente.

La ecuación para un solo electrón independiente de t de Schrödinger fue expresada por Saito *et al.* (1998) como:

$$E\Psi = \hat{H}\Psi \quad (3.3)$$

donde E es la energía del sistema, Ψ es la función de onda y \hat{H} es el operador Hamiltoniano, definido como:

$$\hat{H} = \frac{\hbar^2}{2m} \nabla^2 + V(r) \quad (3.4)$$

donde m es la masa de la partícula, ∇^2 es el operador Laplaciano y $V(\mathbf{r})$ es la energía potencial. La función de onda presentada en (3.3) puede ser escrita como una combinación lineal de orbitales atómicos (Slater & Koster, 1954):

$$\Psi(r) = \sum_{j\beta} c_{j\beta} \phi_{j\beta}(r - R_j) \quad (3.5)$$

donde se avanza a lo largo de orbitales ($\beta = s, p_x, p_y, p_z$) y átomos ($j = 1, \dots, N$). Para resolver la ecuación de Schrödinger (3.3) ha de minimizarse la expresión:

$$E = \frac{\int \Psi^* \hat{H} \Psi dr}{\int \Psi^* \Psi dr} \quad (3.6)$$

Apoyándose para ello en la propia definición de las funciones de onda se llega a que:

$$E = \frac{\sum_{i\alpha, j\beta} c_{j\beta} \langle \phi_{i\alpha} | \hat{H} | \phi_{j\beta} \rangle}{\sum_{i\alpha, j\beta} c_{i\alpha}^* c_{j\beta} \langle \phi_{i\alpha} | \phi_{j\beta} \rangle} \quad (3.7)$$

Una ecuación sobre la que podemos optimizar respecto a $c_{i\alpha}^*$:

$$0 = \frac{\sum_{j\beta} c_{j\beta} \langle \phi_{i\alpha} | \hat{H} | \phi_{j\beta} \rangle}{\sum_{i\alpha, j\beta} c_{i\alpha}^* c_{j\beta} \langle \phi_{i\alpha} | \phi_{j\beta} \rangle} - \frac{\sum_{i\alpha, j\beta} c_{i\alpha}^* c_{j\beta} \langle \phi_{i\alpha} | \hat{H} | \phi_{j\beta} \rangle}{\left(\sum_{i\alpha, j\beta} c_{i\alpha}^* c_{j\beta} \langle \phi_{i\alpha} | \phi_{j\beta} \rangle \right)^2} \sum_{j\beta} c_{j\beta} \langle \phi_{i\alpha} | \phi_{j\beta} \rangle \quad (3.8)$$

Que permire reescribir la ecuación de Schrödinger de la siguiente manera:

$$E \sum_{j\beta} S_{i\alpha, j\beta} c_{j\beta} = \sum_{j\beta} H_{i\alpha, j\beta} c_{j\beta} \quad (3.9)$$

Donde:

$$H_{i\alpha, j\beta} \equiv \langle \phi_{i\alpha} | \hat{H} | \phi_{j\beta} \rangle \equiv \int \phi_{i\alpha}^*(r - R_i) \hat{H} \phi_{j\beta}(r - R_j) dr \quad (3.10)$$

$$S_{i\alpha, j\beta} \equiv \langle \phi_{i\alpha} | \phi_{j\beta} \rangle \equiv \int \phi_{i\alpha}^*(r - R_i) \phi_{j\beta}(r - R_j) dr \quad (3.11)$$

Denominando $H_{i\alpha, j\beta}$ y $S_{i\alpha, j\beta}$ son conocidos como integrales de salto (*hopping*) y superposición (*overlap*), respectivamente.

De este modo, se obtiene un problema de autovalores definido como:

$$Hc = ES c \quad (3.12)$$

Con H y S el Hamiltoniano del modelo *Tight Binding* (TBH) y matriz de superposición, respectivamente. Ambas son matrices de dimensión $4N \times 4N$ con N el número de átomos del sistema.

Nótese como S está definida de tal manera que, en una base ortogonal (lo cual se ha impuesto al principio del desarrollo) sería la matriz identidad I , simplificando el problema (3.12) en gran medida.

3.2 Modelo *Tight Binding*

Para una descripción exhaustiva del modelo es preciso partir de la energía total de un cristal bajo el marco del potencial *Tight Binding* definido por Xu *et al.* (1992):

$$E_{total} = E_{band} + E_{repulsive} \quad (3.13)$$

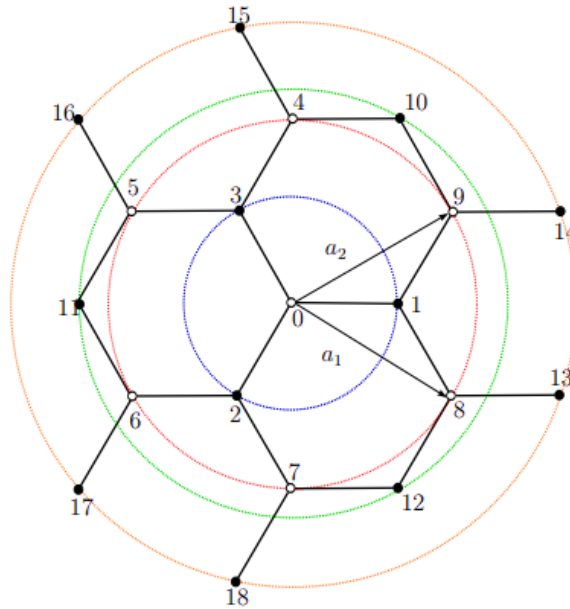


Figura 11. Átomos vecinos en una red de grafeno (átomos tipo 1: blanco, átomos tipo 2: negro)

El primer término hace referencia a la **energía de banda** (*band structure energy*), cuya expresión es:

$$E_{band} = \sum_i f_i \epsilon_i \quad (3.14)$$

Donde cada ϵ_i son los autovalores de un electrón en las posiciones ocupadas de la red y f_i es la función de ocupación electrónica (Fermi-Dirac):

$$f(\epsilon_i) = \frac{1}{1 + e^{\frac{\epsilon_i - E_F}{kT}}} = \begin{cases} 1 & \Leftrightarrow \epsilon_i < E_F \\ 0 & \end{cases} \quad (3.15)$$

Siguiendo con las definiciones asociadas a esta energía se tiene que el Hamiltoniano empírico del *Tight Binding* es:

$$H_{i\alpha,j\beta} = E_\alpha \delta_{ij} \delta_{\alpha\beta} + h_{i\alpha,j\beta} (1 - \delta_{ij} \delta_{\alpha\beta}) \quad (3.16)$$

o desarrollado en forma matricial:

$$H_{i\alpha,j\beta} = \begin{pmatrix} H_{1\alpha,1\beta} & \cdots & \cdots & \cdots & H_{1\alpha,N\beta} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \cdots & \cdots & H_{i\alpha,j\beta} & \cdots & \cdots \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ H_{N\alpha,1\beta} & \cdots & \cdots & \cdots & H_{N\alpha,N\beta} \end{pmatrix}$$

Recordar que:

- i, j : posiciones atómicas. Van tomando valores de 1 a N.
- α, β : orbitales atómicos. Pueden ser s, p_x, p_y o p_z

Se va a considerar un modelo que tiene en cuenta simplemente interacciones hasta el primer vecino. Por ejemplo, dado una posición $i = 0, j$ solo podrá valer 1, 2 y 3 (ver Figura 11). Por esto, los parámetros On-Site, E_α , que constituyen la matriz Hamiltoniana son las correspondientes energías orbitales atómicas E_s y E_p así como de una serie de funciones $h_{i\alpha,j\beta}$ de la forma:

$$h_{i\alpha,j\beta} = V_{\alpha\beta} S(r_{ij}) \quad (3.17)$$

Parámetros a su vez estructurados como:

- $V_{\alpha\beta} = V_{ss\sigma}, V_{sp\sigma}, V_{pp\sigma}, V_{ss\pi}$, constantes tabuladas.
- $S(r_{ij}) = \begin{cases} \left(\frac{r_0}{r_{ij}}\right)^n \exp\left[n\left(-\left(\frac{r_{ij}}{r_c}\right)^{n_c} + \left(\frac{r_0}{r_c}\right)^{n_c}\right)\right] & \Leftrightarrow r_{ij} \leq r_1 \\ c_{0s} + c_{1s}(r_{ij} - r_1) + c_{2s}(r_{ij} - r_1)^2 + c_{3s}(r_{ij} - r_1)^3 & \Leftrightarrow r_1 \leq r_{ij} \leq r_m \end{cases} \quad (3.18)$

Con $r_m = 2.6\text{\AA}$, distancia entre primer y segundo vecino de la red, la distancia de corte. Se tiene que r_{ij} es el módulo del vector de posición entre los átomos i y j . Los demás parámetros incluidos en las expresiones $(r_0, r_1, r_c, n, n_c, c_{0s}, c_{1s}, c_{2s}, c_{3s})$ son también constantes tabuladas.

El otro término a tener en cuenta es la **energía de repulsión de corto alcance** (*Short range repulsive energy*), cuya expresión es:

$$E_{repulsive} = \sum_i f\left(\sum_j \Phi(r_{ij})\right) \quad (3.19)$$

Siendo los parámetros que conforman la expresión, en este caso:

- $f = \sum_{n=0}^4 c_n X^n \quad (3.20)$

$$\bullet \quad \Phi(r_{ij}) = \begin{cases} \Phi_0 \left(\frac{d_0}{r_{ij}} \right)^n \exp \left[m \left(- \left(\frac{r_{ij}}{d_c} \right)^{m_c} + \left(\frac{d_0}{d_c} \right)^{m_c} \right) \right] \Leftrightarrow r_{ij} \leq d_1 \\ c_{0\phi} + c_{1\phi}(r_{ij} - d_1) + c_{2\phi}(r_{ij} - d_1)^2 + c_{3\phi}(r_{ij} - d_1)^3 \Leftrightarrow d_1 \leq r_{ij} \leq d_m \end{cases} \quad (3.21)$$

Con $d_m = 2.6\text{\AA}$ de nuevo. De forma análoga al caso de la función $S(r_{ij})$, todos los parámetros involucrados en $\Phi(r_{ij})$ son constantes y conocidos. Éstos se recogen en las siguientes tablas:

 Tabla 2. Parámetros *On-Site*

Parámetros	Valor (eV)
E_s	-2.99
E_p	3.71
$V_{ss\sigma}$	-5.0
$V_{sp\sigma}$	4.7
$V_{pp\sigma}$	5.5
$V_{pp\pi}$	-1.55

 Tabla 3. Coeficientes para $S(r_{ij})$

Parámetros	Valor
n	2.0
n_c	6.5
$r_c(\text{\AA})$	2.18
$r_0(\text{\AA})$	1.536329
$r_1(\text{\AA})$	2.45
c_{0s}	$6.7392620074314 \cdot 10^{-3}$
c_{1s}	$-8.1885359517898 \cdot 10^{-3}$
c_{2s}	0.1932365259144
c_{3s}	0.3542874332380

Tabla 4. Coeficientes de f

Parámetros	Valor
c_0	-2.5909765118191
c_1	0.5721151498619
c_2	$-1.7896349903996 \cdot 10^{-3}$
c_3	$2.3539221516757 \cdot 10^{-5}$
c_4	$-1.24251169551587 \cdot 10^{-7}$

Tabla 5. Parámetros para $\Phi(r_{ij})$

Parámetros	Valor
$\Phi_0(eV)$	8.18555
m	3.30304
m_c	8.6655
$d_c(\text{Å})$	2.1052
$d_0(\text{Å})$	1.64
$d_1(\text{Å})$	2.57
$c_{0\phi}$	$2.2504290109 \cdot 10^{-8}$
$c_{1\phi}$	$-1.4408640561 \cdot 10^{-6}$
$c_{2\phi}$	$2.1043303374 \cdot 10^{-5}$
$c_{3\phi}$	$6.6024390226 \cdot 10^{-5}$

4 APLICACIÓN PRÁCTICA

*Los que se enamoran de la práctica sin la teoría
son como los pilotos sin timón ni brújula,
nunca podrán saber a dónde van*

- Leonardo Da Vinci-

Las computadoras han traído con su irrupción en el mundo actual cambios profundos a los que parece que la sociedad está ya más que acostumbrada, cambios que se han introducido en casi todo campo de estudio. Es todo un reto citar alguna disciplina en la que no se haga uso, de un modo u otro, de computadoras y software especializado para la resolución de problemas teórico-prácticos.

4.1 Introducción a la simulación computacional

La historia del encuentro entre la actividad científica y las simulaciones computacionales se remonta a la introducción de nuevas metodologías y herramientas que simplifican enormemente el tedioso y complicado cálculo obteniendo resultados de una mayor precisión. Una realidad soñada desde el siglo 17 por Gottfried Leibniz, quien estableció las bases de la arquitectura computacional, como el **sistema binario** o la rueda de Leibniz.

Hoy en día, el escenario ha avanzado sustancialmente. La posibilidad de realizar simulaciones de sistemas físicos cada vez más rápidas y precisas, suponen un cambio tecnológico-científico que pone de relieve el hecho de que las simulaciones computacionales deben ser una herramienta empleada hoy en día. El impacto de las simulaciones computacionales en la actividad científica estimula la curiosidad por conocer los alcances y características tanto de dichas simulaciones, como de los arraigados modelos científicos sobre los que nos basamos. En el momento en el que se emplea como una herramienta, la simulación computacional permite a base de comparativas y feedback, mejorar tanto el modelo numérico como teórico.

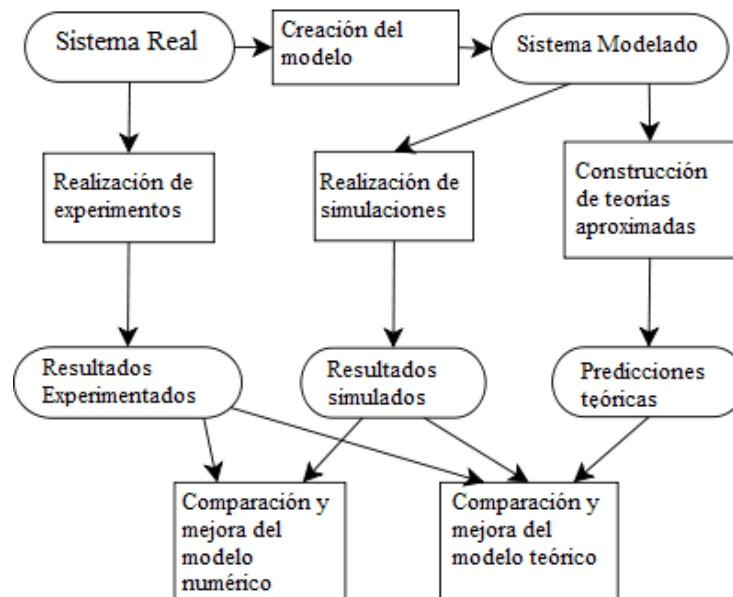


Figura 12. Algoritmo para la resolución y mejora de un modelo científico

Han sido muchos los softwares desarrollados a esta finalidad desde los comienzos de la simulación numérica allá por los años 50, conocidos como Sistemas de Álgebra Computacional (CAS). Algunos de los más conocidos son Mathematica, MATLAB, Maple, Cadabra o VEGA entre otros muchos. Serán los dos primeros los que protagonizarán esta sección del proyecto.

4.2 Descripción del entorno Mathematica

Lanzado en 1988 por Wolfram Research, este software permite el cálculo y programación matemática **simbólicos** todo en uno. Su estructura se divide en dos partes: el **kernel** y el **front end**. El primero es el “corazón” del programa, encargándose de la realización de los cálculos requeridos en el segundo, el **front end**, el cual recoge entradas del usuario y plasma las salidas correspondientes sobre una misma interfaz.



Figura 13. Logo Mathematica

Dicha interfaz incorpora en sus últimas versiones archivos de Formato de Documentos Computables (CDF), con los cuales se puede llevar a cabo en una sola ventana cálculos, visualizaciones de los resultados gráficos, texto, interactividades, etc.

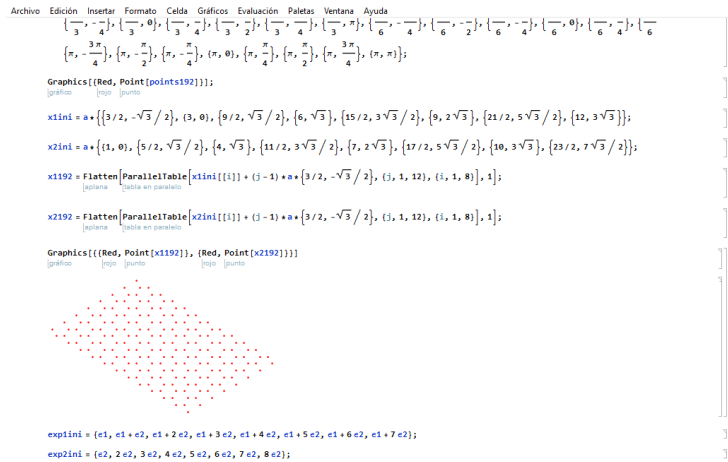


Figura 14. Disposición sobre una misma interfaz de texto, *script* matemático y gráficas

Mathematica cuenta con una gran variedad de bibliotecas matemáticas entre las que se puede elegir desde funciones básicas hasta funciones especiales de alta complejidad, pasando por sistemas de control o expresiones de estadística multivariable. Cuenta con un potente motor gráfico en el que, desde las últimas versiones, se puede disfrutar del desarrollo de gráficos de gran precisión y claridad.

En su otra cara, la de lenguaje de programación, Mathematica facilita un lenguaje de medio-alto nivel enfocado a desarrollos matemáticos. A priori, facilita su comprensión y supone una cómoda herramienta para la resolución de problemas ingenieriles sin separarse en demasía de un modelo matemático escrito en papel; sin embargo, supone el aprendizaje de una sintaxis única y un poco confusa al principio.

Aparte de la anterior, una de las principales desventajas que tiene es que los requerimientos de hardware son muy altos en cuanto se plantea un modelo numérico exigente, motivo principal del presente trabajo.

4.3 Descripción del entorno MATLAB

Como ya se ha comentado anteriormente, Mathematica es un software de carácter simbólico por lo que resulta cómodo y efectivo para la definición de modelos físicos de gran complejidad como el TB o cualquiera de los demás modelos del carbono ya planteados. Frente a este tipo tenemos los softwares **numéricos**. Éstos se diferencian en que no son una representación en lenguaje máquina de modelos matemáticos ya existentes con soluciones analíticas conocidas, sino que cumplen con las exigencias del usuario mediante sus propios algoritmos.

El concepto de computación numérica arroja una serie de características que pueden resultar atractivas para una definición de un potencial interatómico, entre ellas:

- Abre la posibilidad de calcular operaciones sin resultado analítico conocido mediante factorización o interpolación.
- Aunque no necesariamente más preciso, dado que un modelo simbólico siempre dará una solución exacta de obtenerla, los modelos numéricos alcanzan precisiones más que suficientes para aplicaciones ingenieriles.
- Los resultados, pese a poder darse el caso de no ser exactos hasta el último decimal, a menudo suponen tiempos de computación menores. Para una precisión suficiente, una solución obtenida con menores tiempos de computación siempre inclina la balanza hacia que esta sea un mejor resultado.

Sin embargo, no es posible computar numéricamente sin antes adaptar la física detrás del problema al lenguaje del programa. Es por esto por lo que los desarrollos numéricos van unidos aquellos simbólicos, siendo un fallo en uno un perjuicio para el otro. Dado la corroborada validez del modelo físico y de su implementación en Mathematica, es el momento de proporcionar un desarrollo numérico que permita lidiar con él con comodidad.

En MATLAB se encuentra una de las herramientas para el cálculo numérico más potentes hoy en día. Desarrollado por MathWorks es, junto a su “hermano” Simulink, uno de los programas más completos para el

análisis de datos, por lo que facilita el trabajo en muchos campos empíricos.

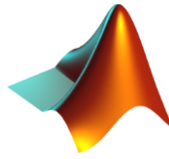


Figura 15. Logo de MATLAB

MATLAB, además, cuenta con un paradigma de programación que favorece la creación de *scripts* eficientes al tener una sintaxis similar a C o Python. Suponiendo un buen conocimiento de ambos programas, sería posible realizar modelos empíricos en MATLAB mediante bucles sencillos y traducirlo de manera que se pudiese extrapolar un modelo simbólico en Mathematica; sin embargo, este último siempre tendrá un tiempo de computación superior y una precisión menos sólida que la que daría MATLAB.

Se puede encontrar también distintas *third party applications* para aumentar los recursos disponibles. Éstas aparecen en ventanas externas para complementar al código:

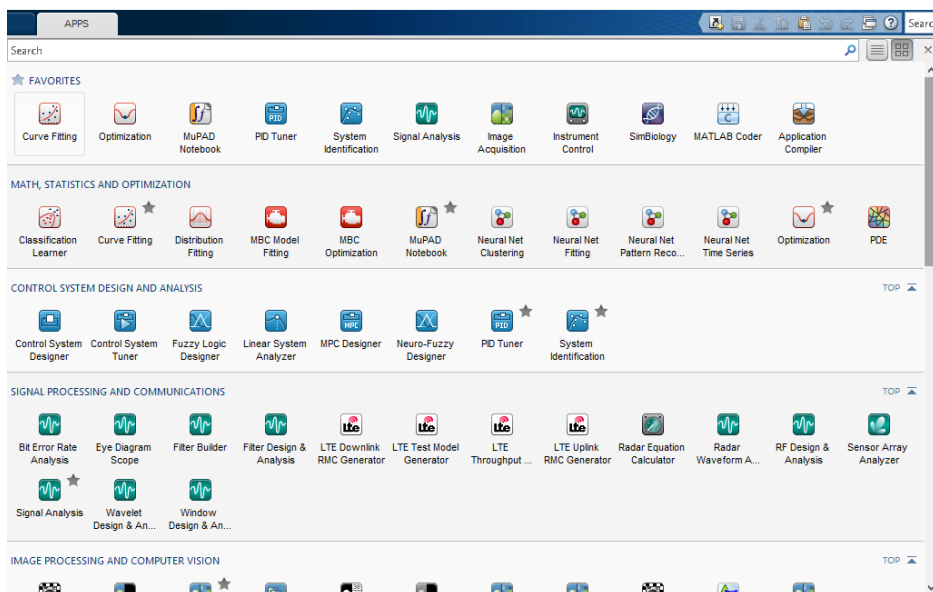


Figura 16. *Apps* disponibles en MATLAB 2015

Algunas son ajustes y optimizaciones gráficas, procesamiento de señales o incluso enfocadas a la biología computacional. Especial mención merece el MATLAB Coder, *app* que permite una traducción desde .m (código en MATLAB) a .cpp mediante una aplicación perteneciente al software, pero sin ser parte del código.

El entorno per se de MATLAB es intuitivo. Lo primero con lo que se interactúa es la **ventana de comandos**, una ventana en la que es posible escribir códigos de pequeña complejidad, usarlo como una potente calculadora o para que muestre en pantalla los resultados desarrollados en códigos más complejos.

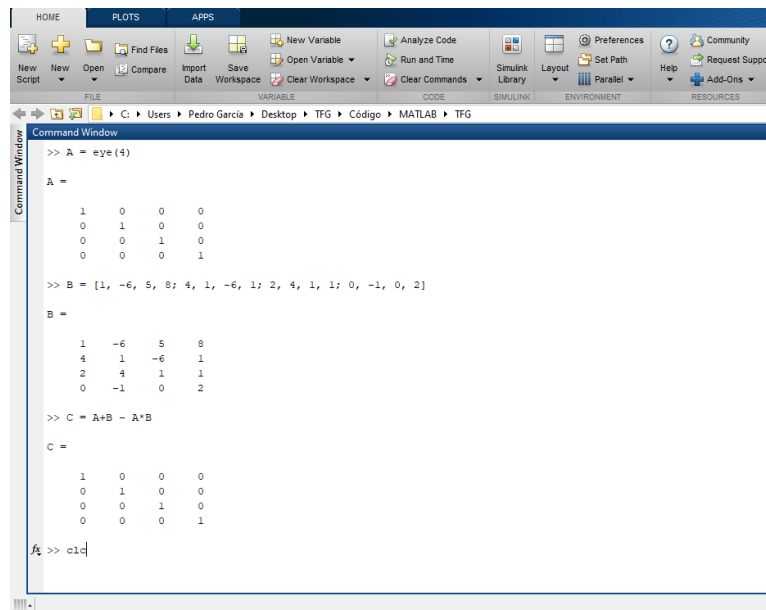


Figura 17. Captura de la ventana de comandos

La otra parte que constituye MATLAB como tal es el **editor**. En él se dispone de un entorno enfocado a la escritura de código como podría ser Notepad++ o Adobe Dreamweaver, con la evidente diferencia de que el editor de MATLAB está enfocado al desarrollo de código .m dentro de su interfaz.

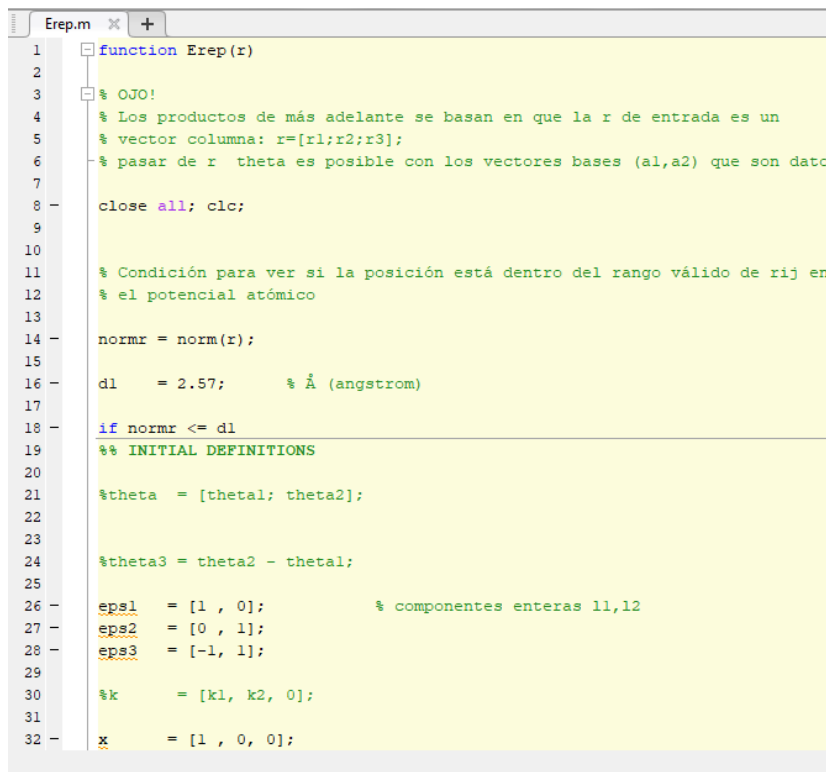


Figura 18. Captura del editor de código

En resumen, se busca disponer de una versión .m del modelo TB ya desarrollado en Mathematica para poder llevar más lejos la simulación a casos prácticos de mayores dimensiones y en menores tiempos de computación para una misma computadora.

4.4 Soluciones a las diferencias de sintaxis

A la hora de plantear una versión en MATLAB del modelo de Mathematica, se ha tenido que evitar enfocar su estructura como una traducción directa ya que las resoluciones llevadas a cabo en el *input* (*kernel displacement TB harmonic.nb*) siguen de manera exacta el planteamiento teórico del modelo.

Para lograr que la solución converja en la sintaxis de MATLAB, se han tenido que crear traducciones dado casos concretos como los siguientes:

- Sumatorios: así como en Mathematica se pueden realizar sumatorios con la función propia *Sum [f, {i, imax}]*, en MATLAB estos se han planteado a través de bucles *for*.
- Trasposiciones: en la sintaxis de MATLAB es de vital importancia tener una estructura ordenada de filas y columnas en las variables empleadas. Como su propio nombre indica, *MATrix LABoratory*, el poder de este lenguaje reside en el tratamiento en forma de matriz de sus variables, es por ello por lo que operaciones que afecten a las dimensiones han de ser tratadas con especial cuidado. Mathematica emplea la función *Transpose[list]* la cual no afecta a productos matriciales cuando se emplea, lo cual puede llevar a un cambio en las dimensiones de la matriz resultado. Esta problemática se localizó a la hora de realizar la transformación de Fourier en el cálculo de las matrices Ψ de los cuatro primeros *bonds*.
- Rotación de matrices: de nuevo, Mathematica consta de una función interna que rota elementos (*RotationMatrix [θ, ω]*) en la cual se determina cuanto y respecto a que eje ha de rotarse. En lugar de eso, se han empleado las matrices necesarias mediante su cálculo con los cosenos directores.
- Tratamiento de tensores: el principal problema encontrado a la hora de obtener una solución en MATLAB del sistema. Pese a su potencia a la hora de tratar matrices, MATLAB plantea un enfoque diferente para trabajar con tensores ya que estos se almacenan como una lista de matrices $m \times n$, pudiendo obtenerse a qué “capa” del tensor se quiere acceder mediante un tercer parámetro (la tercera dimensión). Esto diverge del modo del que se usa de traducir dado que Mathematica agrupa en una variable tipo matriz todas las matrices $m \times n$ y se accede a ellas mediante un par de valores i, j . Es por esta nomenclatura que se ha trabajado en todo momento con elementos matriciales obviando la posibilidad de los tensores en aras de obtener un código menos engorroso y fácil de editar en el futuro.

4.5 Proceso de resolución

El núcleo del cálculo se realiza en el archivo *Main_code.m*. Este recibe como parámetros de entrada un par de valores (θ_1, θ_2) . Éstos son las coordenadas $(\mathbf{l}_1, \mathbf{l}_2) \in \mathbb{Z}^2$ dentro de la red de Bravais en el dominio de Fourier (ver (2.7)). Un breve resumen de su definición y utilidad de su periodicidad se puede encontrar en el Apéndice A al final de la lectura.

Se conoce que los parámetros de entrada han de estar en el rango $[-\pi, \pi]$, para un buen planteamiento del modelo en el sentido de unicidad de las soluciones, a través del operador discreto Laplace-deRham:

$$\hat{\Delta} = d\delta + \delta d \quad (4.1)$$

Ya que, aplicándose al modelo, se obtiene:

$$\hat{\Delta}(\theta) = Q_2 P_2 = 2(3 - \cos \theta_1 - \cos \theta_2 - \cos \theta_3) \quad (4.2)$$

Y de ahí se puede deducir que, si los parámetros de entrada están en el rango anterior, $\hat{\Delta}(\theta) \neq 0$.

En las primeras líneas de código se procede a crear las definiciones iniciales de variables de utilidad en el futuro, desde los vectores de posición hasta las matrices de rotación antes mencionadas, incluso las matrices Q_1 y Q_2 al constarse ya de la variable θ .

A partir de aquí el *core* del sistema hace llamadas recursivas a los códigos *Eband.m*, *Ebdsr.m* y *Erep.m* para obtener de ellos los valores de las energías de banda, banda de corto alcance (*short range*) y repulsión, respectivamente.

Las constantes de fuerza en el dominio de Fourier se obtienen mediante la suma de los correspondientes términos de cada capa de vecinos (primeros *bonds*, segundos *bonds*, terceros *bonds*, ...) acumulándose en submatrices de dimensiones 3×3 . A cada una de estas submatrices se le aplica un factor de conversión de 1.6022×10^4 para obtener sus valores en las unidades de $g/cm s$.

En la variable Ψ_{ij} se irán acumulando cada una de estas submatrices, dando como resultado una matriz 9×9 .

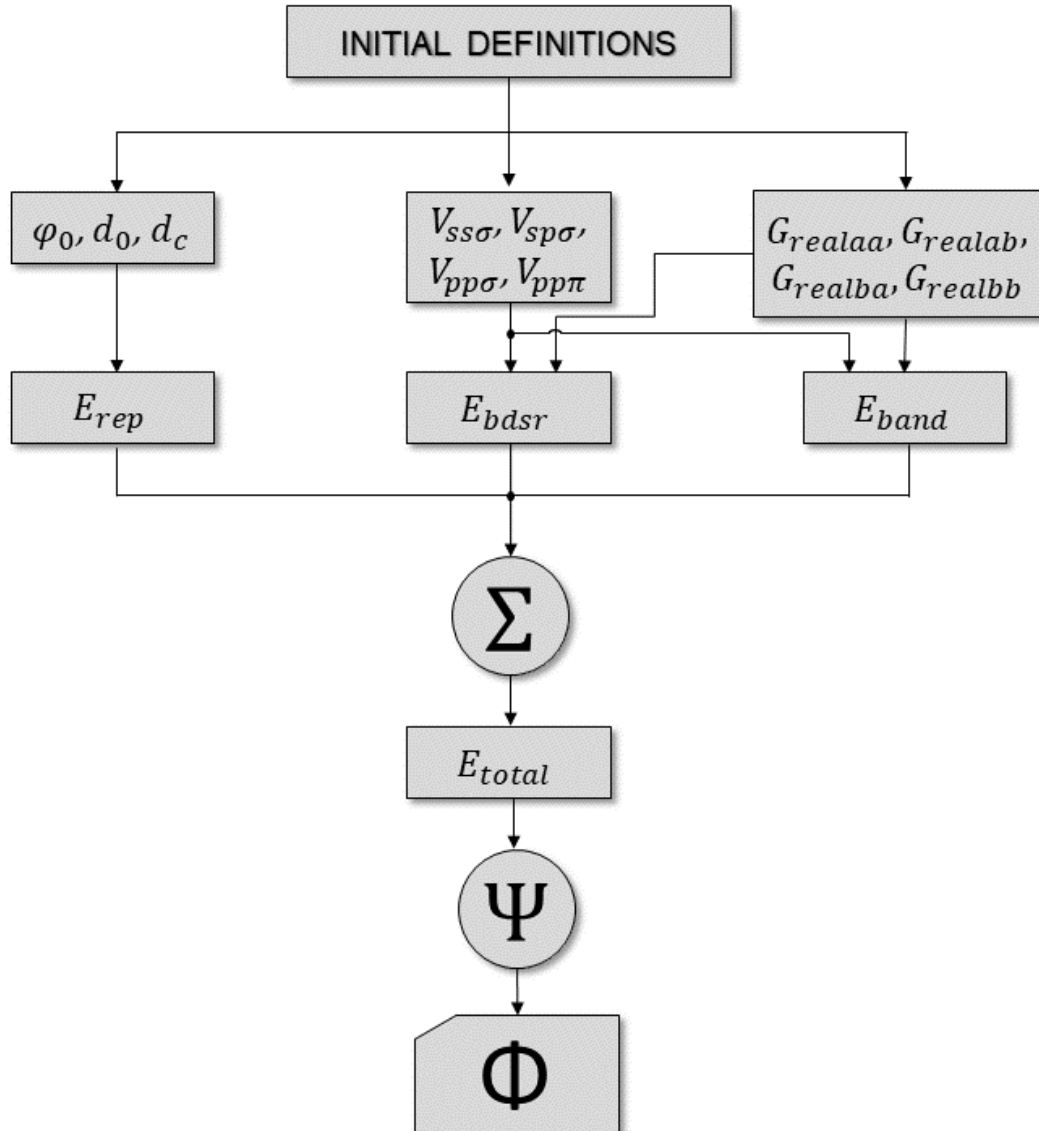


Figura 19. Diagrama de flujo correspondiente al proceso de resolución

El resultado ultimo, la matriz Φ_{ij} que contiene la energía de interacción resultante de un desplazamiento unitario en la dirección de la coordenada j del átomo $e_0(\mathbf{m}, \beta)$ y un desplazamiento unitario en la dirección de la coordenada i del átomo $e_0(\mathbf{l}, \alpha)$. Su obtención a partir de Ψ_{ij} se obtiene mediante el producto exterior generalizado de ésta con las matrices Q_1^T y Q_1^* . En Mathematica al constar de funciones para el sumatorio dependientes de índices dados por el usuario y de una sintaxis de acceso a submatrices, este producto se hace sin más que presentar las matrices con unos índices acertados, en MATLAB, sin embargo, es preciso hacer un cálculo más incisivo.

Como se ha comentado previamente, la matriz Ψ_{ij} es de dimensiones 9×9 , mientras que Q_1^T y Q_1^* son 2×3 y 3×2 respectivamente. Esto significa que para que el producto matricial se pueda realizar, ha de seleccionarse una submatriz antes de multiplicar.

La selección de esta matriz se realiza recorriendo internamente Ψ_{ij} , para un i, j dados, y extrayendo de ésta la submatriz correspondiente, generando unas matrices auxiliares de dimensión 3×3 que coinciden con las dimensiones de las matrices Q . En resumidas cuentas, ha de reducirse Ψ_{ij} de 9×9 a 3×3 .

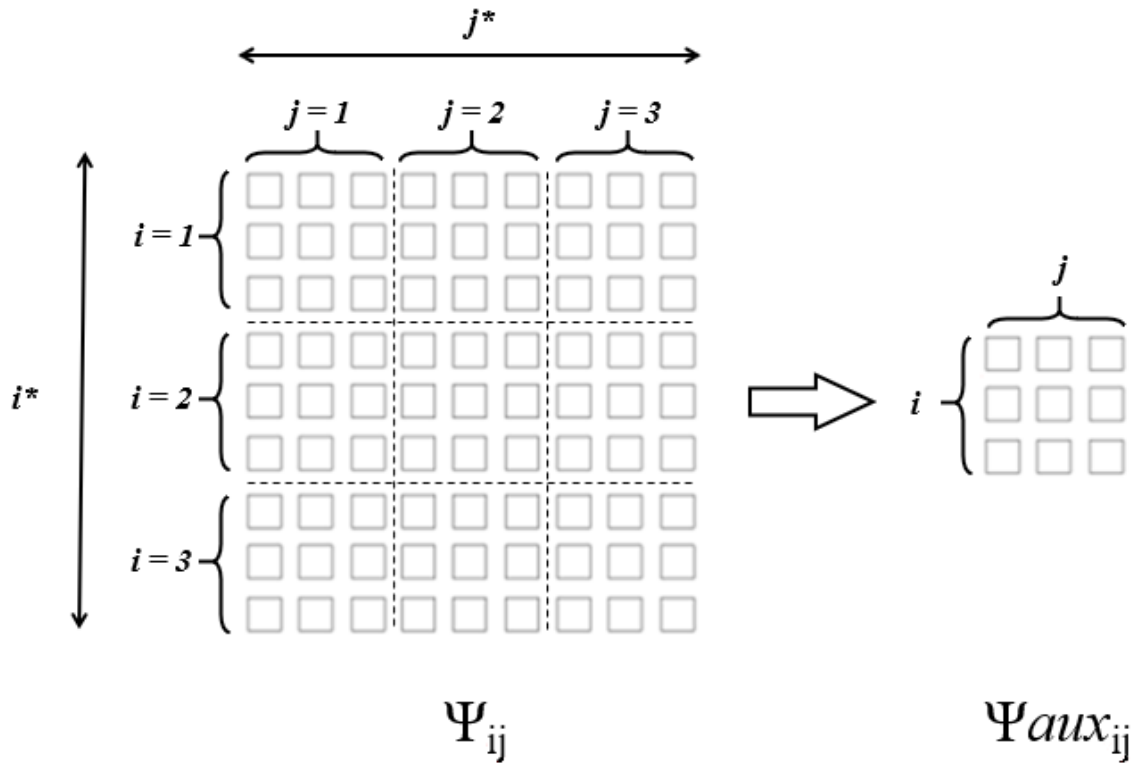


Figura 20. Croquis del redimensionamiento de la matriz Ψ_{ij}

Para reflejar este cambio de índices en lenguaje de código se ha usado el Teorema de Tales. La necesidad es encontrar una manera de que, para un valor de i entre 1 y 3, se recojan los tres valores de i^* (siendo i^* las coordenadas, desde 1 hasta 9, de la matriz original) que corresponden a las filas que componen la submatriz que se empleará en esa iteración, ídem con el parámetro j .

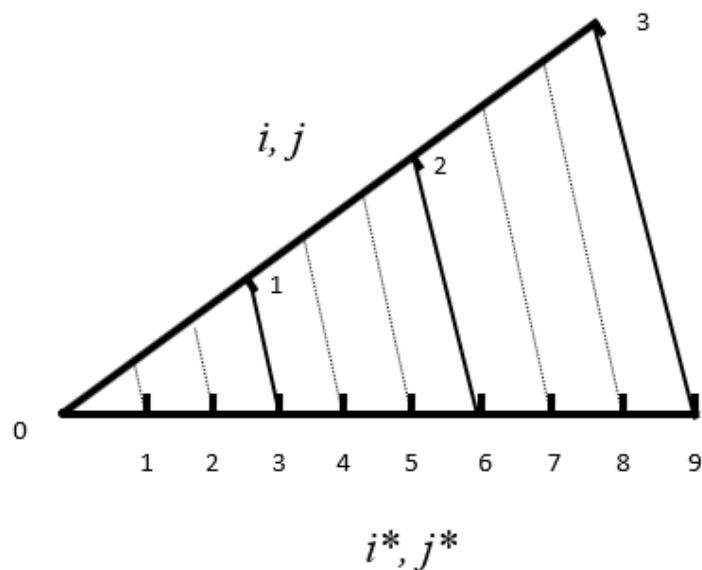


Figura 21. Relación entre índices por medio del Teorema de Tales

A la vista de como este Teorema relaciona los índices de ambas matrices, se ve que para el valor ultimo es tres veces el índice i , osea $i^*_{final} = 3 * i$; y dado que se ha fijado ya el final, las dos posiciones anteriores se obtendrán

restando uno y dos al valor final. MATLAB consta con un tipo de notación muy útil en esta situación, en la que al indicar los índices de una matriz o vector, se puede dar el valor inicial y el final separados por dos puntos y el programa automáticamente realizará las operaciones iterando en un bucle e incrementando de uno en uno el índice de la matriz.

Para acabar de traducir la operación, es preciso saber por último cómo ir acumulando el resultado de cada producto de las matrices Q con la submatriz que se ha extraído. El proceso de una multiplicación de matrices se reduce a un par de bucles *for* anidados para que recorran las matrices en filas y columnas, en cada iteración se obtiene un valor que ha de acumularse para que se haga efectivo el sumatorio.

Juntando todo lo razonado anteriormente, el bucle se reduce a:

```
for i = 1:3
    for j = 1:3
        Phi11 = Phi11 + Q1T (1, i) * Psi (3*i-2:3*i, 3*j-2:3*j) * Q1C (j, 1);
        Phi22 = Phi22 + Q1T (2, i) * Psi (3*i-2:3*i, 3*j-2:3*j) * Q1C (j, 2);
        Phi12 = Phi12 + Q1T (1, i) * Psi (3*i-2:3*i, 3*j-2:3*j) * Q1C (j, 2);
        Phi21 = Phi21 + Q1T (2, i) * Psi (3*i-2:3*i, 3*j-2:3*j) * Q1C (j, 1);
    end
end
```

Siendo Phi11, Phi22, Phi12 y Phi21 cuatro matrices que constituyen la matriz final Φ_{ij} .

4.6 Resultados

Para la comprobación de que ambos códigos convergen en el mismo resultado, se han ejecutado sendos códigos para una serie de valores dentro del rango que se estableció en el apartado anterior. Para la depuración del código se partió con los valores de $\theta_1 = 0.5$ y $\theta_2 = 0.25$. Con ellos, el código de Mathematica devuelve la siguiente matriz Φ_{ij} :

$$\begin{pmatrix} 1.01977 \times 10^6 - 3318.81 i & -8810.03 - 1761.07 i & 0. + 0. i & -992661. + 194321. i & 3010.85 - 38906.5 i & 0. + 0. i \\ -8810.03 - 2071.16 i & 1.0096 \times 10^6 + 3318.81 i & 0. + 0. i & 7904.13 - 39211.3 i & -994526. + 141396. i & 0. + 0. i \\ 0. + 0. i & 0. + 0. i & 300541. + 0. i & 0. + 0. i & 0. + 0. i & -294318. + 49528.3 i \\ -993911. - 187798. i & 6747.82 + 42990.5 i & 0. + 0. i & 1.01588 \times 10^6 - 3310.73 i & -8199.49 - 1271.43 i & 0. + 0. i \\ 2706.51 + 42792.8 i & -993300. - 147730. i & 0. + 0. i & -9051.48 - 2684.92 i & 1.01894 \times 10^6 + 3122.2 i & 0. + 0. i \\ 0. + 0. i & 0. + 0. i & -294326. - 49461.2 i & 0. + 0. i & 0. + 0. i & 296640. - 67.0695 i \end{pmatrix}$$

Figura 22. Matriz Φ_{ij} en Mathematica para $\theta_1 = 0.5$, $\theta_2 = 0.25$

El código en MATLAB devuelve, para los mismos valores, el siguiente resultado:

```
Columns 1 through 3
1019772.31414623 -      3318.806927629921      -8810.03189155084 -      1761.068213740571      0 +      0i
-8810.03189155052 -      2071.15993237031      1009599.36617275 +      3318.806927629931      0 +      0i
      0 +      0i      0i      300541.327301092 +      9.66338120633736e-13i
-993911.247498988 -      187798.2802577721      6747.81563163767 +      42990.48819344151      0 +      0i
2706.51495339649 +      42792.76972340571      -9933299.522851118 -      147730.4622832531      0 +      0i
      0 +      0i      0i      0 +      0i      -294326.275856874 -      49461.1809243598i

Columns 4 through 6
-992660.807738878 +      194321.2333968021      3010.84920240208 -      38906.4836903831i      0 +      0i
7904.13456928617 -      39211.25841283021      -994526.273102433 +      141396.0371220071      0 +      0i
      0 +      0i      0i      0 +      0i      0i      -294317.84822387 +      49528.2504706885i
1015882.39867655 -      3310.727142473151      -8199.4908527954 -      1271.43418955904i      0 +      0i
-9051.47554143853 -      2684.922536455051      1018936.51785147 +      3122.199164689531      0 +      0i
      0 +      0i      0i      0 +      0i      0i      296639.638569212 -      67.0695463287138i
```

Figura 23. Matriz Φ_{ij} en MATLAB para $\theta_1 = 0.5$, $\theta_2 = 0.25$

Los valores, tanto la parte imaginaria como la real, coinciden exactos hasta el segundo decimal momento en el que Mathematica deja de ofrecer valores por pantalla. Para obtener a la vez una mejor forma de visualizar los resultados y una comprobación de que ambos códigos coinciden más allá de lo que los programas le ofrecen al usuario al ejecutarlos, se han tomado ambas matrices para representar sus valores de manera gráfica. De este

modo se pueden localizar discrepancias de un vistazo sin necesidad de operar con las matrices o de ir término a término comprobando.

Para este mismo ejemplo, los resultados se han representado en tres dimensiones, siendo las filas y las columnas los ejes X e Y mientras que el eje Z determina el valor que toma esa posición matricial.

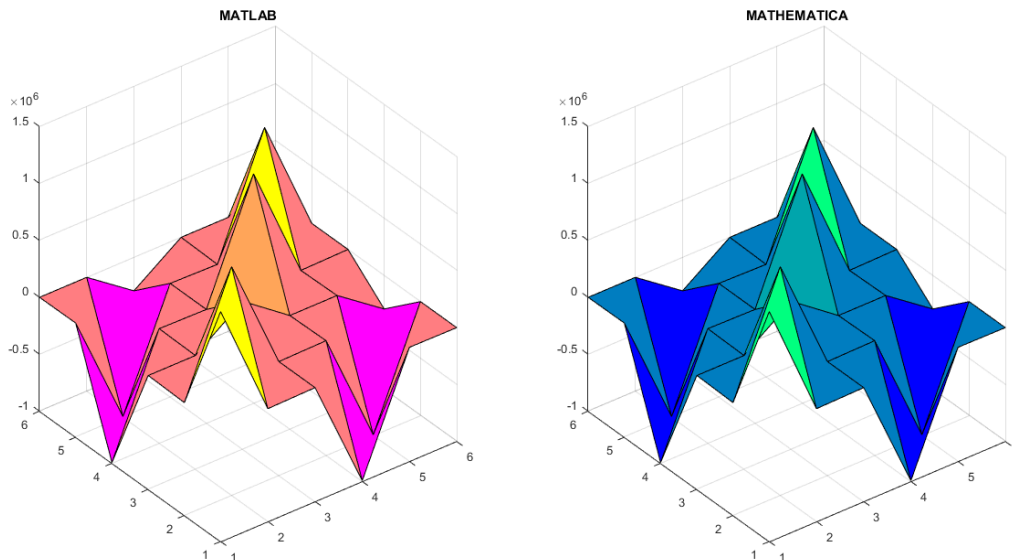


Figura 24. Representación de la parte real de Φ_{ij} para $\theta_1 = 0.5$, $\theta_2 = 0.25$ en ambos programas

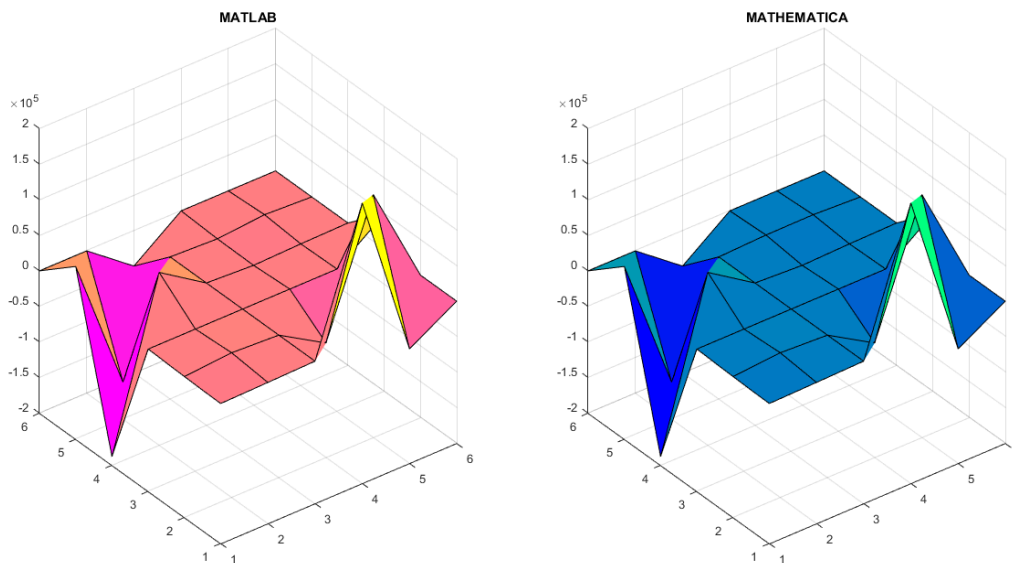


Figura 25. Representación de la parte imaginaria de Φ_{ij} para $\theta_1 = 0.5$, $\theta_2 = 0.25$ en ambos programas

Si fijamos una coordenada a partir de los resultados, se puede generar una vista lateral de estos para mejor comprobación de que ambos códigos aportan los mismos resultados.

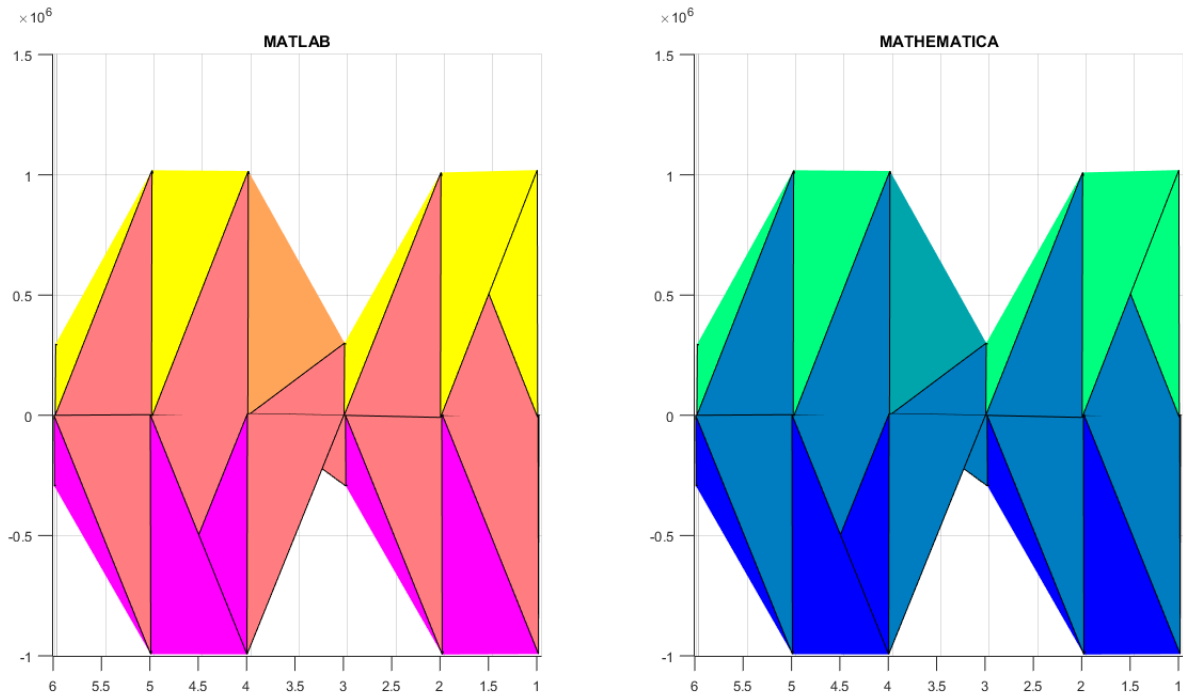


Figura 26. Vista lateral. Mostrando los resultados para los valores reales con una x constante

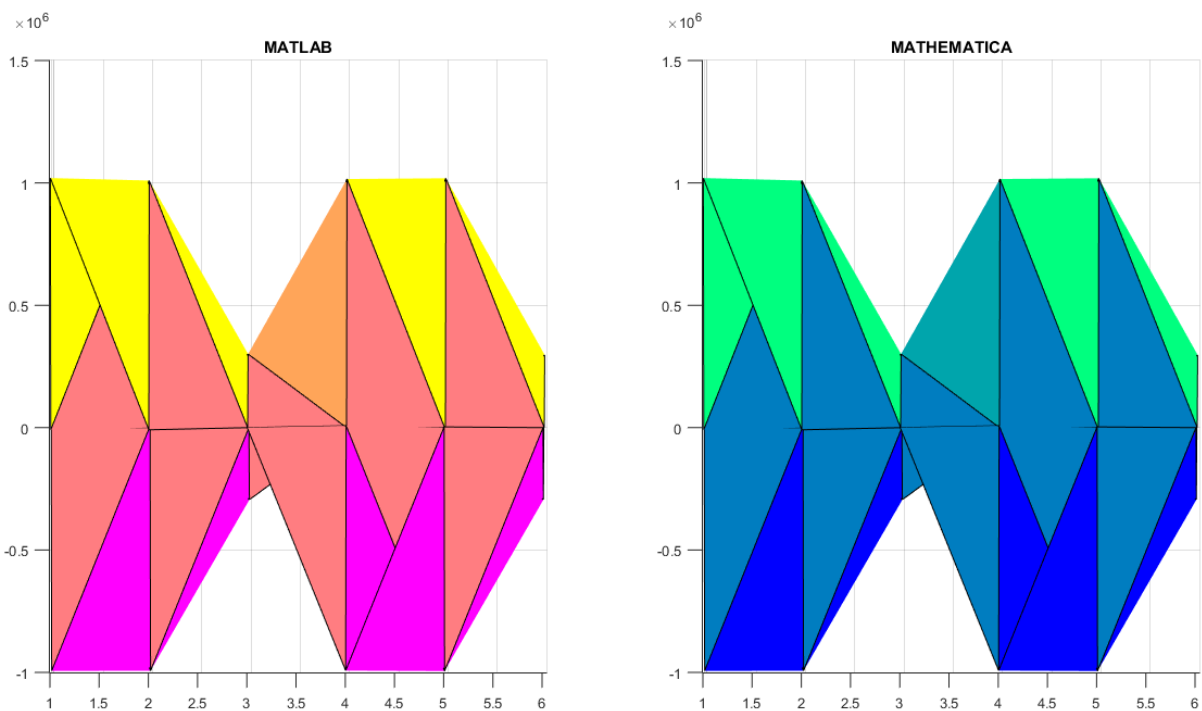


Figura 27. Vista lateral. Mostrando los resultados para los valores reales con una y constante

A la vista de los resultados se puede acertar a decir que el código en MATLAB aproxima de manera correcta los valores obtenidos originalmente en Mathematica.

Una vez conocido que el código desde el que se trabaja muestra los mismos valores que el código de partida, ha de comprobarse de que la hipótesis inicial que se busca ratificar es en efecto cierta: el cálculo se realiza de manera más rápida con MATLAB que con Mathematica.

Para una serie de distintos valores se ha cronometrado cuanto se tarda en el cálculo de la matriz Φ_{ij} en ambos métodos. Se ha incluido a la hora de cronometrar la definición de variables, las posibles llamadas a códigos auxiliares y el propio desarrollo del cálculo, fuera están procesos como las representaciones gráficas. Para Mathematica, la función utilizada es *TimeUsed []*, mientras que en MATLAB son los comandos *tic* y *toc*.

Es indiferente el par de valores dados, ambos códigos no se ven perjudicados por los parámetros de entrada. Para un ejemplo cualquiera, los valores obtenidos en las 10 primeras ejecuciones son los siguientes:

Tabla 6. Tiempos de resolución en segundos

MATLAB	Mathematica
0.006619	4.767
0.009171	5.173
0.006111	4.983
0.007801	5.017
0.005794	5.281
0.010212	5.39
0.005533	5.069
0.007007	5.158
0.007939	5.421
0.005183	5.564

Hay que señalar que cada vez que ambos códigos se han ejecutado, se ha procedido previamente a hacer una limpieza de la memoria acumulada mediante los comandos *clear all* en MATLAB y *ClearSystemCache []* en Mathematica.

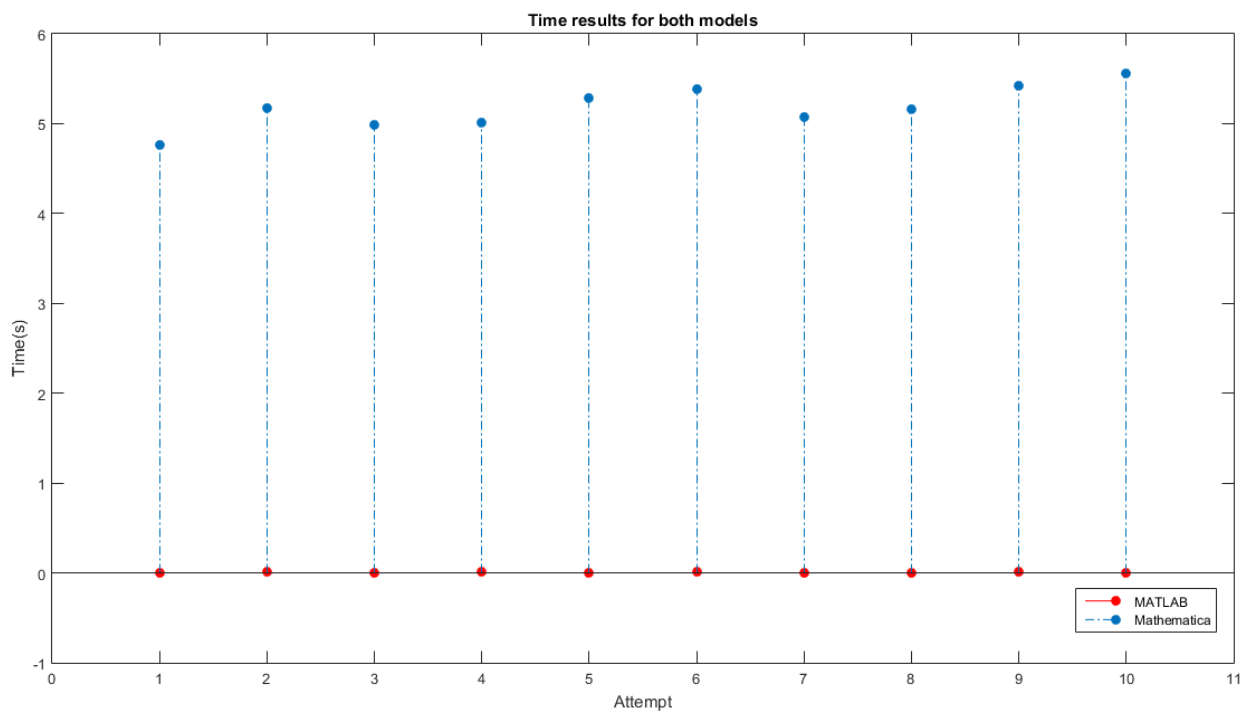


Figura 28. Tiempos de resolución, en segundos, de ambos métodos

Se puede ver como el tiempo de ejecución en Mathematica ronda los 5 segundos, mientras que en el código propuesto los tiempos son del orden de las milésimas de segundo. Los datos se analizarán en el siguiente punto.

5 CONCLUSIONES Y TRABAJOS FUTUROS

*El futuro no es lo que va a pasar,
sino lo que vamos a hacer.
- Jorge Luis Borges-*

En base a lo obtenido anteriormente, en este capítulo se busca sintetizar e intentar dar respuesta a las cuestiones planteadas al comienzo del presente trabajo. Es conveniente también incluir los posibles caminos a seguir en el caso de continuar con la investigación aquí iniciada.

5.1 Análisis de los resultados

Como objetivo se había propuesto alcanzar una representación computacional de un potencial *Tight Binding* mediante las herramientas que proporciona MATLAB. En caso de que esto fuera posible, es de recibo analizar si su empleo pudiese ser de algún modo beneficioso para un empleo científico.

La primera parte se cumple al representar el modelo con exactitud los resultados que se obtienen de Mathematica, y por tanto de la realidad, con una precisión más allá de los decimales proporcionados al usuario por pantalla. Una vez sabido que el modelo reflejará de manera fidedigna los resultados numéricos, un análisis de rendimiento en la ejecución de ambos programas indica que, bajo las mismas condiciones, MATLAB proporciona un consumo de CPU menor que Mathematica tal y como se había asumido.

El motivo por el cual esto sucede es por el diseño de MATLAB para el cálculo numérico. Su potencia radica en el manejo matricial, perfecto para la resolución de sistemas como el que se ha traducido. Si bien es cierto que es menos elegante y precisa de un conocimiento básico de su lenguaje para llevar a cabo ciertas funciones, los resultados proporcionan resultados en un **0.1%** el tiempo de Mathematica.

En labores científicas en las que los tiempos de computación van más allá de los encontrados en este trabajo, una reducción como tal implica ahorros sustanciales que hacen considerar este programa como una alternativa real.

5.2 Trabajos futuros

Aunque se ha logrado alcanzar la respuesta a las preguntas planteadas, es de interés suponer extensiones a lo aquí desarrollado para poder seguir aumentando en eficiencia y versatilidad los cálculos sobre las redes de grafeno.

- Extender el alcance del cálculo. En el modelo descrito, se ha puesto punto y final al cálculo una vez obtenida la matriz Φ_{ij} , pero por supuesto el potencial tiene muchos más puntos que se podrían ver beneficiados por este nuevo sistema de cálculo.

- Integrar otros tipos de potenciales. Al constar de mayor agilidad en el cálculo computacional, se podría plantear la integración de modelos ab initio. Se alcanzaría así resultados numéricos de mayor precisión con unos tiempos de computacional asequibles.
- Mejora en la experiencia de usuario. MATLAB permite la creación de aplicaciones ejecutables para condensar las variables de entrada de un código mediante inputs de entrada de *html* que facilitan a un usuario no conocedor del programa la obtención de resultados de una manera más limpia. Esto puede ser combinado con unas representaciones gráficas más detalladas bien a base de atributos del propio MATLAB, bien mediante la conexión con un programa externo que tome los resultados y los muestre con un mayor nivel de detalle.
- Uso de MATLAB Coder. Una de las muchas *third party applications* ya comentadas que permite la traducción de un código de MATLAB directamente al lenguaje C++. Es preciso ser conocedor minimamente de este último para poder cubrir los puntos flacos que puedan surgir de la traducción mediante este proceso. C++ consta de gran versatilidad y puede seguir reduciendo costos computacionales si se implementase.

APÉNDICE A

TRANSFORMADA DE FOURIER

A.1 Definición matemática

La transformada discreta de Fourier de $f: \mathbb{Z}^n \rightarrow \mathbb{R}$ se define como:

$$\hat{f}(\boldsymbol{\theta}) = \sum_{\mathbf{l} \in \mathbb{Z}^n} f(\mathbf{l}) e^{-i\boldsymbol{\theta} \mathbf{l}} \quad (\text{A.1})$$

Siendo su expresión inversa, o también llamada antitransformada:

$$f(\mathbf{l}) = \frac{1}{(2\pi)^n} \int \hat{f}(\boldsymbol{\theta}) e^{i\boldsymbol{\theta} \mathbf{l}} d^n \boldsymbol{\theta} \quad (\text{A.2})$$

Esta última integral comprendida entre $[-\pi, \pi]^n$.

A.2 Propiedades fundamentales

Definiendo la convolución como

$$f * g(\mathbf{l}) = \sum_{\mathbf{l}' \in \mathbb{Z}^n} f(\mathbf{l} - \mathbf{l}') g(\mathbf{l}'), \quad (\text{A.3})$$

se obtiene que

$$\widehat{f * g} = \hat{f} \hat{g} \quad (\text{A.4})$$

Resultado a menudo conocido como Teorema de la convolución. Suponiendo además que f y g son cuadrados sumables

$$\sum_{\mathbf{l} \in \mathbb{Z}^n} f(\mathbf{l}) g^*(\mathbf{l}) = \frac{1}{(2\pi)^n} \int \hat{f}(\boldsymbol{\theta}) \hat{g}^*(\boldsymbol{\theta}) d^n \boldsymbol{\theta} \quad (\text{A.5})$$

que da lugar a la identidad de Parseval para la DFT.

A.3 Funciones periódicas

El empleo de la transformada de Fourier a funciones periódicas es de gran interés en las aplicaciones prácticas. Sea un conjunto $Y \subset \mathbb{Z}^n$, la celda unidad, tal que su traslación $\{Y + \mathbf{L}^i \mathbf{A}_i, \mathbf{L} \in \mathbb{Z}^n\}$, para algunos vectores de traslación $\mathbf{A}_i \in \mathbb{Z}^n$ con $i = 1, \dots, n$, define una partición de \mathbb{Z}^n .

Sea \mathbf{A}_i la correspondiente base dual y $\mathbf{B}_i = 2\pi \mathbf{A}_i$ la base recíproca. Una función de red $f: \mathbb{Z}^n \rightarrow \mathbb{R}$ es Y -periódica si $f(\mathbf{l}) = f(\mathbf{l} + \mathbf{L}^i \mathbf{A}_i)$ para todo $\mathbf{l} \in \mathbb{Z}^n$. La transformada discreta de Fourier de una función de

red Y -periódica se puede escribir de la siguiente forma

$$\hat{f}(\boldsymbol{\theta}) = \sum_{\mathbf{L} \in \mathbb{Z}^n} \sum_{\mathbf{l} \in Y} f(\mathbf{l}) e^{-i\boldsymbol{\theta}(\mathbf{l} + \mathbf{l}' A_j)} = \frac{1}{|Y|} \left\{ \sum_{\mathbf{l} \in Y} f(\mathbf{l}) e^{i\boldsymbol{\theta} \mathbf{l}} \right\} \left\{ |Y| \sum_{\mathbf{L} \in \mathbb{Z}^n} e^{i\boldsymbol{\theta}(\mathbf{L} + \mathbf{l}' A_j)} \right\} \quad (\text{A.6})$$

donde $|Y|$ es el número de puntos en Y . Pero

$$|Y| \sum_{\mathbf{L} \in \mathbb{Z}^n} e^{i\boldsymbol{\theta}(\mathbf{L}' A_j)} = (2\pi)^n \sum_{\mathbf{H} \in \mathbb{Z}^n} \delta(\boldsymbol{\theta} - \mathbf{H}_i \mathbf{B}^i) \quad (\text{A.7})$$

y, por tanto,

$$\hat{f}(\boldsymbol{\theta}) = \frac{(2\pi)^n}{|Y|} \sum_{\boldsymbol{\theta} \in Z} \hat{f}_{\boldsymbol{\theta}} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}) \quad (\text{A.8})$$

donde

$$\hat{f}(\boldsymbol{\theta}) = \sum_{\mathbf{l} \in Y} f(\mathbf{l}) e^{-i\boldsymbol{\theta} \mathbf{l}} \quad (\text{A.9})$$

y Z es la intersección B^i y $[-\pi, \pi]$. Además, la antitransformada resulta en

$$f(\mathbf{l}) = \frac{1}{|Y|} \sum_{\boldsymbol{\theta} \in Z} \hat{f}_{\boldsymbol{\theta}} e^{i\boldsymbol{\theta} \mathbf{l}} \quad (\text{A.10})$$

Para funciones periódicas, la identidad de Parseval toma la forma

$$\sum_{\mathbf{l} \in Y} f(\mathbf{l}) g^*(\mathbf{l}) = \frac{1}{|Y|} \sum_{\boldsymbol{\theta} \in Z} \hat{f}_{\boldsymbol{\theta}} \hat{g}_{\boldsymbol{\theta}}^* \quad (\text{A.11})$$

Del mismo modo, sean f y g funciones de red de valores complejos, esta última periódica. Sustituyendo (A.8) en el teorema de convolución se obtiene

$$(\widehat{f * g})(\boldsymbol{\theta}) = \hat{f}(\boldsymbol{\theta}) \left(\frac{(2\pi)^n}{|Y|} \sum_{\boldsymbol{\theta} \in Z} \hat{g}_{\boldsymbol{\theta}} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}) \right) = \frac{(2\pi)^n}{|Y|} \sum_{\boldsymbol{\theta} \in Z} \hat{f}(\boldsymbol{\theta}) \hat{g}_{\boldsymbol{\theta}} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}) \quad (\text{A.12})$$

de donde es posible razonar que

$$(\widehat{f * g})_{\boldsymbol{\theta}} = \hat{f}(\boldsymbol{\theta}) \hat{g}_{\boldsymbol{\theta}} \quad (\text{A.13})$$

Finalmente, la media de una función periódica es tal que

$$\langle f \rangle = \frac{1}{|Y|} \sum_{\mathbf{l} \in Y} f(\mathbf{l}) = \frac{1}{|Y|} \hat{f}_{\boldsymbol{\theta}} \quad (\text{A.14})$$

BIBLIOGRAFÍA

- [1] M.P. Ariza and M. Ortiz, *Discrete dislocations in graphene*, (2010).
- [2] Juan Pedro Méndez Granado, *Harmonic/Nonharmonic model of graphene and its structural defects based on a Tight Binding interatomic potential*, (2015).
- [3] C. H. Xu, C. Z. Wang, C.T. Chan and K. M. HO, *A transferable tight-binding potential for carbon*, (1992).
- [4] Rupali Kundu, *Tight binding parameters for graphene*, (2009).
- [5] Robert Ipanaqué Chero, Ricardo Velesmoro León, *Breve maual de Mathematica*, (2005).